Probabilistic Analysis of Distributed Processes with Focus on Consensus

by

Frederik Mallmann-Trenn

M.Sc., Simon Fraser University, 2014
B.Sc., Universität Paderborn, 2012

Dissertation Submitted in Partial Fulfillment of the
Requirements for the Joint Degree (cotutelle) of
Doctor of Philosophy
and
Doctor of the École normale supérieure
(Paris Sciences Lettres)

in the
School of Computing Science, Faculty of Applied Sciences
at Simon Fraser University (Canada);
Département d'informatique, École normale supérieure,
École doctorale 386: Sciences mathématiques de Paris Centre (France)

Simon Fraser University
Fall 2017

Approval

Name: **Frederik Mallmann-Trenn**

Degree: **Doctor of Philosophy**
and
Doctor of the École normale supérieure
(Paris Sciences Lettres) (Computer Science)

Title: ***Probabilistic Analysis of Distributed Processes with Focus on Consensus***

Supervisory Committee: **Chair:** Bartek Błaszczyszyn
Research Director

Petra Berenbrink (Senior Supervisor)
Professor
School of Computing Science SFU

Bartek Błaszczyszyn (Internal Examiner ENS)
Research Director
Département d'informatique ENS

Andrei Bulatov (Internal Examiner SFU)
Professor
School of Computing Science SFU

Colin Cooper (Reviewer)
Professor
Department of Informatics
King's College London

Pierre Fraigniaud (External Examiner, Reviewer)
Research Director
IRIF

Philippe Jacquet (External Examiner)
Research Director
NOKIA Bell Labs

Claire Mathieu (Senior Supervisor)
Research Director
Département d'informatique ENS

Date Approved: 22 September 2017

Abstract

This thesis is devoted to the study of stochastic decentralized processes. Typical examples in the real world include the dynamics of weather and temperature, of traffic, the way we meet our friends, etc. We take the rich tool set from probability theory for the analysis of Markov Chains and employ it to study a wide range of such distributed processes: Forest Fire Model (social networks), Balls-into-Bins with Deleting Bins, and fundamental consensus dynamics and protocols such as the Voter Model, 2-Choices, and 3-Majority.

keywords— stochastic processes; distributed computing; consensus; leader election; random walks; social networks

Résumé

Cette thèse est consacrée à l'étude des processus stochastiques décentralisés. Parmi les exemples typiques de ces processus figurent la dynamique météorologique, la circulation automobile, la façon dont nous rencontrons nos amis, etc. Dans cette thèse, nous exploitons une large palette d'outils probabilistes permettant d'analyser des chaînes de Markov afin d'étudier un large éventail de ces processus distribués : modèle des feux de forêt (réseaux sociaux), balls-into-bins avec suppression, et des dynamiques et protocoles de consensus fondamentaux tels que Voter Model, 2-Choices, et 3-Majority.

mots clés— processus stochastiques; processus distribués; consensus; élection de chef; marches aléatoires; réseaux sociaux

Acknowledgments

First and foremost, I'd like to thank my advisors Petra Berenbrink and Claire Mathieu for their indispensable support and advice extending far beyond research.

I thank Colin Cooper and Pierre Fraigniaud for having accepted to review this manuscript and I'm grateful to Bartek Błaszczyszyn, Andrei Bulatov, and Philippe Jacquet for being part of the jury.

Special thanks go to Vincent Cohen-Addad and Varun Kanade, who I consider to be my academic big brothers and valuable friends. I also thank Robert Elsässer for his support and all the intuition he shared about randomized processes. Thanks go to Ralf Klasing and Adrian Kosowski for inviting me, and to George Giakkoupis and Anne-Marie Kermarrec for the research visit me during my Masters. I thank Thomas Sauerwald for inviting me to Cambridge and for sharing so much of his knowledge about random walks.

It was a great pleasure to be part of the TALGO group at ENS and I thank Chien-Chung Huang, Éric Colin de Verdière, Zhentao Li, Hang Zhou, and especially Víctor Verdugo for the nice moments and inspiring discussions. I also wish to thank Friedhelm Meyer auf der Heide and Peter Kling for their advice and for enabling me to participate early in research, as well as Achim Willenbring who laid the foundation of my knowledge in theoretical computer science.

Thanks goes to my PhD fellows Dominik Kaaser and Emanuele Natale for being great and reliable collaborators and friends.

I would also like to thank my collaborators Andrea Clementi, Tom Friedetzky, Reut Levi, Zvi Lotker, Przemysław Uznański, and Chris Wastell for their discussions, contributions, and the good time.

Thanks to my friends: Adrien, Ash, Christian, les dentists de Rennes, Edouard, Hedayat, Iman, Marten, Max, Mehran, Leila, los melocotones, die Reinoldus AG (Arne, Theres, Tristan, Quay), Remy, Sara, Thianhlanh, and Till.

Merci Célia pour ton soutien et ton énergie positive.

Zu guter Letzt danke ich meiner Familie für die unendliche Unterstützung.

Contents

Approval ii

Table of Contents v

List of Figures ix

1 Introduction 2

1.1 Introduction française . . . 9

1.2 Organization and Publications . . . 14

2 General Notation 17

I Probabilistic Analysis of Distributed Dynamic Processes 20

3 Contributions Dynamic Processes 21

4 Social Networks: The Forest Fire Model [KLL+16] 25

4.1 Results . . . 27

4.2 Approach and Technical Contributions . . . 28

4.3 Related work . . . 28

4.4 Relating graph and line process . . . 30

4.4.1 Line process . . . 30

4.4.2 The ambassador graph . . . 31

4.4.3 Coupling . . . 33

4.4.4 Proofs of the Graph results from the Line results - Proof of Theorem 4.1 and Theorem 4.3 . . . 34

4.5 Analysis of the Line Fire Process . . . 35

4.5.1 High-Level Proof Overview . . . 35

4.5.2 Proof of Lemma 4.4 . . . 38

4.5.3 Proof of Lemma 4.13 . . . 39

4.5.4 Proof of Proposition 4.14 . . . 44

4.5.5 The foundations of ϕ . . . 50

4.6 Lower bound - Proof of Lemma 4.5 . . . 51
4.7 Future Work and Conclusion . . . 52
4.7.1 Further models . . . 53
4.7.2 Conclusion . . . 54

5 Balls-into-Bins with Deleting Bins [BFK+16a] 55
5.1 Results . . . 56
5.2 Approach and Technical Contributions . . . 56
5.3 Related Work . . . 57
5.4 Model & Preliminaries . . . 60
5.5 The 1-Choice Process . . . 61
5.5.1 Maximum Load - Proof of Theorem 5.2 . . . 62
5.5.2 Stability – Proof of Theorem 5.1 . . . 63
5.5.3 Lower Bound on Maximum Load - Proof of Theorem 5.3 . . . 64
5.6 The 2-Choice Process . . . 66
5.6.1 Bounding the Smoothness – Proof of Proposition 5.7 . . . 67
5.6.2 Maximum Load – Proof of Theorem 5.6 . . . 76
5.6.3 Stability – Proof of Theorem 5.5 . . . 80

6 Future Work - Dynamic Processes 83

II Probabilistic Analysis of Consensus Dynamics and Protocols 84

7 Contributions Consensus Processes 85
7.1 Consensus Dynamics . . . 86
7.1.1 Overview of Results . . . 87
7.2 Consensus Protocols . . . 89
7.2.1 Overview of Results . . . 90
7.3 Related Work . . . 91
7.3.1 Voter Model . . . 91
7.3.2 2-Choices . . . 92
7.3.3 3-Majority . . . 93
7.3.4 Further Consensus Dynamics . . . 93
7.3.5 Consensus Protocols . . . 94

8 Voter Model [KMS16] 98
8.1 Results . . . 98
8.2 Approach and Technical Contributions . . . 100
8.3 Bounding the coalescence time . . . 102
8.3.1 The coalescence process . . . 102

8.3.2 A more amenable process . . . 103
8.3.3 Meeting Time Distribution Prior to t_{meet} . . . 108
8.3.4 Upper Bound - Proof of Theorem 8.1 . . . 111
8.3.5 Lower Bound - Proof of Theorem 8.2 . . . 117

9 3-Majority [BCE+17] 131
9.1 Results . . . 132
9.2 Approach and Technical Contributions . . . 133
9.3 Consensus Model & Technical Framework . . . 134
9.3.1 Comparing Anonymous Consensus Processes . . . 134
9.3.2 Coupling two AC-Processes - Proof of Theorem 9.4 . . . 135
9.4 Upper Bound for 3-Majority- Proof of Theorem 9.8 . . . 137
9.4.1 Analysis of Phase 1: 3-Majority vs. Voter . . . 138
9.4.2 Analysis of Phase 1: A Bound for Voter . . . 139
9.5 Limitations of 1-Step Coupling . . . 144

10 2-Choices [EFK+16, BCE+17] 146
10.1 Results . . . 147
10.2 Approach and Technical Contributions . . . 148
10.3 Plurality Consensus with Two Choices . . . 148
10.3.1 Upper bound - Proof of Theorem 10.2 . . . 149
10.4 Lower Bound for 2-Choices . . . 153
10.4.1 Worst-Case Lower bound - Theorem 10.5 . . . 153
10.4.2 Complementing our Upper bounds Theorem 10.6 and Theorem 10.7 155
10.5 Comparison with the 3-Majority Process . . . 158

11 Rapid Asynchronous Consensus [EFK+16] 160
11.1 Results . . . 161
11.2 Approach and Technical Contributions . . . 162
11.3 The Asynchronous Protocol . . . 164
11.4 The Key Lemmas . . . 167
11.5 Concentration of the Clocks: Proof of Proposition 11.3 . . . 168
11.6 Analysis of the 2-Choices sub-phase: Proof of Proposition 11.4 . . . 175
11.7 Analysis of the Bit-Propagation Sub-Phase: Proof of Proposition 11.5 . . . 176
11.8 The Endgame: Taking a from $(1-\varepsilon_{\text{Part1}})\cdot n$ to n . . . 181
11.9 Putting Everything Together: Proof of Theorem 11.1 . . . 183
11.10 Increasing the Number of Opinions . . . 185
11.11 Conclusions and Further Work . . . 185

12 Consensus via Load Balancing [BFK+16b] 187

12.1 Results . . . 188
12.2 Approach and Technical Contributions . . . 189
12.3 Communication Model and Notation . . . 190
12.4 Protocol SHUFFLE - Theorem 12.1 . . . 192
12.4.1 Protocol Description . . . 193
12.4.2 Analysis of SHUFFLE . . . 194
12.5 Protocol BALANCE - Theorem 12.9 . . . 204

13 Future Work - Distributed Consensus Processes **206**

Bibliography **208**

A Probabilistic Preliminaries **220**
A.1 Elementary Bounds . . . 221
A.2 Concentration Inequalities and Drift Analysis . . . 221
A.2.1 Binomial Distribution . . . 221
A.2.2 Negative Regression . . . 222
A.2.3 Poisson Distribution . . . 222
A.2.4 Hajek's Theorem . . . 223
A.2.5 Drift Theorem . . . 224
A.2.6 Doob-Martingale . . . 225
A.2.7 Azuma-Hoeffding Inequality . . . 225
A.3 Markov Chains . . . 226
A.3.1 Couplings . . . 226
A.3.2 Strassen's Theorem . . . 226
A.3.3 Positive Recurrence . . . 227
A.4 Markov Chains: Random Walks . . . 227
A.4.1 Spectral Gap . . . 228
A.4.2 Separation Time (t_{sep}) . . . 229
A.4.3 Ramanujan graphs . . . 230
A.4.4 Mixing time bounds via Hitting time . . . 231
A.5 Random Processes . . . 231
A.5.1 Galton-Watson Trees . . . 231
A.5.2 Balls-Into-Bins . . . 232
A.5.3 Póly-Urn . . . 232

List of Figures

Figure 1.1 A social network generated by the Forest Fire Process. 3
Figure 1.2 Typical round of GREEDY[2]. 5
Figure 1.3 Illustration of the central consensus dynamics. 6
Figure 1.4 A réseau social généré par le forest fire processus. 10
Figure 1.5 Un tour typique de GREEDY[2]. 11
Figure 1.6 Illustration des dynamiques aléatoires. 12

Figure 3.1 The general potential approach. 23

Figure 4.1 A social network generated by the Forest Fire Process. 26
Figure 4.2 The change of the potential ϕ in the Forest Fire Process. 36
Figure 4.3 A different take on the potential change ϕ in the Forest Fire Process model. 37
Figure 4.4 The percolation Burn Process (Algorithm 5). 40

Figure 5.1 Typical round of GREEDY[2]. 60
Figure 5.2 Combinatorial approach to bounding the maximum load. 77

Figure 7.1 Illustration of the central consensus dynamics. 86

Figure 8.1 The graph described in Section 8.3.5 with $t_{\text{coal}} = \Omega(t_{\text{meet}} + \sqrt{\alpha} \cdot \log n \cdot t_{\text{mix}})$. 101
Figure 8.2 Illustration of the process P_{imm}. 103
Figure 8.3 An illustration of couplings between the processes. 107
Figure 8.4 The graph described in Section 8.3.5 with $t_{\text{coal}} = \Omega(t_{\text{meet}} + \sqrt{\alpha} \cdot \log n \cdot t_{\text{mix}})$. 118

Figure 9.1 Reducing from consensus to coalescing. 140

Figure 11.1 Graphical representation of one phase of Algorithm 10. Each phase consists of $T = 10 \cdot \Delta$ ticks. 165

Figure 12.1 The coupling of tokens. 197
Figure 12.2 High-level idea of Lemma 12.8. 202

Preliminaries

Chapter 1

Introduction

"So much of life, it seems to me, is determined by pure randomness."

- Sidney Poitier

This thesis is devoted to the study of *stochastic distributed processes*, *i. e.,* processes evolving as a result of randomized decisions of interacting autonomous entities. Such processes surround us in many aspects of our daily lives and are imperative in the realm of distributed computing. Archetypal examples of such processes include the evolution of social networks, distribution of jobs in cloud computing, propagation of opinions and diseases, movement of atoms, evolution of stock markets, etc.[1]

We focus on two groups of these processes: (i) *dynamic processes*, in which the entities arrive (and leave) over time, and (ii) *consensus processes*, in which the entities of a static network interact continuously with each other in order to reach consensus.

The dynamic processes we study in this thesis (Part I) are the *Forest Fire Model*, a model for the creation of social networks, and *Balls-into-Bins with Deletions* modeling the load distribution in systems. We are interested in the asymptotic behavior of these processes as the time goes to infinity.

The consensus processes (Part II) we study are: *Voter*, *2-Choices*, *3-Majority*, as well as faster, albeit slightly more sophisticated, processes.

At first glance, the wide-range of processes considered in this thesis appear to be unrelated; however, the common thread among them lies in our analysis. We relate each of these processes to (general) random walks and employ machinery from different areas of probability theory to analyze them.

[1] Although not all of these processes are inherently stochastic–possibly even deterministic–accurate predictions by means of deterministic models are often infeasible. The sheer amount of data required alone renders this impractical, not to mention the complexity of the computations. Instead, treating theses processes as being random allows, in some cases, for good estimates such as predicting the number of times a die shows 6.

Part I: Dynamic Processes

The first process we study is the Forest Fire Process, a model for generating random graphs representing the evolution of social networks. The nodes in the generated graph represent the users and and edge represents friendship between the adjacent nodes.

Forest Fire Process. In the *Forest Fire Process*, a new user u arrives in every round, and connects to another user v chosen uniformly at random among those already present. User u then becomes friends with a randomly chosen subset of v's friends. These friends then introduce u to some of their friends and so on. The process stops once there is no user left who is willing to introduce u to any of their friends. Once this happens, an edge is added between u and those nodes u was introduced to. This concludes the round and the next rounds begins with the arrival of a new node executing the same process. See Chapter 4 for the precise model and Figure 1.1 for an illustration.

Ten years ago, Leskovec et al. [LKF07] conjectured, based on their simulations, that the Forest Fire Process exhibits the small-world effect, *i. e.,* the expected distance between any pair of users is constant. Under mild assumptions, we were able to prove this rigorously. Furthermore, we show under certain conditions–in which the presence of edges is more unlikely–that the expected distance is logarithmic in the number of rounds.

Studying the Forest Fire Process and other social networks is invaluable in understanding the societies we live in—real and virtual. Communication and the interaction with other human beings has shifted from face-to-face interactions towards online social networks. The social networks surrounding us go far beyond friendships and include business cooperations [Pik13, Jac+08, Jac05], academic collaborations, the spread of diseases [EGA+04], marketing [SR03], romantic relations, and even political ideas. For example, in marketing it is well-known that targeted advertisement is more effective than generic advertisement [Joh13].

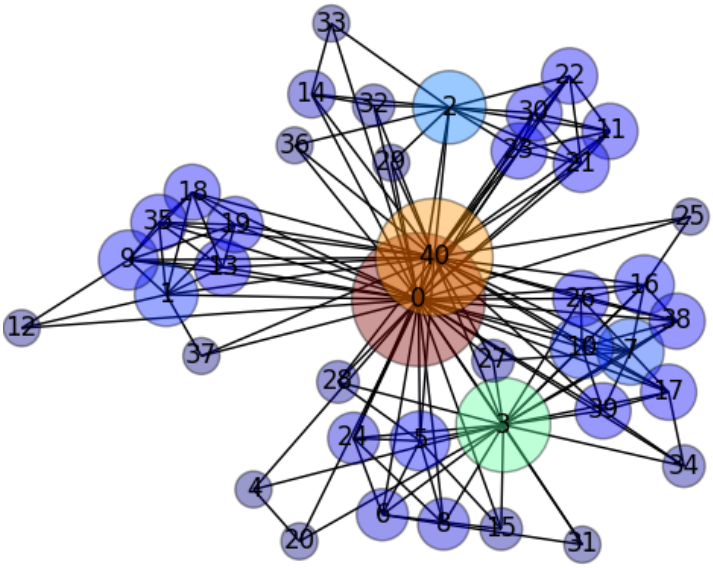


Figure 1.1: A social network generated by the Forest Fire Process. The node sizes and the colors are a function of the degrees and the node labels correspond to the arrival times.

Knowing a user's social network is therefore an important part of efficient marketing as suggested by the saying: "*Show me your friends and I'll tell you who you are*". Moreover, studying social networks can fundamentally change the way we think about social networks and society. A concrete example is the *Milgram Experiment* [TM67] in which participants are asked to send a letter to an unknown person by only forwarding it to people they know on a first-name basis. The recipients then forward the letters by following the same rule, and so on, until the letters reach their intended destination. The experiment had the surprising outcome that the average length of the chain, of the letters that arrived, was 6.2. This experiment branded the terms "small-world", and "six degrees of separation". Nevertheless, the experiment left the question open as to how social networks are structured; inviting mathematicians to design suitable models–such as the Forest Fire Process [LKF07]. The motivation behind studying the Forest Fire Process in particular are twofold. First, the Forest Fire Process models that we often meet our friends through mutual friends. Second, the Forest Fire Process encapsulates–as simulations suggest [LKF07]–three important properties observed in many social networks: (i) the small-world effect, (ii) the "densification" of edges, *i. e.,* the number of edges in the network is super constant in the number of users, and (iii) a power-law distribution of the out-edges. See Chapter 4 for an overview of other social network models.

The key to the analysis is a potential function, which allows us to show that whenever the distance of a user in the network to the initial set of users is large, then the potential will decrease in expectation over the course of the arrival of the next *two* users. In particular, the potential can be modeled as a general random walk on the natural numbers with a drift towards zero (apart from finitely many states) and tail bounds on the absolute change per round. To bound the potential change, we couple the original process with a Galton-Watson Tree.

Balls-into-Bins with Deletions. The Balls-into-Bins with Deletions process models the load distribution in a system with n nodes (bins) representing n queues. Tasks (balls) arrive over time and move to bins according to different strategies. We consider two strategies: Greedy[1] and Greedy[2]. The process works as follows. At each round a batch of 0 to n balls arrives: Each of n potential balls spawns w.p. $\lambda < 1$ and each of the spawned balls chooses (i) uniformly at random from n bins (Greedy[1]) or (ii) greedily from two bins sampled uniformly at random from n bins (Greedy[2]). At the end of each round, all non-empty bins delete one ball each. See Figure 1.2 for an illustration.

We give bounds on the load of the bins after an arbitrary number of steps (possibly super-exponential in the number of bins n) and show an exponential difference in the load of processes Greedy[1] and Greedy[2]. We show that the corresponding Markov Chains are positive recurrent and that there is an exponential difference in the load, similarly to the classical two-choices Balls-into-Bins process [ABKU99].

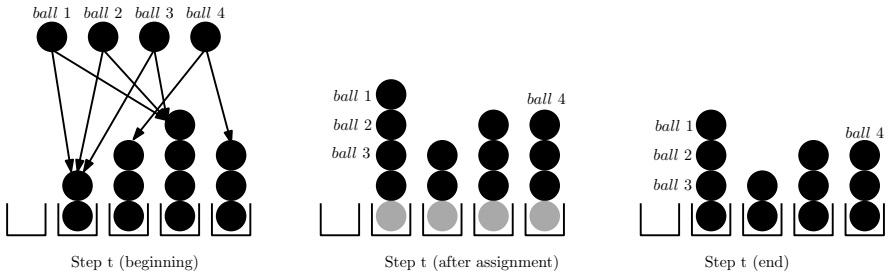


Figure 1.2: The figure depicts a typical round of GREEDY[2]. In this example we have $n = 5$ and 4 balls arrive. Balls 1, 2, and 3 choose the same bin with a load of 2 and a bin with larger node and hence all move the same bin resulting in that bin having the highest load. Moreover, Ball 4 chooses two bins with equal load and chooses one of these uniformly at random. At the end of the round all non-empty bins delete one ball (marked gray).

The Balls-into-Bins with Deletions process can be used to model customers accessing a web-service are assigned to servers, collisions protocols used for contention resolution message routing, as well as (iii) real-life queues at airports, supermarkets, *etc.* [CMM+98, EG16, MRS00]. Further applications are hashing, shared memory emulations on distributed memory machines, load balancing with limited information, and low-congestion circuit routing [MRS00]. Knowing the length of the queues, and therefore the time a user or task spends in the queue is helpful in the design of systems in which waiting times are a major concern.

The analysis follows–on a superficial level–the approach (which we also used in the Forest Fire Process) of reducing the underlying problem to a potential that performs a biased random walk with a drift towards zero (apart from finitely many states) and tail bounds on the absolute change per round. This time, however, it is not necessary to consider two consecutive time steps at once. On the other side, we are faced with a different challenge: It seems hard to "condense" all relevant properties of the load distribution into one single potential. For that reason, our analysis builds on the careful analysis of the interplay of *three* different potentials. Each of these potentials characterizes features of the load distribution, by mapping the features of the load distribution to a natural number.

In the same spirit as in the Forest Fire Process, the changes in the potential can be treated as a general random walk with a drift. However, this time the potentials are in expectation no longer constant but a function of the time step $t = \omega(\text{poly}(n))$. In order to get strong bounds at time t, we first use union bounds in an adaptive way to get rough bounds on the potentials at all time steps up to time step t. Using combinatorial arguments, we show that there must have been a state with a favorable load distribution at some round $t-\text{poly}(n)$. From there on we can apply a more fine-grained potential analysis to characterize the load distribution at time t.

See Chapter 3 for further details about the high-level analysis and common aspects of both dynamic processes.

Part II: Consensus Processes

The second part of the thesis focuses on the study of consensus processes through the lens of Markov Chains. Consensus processes are processes in which each node of a graph starts with an opinion[2] and all nodes execute simple protocols with the goal of attaining *consensus* quickly, *i. e.,* to agree on one opinion. We distinguish between *consensus dynamics*, which are very simple and require little memory and communication, and *consensus protocols*, which are faster in reaching consensus at the cost of being more complicated often requiring more memory and communication. See Chapter 7 for more details.

It is well-known (*e. g.,* [Ang80, Lyn89]) that reaching consensus deterministically is impossible in many settings of interest. A natural way of defying the impossibility results is by using randomized protocols - the nucleus of this thesis. The prevailing randomized dynamics in this area are Voter, 2-Choices, and 3-Majority, which we define below (see also Figure 1.3 for an illustration). Voter is arguably the simplest randomized dynamic possible and 2-Choices and 3-Majority are almost equally simple. However, 2-Choices and 3-Majority are at the same time efficient self-stabilizing solutions for *Byzantine agreement* [PSL80, Rab83]: achieving consensus in the presence of an adversary that can disrupt a bounded set of nodes each round [BCN+14b, BCN+16, CER14, EFK+16]. We generalize each of these protocols in different ways and settle the question of the fastest protocol among these three on the complete graph.

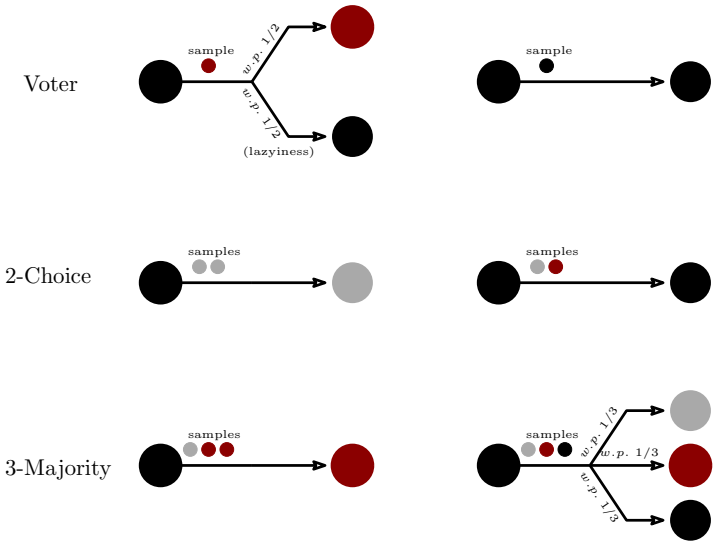


Figure 1.3: The figure depicts the three models with two examples each.

Applications of consensus dynamics and protocols are manifold: The nodes may represent machines in a network: Consider Bitcoin, where the need of reaching consensus in a distributed fashion arises frequently whenever two or more parties "mine" a new block (bundling transactions) simultaneously; otherwise a transaction could be accounted for multiple times. In fact, distributed consensus is one of the most fundamental problem in distributed computing with many applications [DGM+11, Pel02, PVV09, CIG+15, BMPS04]. Arguably the most prominent special case of consensus protocols is *leader election*, the

[2]We assume that there is no ordering of the opinions.

heart of distributed computing. In leader election, all nodes start with distinct opinions and need to agree quickly on one opinion. Applications typically demand both very simple and space-efficient protocols.

Consensus processes can also be used to develop a better understanding of how opinions and believes spread in social networks, as well as for other application in social networks [MS10, MNT14]. Such insights could be used to develop strategies such as determining how and where to distribute free samples of a new product in order to raise market shares. Moreover, developing simple consensus dynamics helps to understand how communication in nature works, *e. g.,* among ants and birds works, as well as many other aspects of nature [BDDS10, CER14, FPM+02, CDS+13, CC12]. In the following we give an overview of the results and techniques we show for each of the consensus processes.

The Voter model (see Chapter 8) works as follows. Initially, every node has a distinct opinion and in each synchronous round each node samples a neighbor uniformly at random and adopts its opinion. It is well-known that the consensus time follows the same distribution as the coalescence time, which is defined as follows. The coalescence time is expected time it takes for n independent random walks starting from different nodes to absorb one another. We thus study the more amenable consensus time and express it in terms of two fundamental quantities related to random walks: The mixing time and the meeting time. As a side product, we obtain tail bounds on the meeting time of random walks prior to the meeting time.[3]

We then study 3-Majority (see Chapter 9), where every node samples three other nodes at random and changes its opinion to the majority among the samples, with ties broken arbitrarily in case all sampled colors are distinct. The analysis of 3-Majority rests on the shoulders of Voter: We show via Strassen's Theorem (Theorem 9.7) the existence of a coupling between the processes which allows to bound the progress of 3-Majority with the progress of Voter (Theorem 9.4). The latter allows, in the setting of many distinct opinions, a much better handle and notion of progress–by making use of the aforementioned "duality" with coalescing random walks. We also extend the well-known duality between the Voter and coalescing random walks to obtain bounds on the expected time required to reduce the number of opinions from n to k. This reduction to coalescing random walks (via theVoter) together with a potential approach allows us to derive the first unconditional bounds for the 3-Majority.

Subsequently, we consider the 2-Choices protocol (see Chapter 10) in which a node only changes its opinion if both of its samples share the same opinion. We obtain the first results for the case of more than 2 different opinions). We complete the picture by showing that 2-

[3]The meeting time of two nodes is the expected time for random walks starting from these two nodes to meet and the meeting time of a graph is the maximum over all pairs of nodes of the graph.

Choices is slower than 3-Majority but at the same time gives better guarantees for *plurality consensus*, meaning that the initially most frequent option prevails.

In the wake of our study of simple consensus dynamics we move to *consensus dynamics* which achieve better guarantees on the plurality consensus at the price of being slightly more sophisticated. First, we harness the guarantees of 2-Choices to develop a considerably faster algorithm (Chapter 11) to reach plurality consensus on the complete graph. Finally, in Chapter 12 we appeal to load balancing to design protocols achieving plurality consensus on general graphs.

Conclusion In essence, we relate a variety of seeming unrelated models to random walks and use powerful machinery developed in the past decades (such as Strassen's Theorem, Hajek's Theorem, Galton-Watson Trees, Póly urns, Doob-Martingales, *etc.*) to shed light on fundamental problems in distributed computing.

1.1 Introduction française

«Une grande partie de la vie, me semble-t-il, est déterminée par le pur hasard»
- Jean-Claude Pirotte

Cette thèse est consacrée à l'étude des processus stochastiques décentralisés qui sont omniprésents dans notre vie quotidienne. Ils décrivent par exemple le mouvement des atomes, les marchés boursiers, les personnes que nous rencontrons, les élections, etc. [4], et nous étudions dans cette thèse une large gamme d'exemples modélisés par des processus stochastiques : les réseaux sociaux, les processus de balles dans les bacs et les dynamiques et protocoles fondamentaux de consensus.

La particularité du travail réalisé dans cette thèse est de ramener systématiquement l'étude de ces processus à l'étude de marches aléatoires sur des ensembles de basse dimension (par exemple les entiers naturels), à l'aide d'une large palette d'outils de probabilités.

Première partie : processus dynamiques en temps infini. La première partie de cette thèse est consacrée aux processus dynamiques en temps infini. Nous commencerons par nous pencher sur le processus dynamiques de nouveaux utilisateurs sur les réseaux sociaux. En effet, l'étude des réseaux sociaux est primordiale dans la compréhension des sociétés dans lesquelles nous vivons - réelles et virtuelles. La communication et l'interaction avec autrui ont évolué : de visu (réel) vers les réseaux sociaux en ligne (virtuel). Les réseaux sociaux qui nous entourent vont bien au-delà des liens amicaux et incluent les liens commerciaux, les collaborations académiques, les relations amoureuses, voire l'échange d'idées politiques et la propagation des maladies. L'étude de ces réseaux sociaux est capitale dans divers domaines, y compris l'économie, les sciences sociales, le marketing, la propagation des maladies, la politique [Pik13, Jac+08, SR03, EGA+04, Jac05], etc.

Afin de comprendre les réseaux sociaux il est nécessaire de trouver un modèle qui présente les mêmes caractéristiques que ces dernieres. Dans cet esprit, de nombreux modèles ont été proposés (voir Chapter 4 pour une vue d'ensemble). Un modèle bien connu est le Forest Fire Process introduit par [LKF07]. Le modèle Forest Fire est un modèle pour les réseaux sociaux dans lequel arrivent perpétuellement de nouveaux utilisateurs : à chaque instant, un utilisateur arrive et se connecte à un autre utilisateur choisi uniformément au hasard parmi ceux déjà présents. Le nouvel utilisateur exécute alors un processus récursif simple pour déterminer ses connexions, c'est-à-dire son voisinage (voir Chapter 4 pour le modèle précis et Figure 1.4 pour une illustration).

[4] Bien que tous ces processus ne soient pas intrinsèquement stochastiques, les méthodes de prédiction basées sur des modèles déterministes nécessitent bien souvent des calculs infaisables en pratique car trop complexes. Une façon de remédier à cette complexité est les traiter de façon stochastique. Des prédictions précises par des modèles déterministes sont souvent infaisables face à la complexité des calculs. Le traitement de ces processus comme inhérents aléatoires permet de bonnes estimations, comme par example la prédiction du nombre de fois où un dé montre 6.

Il y a 10 ans, Leskovec et al. [LKF07] ont fait la conjecture suivante à partir des résultats de leurs simulations : Le Forest Fire Process présente « l'effet du petit monde », c'est-à-dire que la distance attendue entre n'importe quelle paire d'utilisateurs est constante. À partir des hypothèses faibles, nous avons pu prouver ceci. L'idée clé de la preuve est l'introduction d'une fonction potentielle qui nous permet de montrer que chaque fois que la distance d'un utilisateur dans le réseau à l'ensemble initial d'utilisateurs est grande, alors le potentiel diminuera en espérance au cours de l'arrivée des prochains *deux* utilisateurs. En particulier, le potentiel peut être modélisé comme une marche aléatoire générale sur les nombres naturels biaisée vers zéro (sauf pour un nombre finis d'états). En outre, le changement absolu par tour du potentiel est centré.

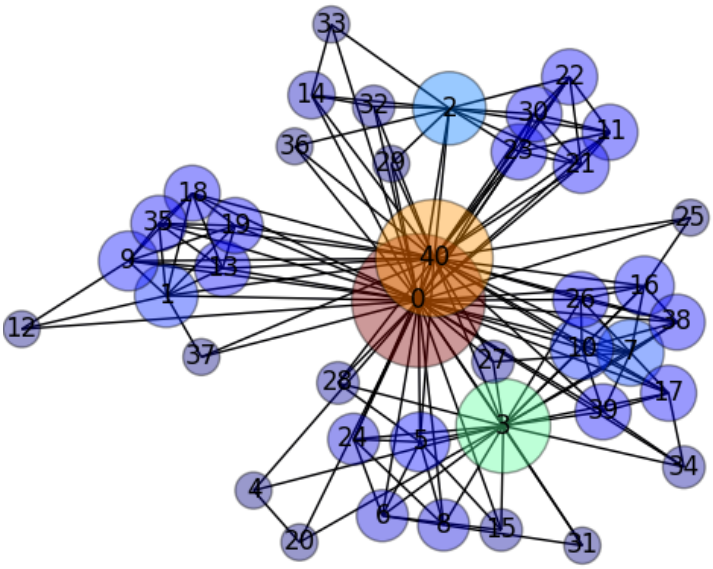


Figure 1.4: A réseau social généré par le forest fire processus.

Dans un second temps, nous consacrerons notre étude au processus dynamiques d'utilisateurs accédant aux serveurs d'un service web et aux problèmes de routage qui consistent à répartir efficacement ces utilisateurs entre les différents points d'accès aux serveurs. Ce problème est classiquement modélisé par un problème de balls-into-bins avec suppression (voir Chapter 5), processus fondamental en informatique distribuée, modélisant la charge de machines (ou processeurs) dans les systèmes et les files d'attentes pour accéder à ces services.

En pratique, il est essentiel de pouvoir prédire la longueur des files d'attente et, par conséquent, le temps que l'utilisateur ou la tâche passe dans la file afin de concevoir efficacement les infrastructures.

Dans Chapter 5, nous montrons en quoi ce problème peut être modélisé par un problème des balles arrivant de façon aléatoire dans différents bacs. Nous étudions la version des boules dans les bacs qui fonctionne comme suit. À chaque étape du temps, un lot de 0 à n boules arrive : chaque n boule potentielle apparaît avec probabilité $\lambda < 1$ et chacune des boules apparues choisit (I) uniformément au hasard à partir de n bins (Greedy[1]) ou (ii) de manière glouton parmi deux bacs échantillonnés uniformément au hasard à partir de n bins (Greedy[2]). À la fin de chaque tour, chacun des bacs non vide supprime une boule (voir Figure 1.5 pour une illustration).

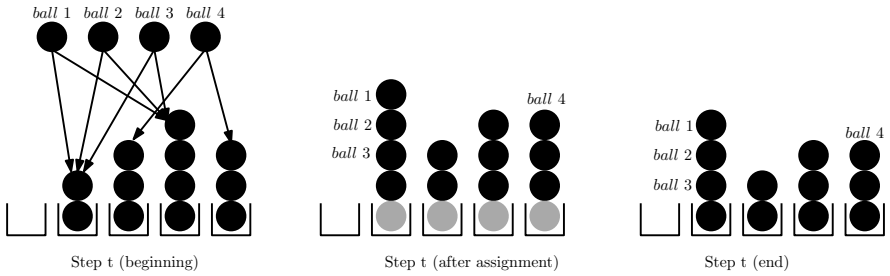


Figure 1.5: Un tour typique de GREEDY[2].

Nous attribuons des limites à la charge des bacs après un nombre arbitraire d'étapes (éventuellement super exponentielles dans le nombre de bacs n) et montrons une différence exponentielle dans la charge de processus GREEDY[1] et GREEDY[2]. Nous montrons que les chaînes de Markov correspondantes sont récurrentes positives et qu'il y a une différence exponentielle dans la charge, similaire à celle des processus classiques à deux choix de boules dans les bacs [ABKU99]. Les applications sont Hashing, les émulations de mémoire partagée sur les machines à mémoire distribuée, l'équilibrage de charge avec des informations limitées et le routage des circuits à faible congestion [MRS00].

Notre analyse s'appuie sur une analyse minutieuse de l'interaction entre différents potentiels que nous introduisons pour notre analyse. Chacun de ces potentiels représente les caractéristiques de la répartition de la charge, et l'évolution de ces potentiels s'apparente à une marche aléatoire générale avec une dérive. Afin d'obtenir des comportement asymptotiques, nous utilisons d'abord les limites de l'union d'une manière adaptative, puis à l'aide d'arguments combinatoires en d de l'approche générale (voir Forest Fire Process).

Deuxième partie : Dynamique de consensus et protocoles de consensus. La deuxième partie de la thèse est consacrée à l'étude de la dynamique et des protocoles de consensus à travers l'objectif des chaînes de Markov : les dynamiques de consensus sont des processus dans lesquels chaque noeud d'un graphe commence par une opinion [5] et tous les nœuds exécutent des protocoles simples dans le but d'atteindre un *consensus* rapidement, c'est-à-dire de s'entendre sur un seul avis. Les nœuds peuvent être utilisés dans un réseau comme par exemple dans le cas de Bitcoin ou chaque fois que deux ou plusieurs parties « découvrent » un nouveau bloc (regroupement de transactions) simultanément. Dans ce cas, tous les joueurs doivent se mettre d'accord sur l'un de ces blocs de manière distribuée. En fait, le consensus distribué est l'un des problèmes les plus fondamentaux en informatique distribuée avec de nombreuses applications dans l'informatique distribuée [DGM+11, Pel02, PVV09, CIG+15, BMPS04]. Le cas le plus important des protocoles de consensus est, sans doute, l'*élection de chef*, le cœur de l'informatique distribuée. Dans l'élection de chef, tous les nœuds commencent par des opinions distinctes et doivent s'entendre rapidement sur

[5] Nous supposons qu'il n'y a pas de ordre des opinions.

un seul avis. Toutes ces applications exigent généralement des protocoles très simples et économes en espace.

Les protocoles de consensus peuvent également être utilisés pour développer une meilleure compréhension de la façon dont les opinions et les croyances se diffusent dans les réseaux sociaux et autres applications des réseaux sociaux [MS10, MNT14]. En outre, le développement d'une simple dynamique de consensus aide à comprendre comment la communication dans la nature, par exemple parmi les fourmis et les oiseaux, fonctionne ainsi que d'innombrables autres aspects de la nature, [BDDS10, CER14, FPM+02, CDS+13, CC12].

Il est bien connu (*e. g.,* [Ang80, Lyn89]) que l'obtention d'un consensus de manière déterministe est impossible dans de nombreux contextes d'intérêts.

Une façon naturelle de défier les résultats impossibles est l'utilisation de protocoles randomisés - le noyau de cette thèse. La dynamique aléatoire prédominante dans cette zone est Voter, 2-Choices, et 3-Majority que nous présentons par la suite (voir Figure 1.6 pour une illustration).

Voter est sans doute la dynamique aléatoire la plus simple possible et 2-Choices et 3-Majority sont tout aussi simples et sont en même temps des solutions auto-stabilisantes efficaces pour l'*accord byzantin* [PSL80, Rab83] : parvenir à un consensus en présence d'un adversaire qui peut perturber un ensemble borné de noeuds à chaque tour [BCN+14b, BCN+16, CER14, EFK+16].

Nous généralisons chacun de ces protocoles de différentes manières et réglons la question du protocole le plus rapide parmi ces trois sur le graphique complet.

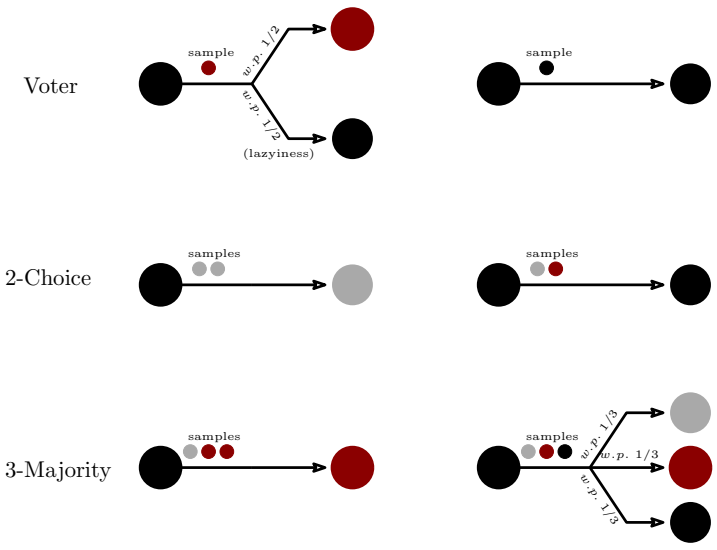


Figure 1.6: Illustration des dynamiques aléatoires.

Le modèle Voter (voir Chapter 8) fonctionne ainsi : initialement, chaque nœud a une opinion distincte et, dans chaque cycle synchrone, échantillonne un voisin uniformément au hasard et adopte son opinion. Il est bien connu que le temps de consensus suit la même distribution que le temps de coalescence, c'est-à-dire le temps qu'il faut pour n marches aléatoires à partir de différents nœuds pour s'annuler. Nous étudions donc le temps de consensus et l'exprimons en deux quantités de marches aléatoires : le temps de mélange et le temps de

réunion. En tant que produit secondaire, nous obtenons des limites de queue sur le temps de la réunion des marches aléatoires avant le temps de réunion. [6]

Nous étudions ensuite 3-Majority (voir Chapter 9), où chaque noeud échantillonne trois autres noeuds au hasard et change son opinion à la majorité parmi les échantillons, avec des liens cassés arbitrairement dans le cas où toutes les couleurs échantillonnées sont distinctes. L'analyse de 3-Majority repose sur les épaules de Voter : nous montrons via le théorème de Strassen l'existence d'un couplage entre les processus qui permettent de lier la progression de 3-Majority avec l'évolution de Voter. Ce dernier permet, dans le cadre de nombreuses opinions distinctes, une meilleure maîtrise et une notion de progrès, en utilisant la dualité susmentionnée avec des marches aléatoires coalescentes. Nous étendons également la dualité bien connue entre le temps de consensus et le temps de coalescence pour obtenir une borne sur le temps requis pour réduire de n opinions à k opinions. Cette réduction à Voter et donc aux marches aléatoires coalescentes avec une approche potentielle nous permet d'obtenir les premières limites inconditionnelles pour le 3-Majority.

Par la suite, nous considérons le protocole 2-Choices (voir Chapter 10) dans lequel un noeud échantillonne deux autres nœuds et ne change son avis que si les deux échantillons partage le même avis. Nous obtenons les premiers résultats pour le cas où plus de deux avis sont différents. De plus, nous complétons l'image globale en montrant que 2-Choices est plus lent que 3-Majority, mais en même temps donne de meilleures garanties pour *consensus de la pluralité*, ce qui signifie que l'opinion initialement majoritaire l'emporte.

À la suite de notre étude de la dynamique de consensus simple, nous nous déplaçons vers les *protocoles de consensus* qui permettent d'obtenir de meilleures garanties sur la pluralité de consensus au prix d'être plus sophistiqués. Tout d'abord, nous exploitons ces garanties de 2-Choices pour développer un algorithme beaucoup plus rapide (Chapter 11) pour atteindre un consensus de la pluralité sur le graph complet. Enfin, dans Chapter 12, nous faisons appel au domaine de la répartition de charge pour concevoir des protocoles obtenant un consensus de pluralité sur des graphes généraux.

[6] Le temps de réunion de deux nœuds est le temps prévue pour les marches aléatoires commençant sur les noeuds respectifs et le de temps réunion d'un graphe est le maximum sur toutes les paires de noeuds du graphe.

1.2 Organization and Publications

In the first part of this thesis (Part I) we analyze two infinite dynamic processes: The Forest Fire Process (Chapter 4) and a Balls-into-Bins version (Chapter 5). See Chapter 3 for an introduction and overview of our results. The second half of the thesis (Part II) considers consensus dynamics and protocols and an introduction and overview can be found in Chapter 7. In this part we study Voter (Chapter 8), 2-Choices (Chapter 10), 3-Majority (Chapter 9), and other protocols (Chapter 11 and Chapter 12). Chapter 2 introduces the bulk of the notation used in this thesis and Appendice A provides the probabilistic preliminaries of this thesis.

Publications comprised in this thesis

- Chapter 4 considers Forest Fire Process and is based on
  V. Kanade, R. Levi, Z. Lotker, F. Mallmann-Trenn, and C. Mathieu: *Distance in the Forest Fire Model How far are you from Eve?* In *Proceedings of the Twenty-Seventh Annual ACM-SIAM Symposium on Discrete Algorithms, SODA'16*, 2016, pages 1602–1620. URL: http://dx.adoi.org/10.1137/1.9781611974331.ch109

- Chapter 5 considers the Balls-into-bins process and is based on
  P. Berenbrink, T. Friedetzky, P. Kling, F. Mallmann-Trenn, L. Nagel, and C. Wastell: *Self-stabilizing Balls & Bins in Batches: The Power of Leaky Bins.* In *Proceedings of the 2016 ACM Symposium on Principles of Distributed Computing*. PODC'16, 2016, pages 83–92. URL: http://adoi.acm.org/10.1145/2933057.2933092

- Chapter 8 considers the Voter and is based on
  V. Kanade, F. Mallmann-Trenn, and T. Sauerwald: *On coalescence time in graphs-When is coalescing as fast as meeting?* In *CoRR*, volume abs/1611.02460, 2016. URL: http://arxiv.org/abs/1611.02460

- Chapter 9 considers the 3-Majority and is based on
  P. Berenbrink, A. E. F. Clementi, R. Elsässer, P. Kling, F. Mallmann-Trenn, and E. Natale: *Ignore or Comply?: On Breaking Symmetry in Consensus.* In *Proceedings of the ACM Symposium on Principles of Distributed Computing, PODC'17, Washington, DC, USA, July 25-27, 2017*, 2017, pages 335–344. URL: http://doi.acm.org/10.1145/3087801.3087817

- Chapter 10 considers the 2-Choices and is based on
  R. Elsässer, T. Friedetzky, D. Kaaser, F. Mallmann-Trenn, and H. Trinker: *Efficient k-Party Voting with Two Choices.* In *CoRR*, volume abs/1602.04667, 2016. URL: http://arxiv.org/abs/1602.04667 and the preceding publication

- Chapter 11 considers our asynchronous protocol and is based on the preceding publication
- Chapter 12 considers our consensus protocol inspired by load balancing and is based on

  P. Berenbrink, T. Friedetzky, P. Kling, F. Mallmann-Trenn, and C. Wastell: *Plurality Consensus in Arbitrary Graphs: Lessons Learned from Load Balancing.* In *24th Annual European Symposium on Algorithms ESA'16*. Volume 57. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, 2016, pages 10:1–10:18. URL: http://drops.dagstuhl.de/opus/volltexte/2016/6361

Publications of the authors not covered in the thesis

- V. Cohen-Addad, V. Kanade, and F. Mallmann-Trenn: *Hierarchical Clustering Beyond the Worst-Case.* In, 2017. To appear in NIPS 2017. URL: http://arxiv.org/abs/1702.03959
- P. Berenbrink, R. Klasing, A. Kosowski, F. Mallmann-Trenn, and P. Uznanski: *Improved Analysis of Deterministic Load-Balancing Schemes.* In *Proceedings of the 2015 ACM Symposium on Principles of Distributed Computing, PODC'15*, 2015, pages 301–310. URL: http://adoi.acm.org/10.1145/2767386.2767413
- P. Berenbrink, G. Giakkoupis, A. Kermarrec, and F. Mallmann-Trenn: *Bounds on the Voter Model in Dynamic Networks.* In *43rd International Colloquium on Automata, Languages, and Programming, ICALP'16*, 2016, pages 146:1–146:15. URL: http://dx.adoi.org/10.4230/LIPIcs.ICALP.2016.146
- V. Cohen-Addad, V. Kanade, F. Mallmann-Trenn, and C. Mathieu: *Hierarchical Clustering: Objective Functions and Algorithms.* In *CoRR*, volume abs/1704.02147, 2017. URL: http://arxiv.org/abs/1704.02147
- P. Berenbrink, T. Friedetzky, F. Mallmann-Trenn, S. Meshkinfamfard, and C. Wastell: *Threshold Load Balancing with Weighted Tasks.* In *2015 IEEE International Parallel and Distributed Processing Symposium, IPDPS'15*, 2015, pages 550–558. URL: http://dx.adoi.org/10.1109/IPDPS.2015.73
- P. Berenbrink, F. Ergün, F. Mallmann-Trenn, and E. S. Azer: *Palindrome Recognition In The Streaming Model.* In *31st International Symposium on Theoretical Aspects of Computer Science STACS*. Volume 25, 2014, pages 149–161. URL: http://drops.dagstuhl.de/opus/volltexte/2014/4454
- V. Kanade, F. Mallmann-Trenn, and V. Verdugo: *How large is your graph?* In *CoRR*, volume abs/1702.03959, 2017. To appear in DISC'17. URL: http://arxiv.org/abs/1702.03959

- D. Kaaser, F. Mallmann-Trenn, and E. Natale: *On the Voting Time of the Deterministic Majority Process*. In *41st International Symposium on Mathematical Foundations of Computer Science, MFCS'16*, 2016, pages 55:1–55:15. URL: http://dx.adoi.org/10.4230/LIPIcs.MFCS.2016.55

- P. Berenbrink, B. Krayenhoff, and F. Mallmann-Trenn: *Estimating the number of connected components in sublinear time*. In *Inf. Process. Lett.* volume 114 (11), 2014, pages 639–642. URL: http://dx.adoi.org/10.1016/j.ipl.2014.05.008

- P. Kling, A. Cord-Landwehr, and F. Mallmann-Trenn: *Slow Down and Sleep for Profit in Online Deadline Scheduling*. In *Design and Analysis of Algorithms - First Mediterranean Conference on Algorithms, MedAlg'12*, 2012, pages 234–247. URL: http://dx.adoi.org/10.1007/978-3-642-34862-4_17

- F. Mallmann-Trenn, C. Mathieu, and V. Verdugo: *Skyline Computation with Noisy Comparisons*. Unpublished manuscript

Chapter 2

General Notation

We denote by $\mathbb{N} = \{0, 1, \dots, \}$, $\mathbb{R}$ the set of all natural and real numbers, respectively. Let $[n] := \{1, 2, \dots, n\}$. We write log for the logarithm with base 2 and ln for the natural logarithm. Let $d \in \mathbb{N}$ and $\mathbf{x}, \mathbf{y} \in \mathbb{R}^d$. We define $\|\mathbf{x}\|_1 := \sum_{i\in[d]} x_i$ and $\|\mathbf{x}\|_2 := \left(\sum_{i\in[d]} x_i^2\right)^{1/2}$. Moreover, let $\mathbf{x}^{\downarrow}$ denote a permutation of $\mathbf{x}$ such that all components are sorted non-increasingly. We write $\mathbf{x} \succeq \mathbf{y}$ and say $\mathbf{x}$ *majorizes* $\mathbf{y}$ for $\|\mathbf{x}\|_1 = \|\mathbf{y}\|_1$ if, for all $l \in [d]$, we have $\sum_{i\in[l]} x_i^{\downarrow} \geq \sum_{i\in[l]} y_i^{\downarrow}$. In general, we will use bold-faced letters to denote vectors and capital letters to denote random variables and sets.

Graphs. Graphs $G = (V, E)$ considered in this thesis may be directed or undirected; typically we assume $|V| = n$ and $|E| = m$, though if there is scope for confusion we use $|V|$ or $|E|$ explicitly. For undirected graphs, for a node $v \in V$ we denote by $N(v)$ its *neighbourhood*, *i. e.,* $N(v) := \{w \mid \{v, w\} \in E\}$, and its degree by $\deg(v) := |N(v)|$. In the case of directed graphs, we denote by $N^+(v) := \{w \mid (v, w) \in E\}$ its *out-neighbourhood* and by $\deg^+(v) := |N^+(v)|$ its *out-degree*. Similarly, $N^-(v) := \{u \mid (u, v) \in E\}$ denotes its *in-neighbourhood* and $\deg^-(v) := |N^-(v)|$ its *in-degree*. Furthermore, $\mathrm{d}_{\mathrm{avg}} := \sum_{v\in V} \deg(v)/n$ denotes the average degree. We use $\mathrm{d}_{\max}$ and $\mathrm{d}_{\min}$ to refer the maximum and minimum degree. Whenever there is scope for confusion, we use the notations $\deg_G(u)$, $N_G(v)$, $\mathrm{d}_{\mathrm{avg}}(G)$, *etc.* to emphasize that the terms are with respect to graph G.

Random Variables. For random variables X and Y we write $X \leq^{\mathrm{st}} Y$ if X is stochastically dominated by Y, *i. e.,* for all $k \in \mathbb{R}$ it holds $\mathbb{P}[\, X \geq k \,] \leq \mathbb{P}[\, Y \geq k \,]$. We denote by $\mathcal{F}_t$ the filtration (*i. e.,*, intuitively speaking, the history of all random decisions) up to time step t. Throughout this thesis, the expression w.h.p. (*with high probability*) means with probability at least $1 - n^{-\Omega(1)}$ and the expression w.c.p. (*with constant probability*) means with probability > 0.

Probability Distributions. We denote by $\mathsf{Bernoulli}(p)$ the Bernoulli distribution with success probability p, by $\mathsf{Bin}(n,p)$ the Binomial distribution, with n independent trials, each having success probability p, by $\mathsf{Geom}(p)$ the geometric distribution with success probability p, *i. e.,* for $X \sim \mathsf{Geom}(p)$ we have $\mathbb{P}[\,X = i\,] = (1-p)^i p$ for $i = 0, 1, \ldots$, by $\mathsf{Poisson}(\lambda)$ the Poisson distribution with mean λ, and by $\mathsf{Uniform}(b)$ the uniform distribution on the elements $\{0, 1, 2, \ldots, b\}$. For a probability vector $\boldsymbol{\Theta} \in [0,1]^d$, we use $\mathsf{Mult}(m, \boldsymbol{\Theta})$ to denote the multinomial distribution for m trials and d categories (the i-th category having probability Θ_i).

Markov Chains. Unless stated otherwise, all random walks are assumed to be discrete-time (indexed by natural numbers) and lazy, *i. e.,* if P denotes the $n \times n$ transition matrix of the random walk, $p_{u,u} = \frac{1}{2}$, $p_{u,v} = \frac{1}{2\deg(u)}$ for any edge $(u,v) \in E$ and $p_{u,v} = 0$ otherwise. We define $p^t_{u,v}$ to be the probability that a random walk starting at $u \in V$ is at node $v \in V$ at time $t \in \mathbb{N}$. Furthermore, let $\mathbf{p}^t_{u,\cdot}$ be the probability distribution of the random walk after t time steps starting at u. By $\boldsymbol{\pi}$ we denote the *stationary distribution*, which satisfies, for undirected graphs, $\pi(u) = \frac{\deg(u)}{2m}$ for all $u \in V$.

Let $d(t) := \max_u \|\mathbf{p}^t_{u,\cdot} - \boldsymbol{\pi}\|_{\mathrm{TV}}$ and $\bar{d}(t) := \max_{u,v} \|\mathbf{p}^t_{u,\cdot} - p^t_{v,\cdot}\|_{\mathrm{TV}}$, where $\|\cdot\|_{\mathrm{TV}}$ denotes the total variation distance. Following Aldous and Fill [AF02], we define the *mixing time* to be $t_{\mathrm{mix}}(\varepsilon) := \min\{t \geq 0 : \bar{d}(t) \leq \varepsilon\}$ and for convenience we will write

$$t_{\mathrm{mix}} := t_{\mathrm{mix}}(1/e).$$

We define separation from stationarity at a given time step as follows: $s(t) := \min\{\varepsilon \colon p^t_{u,v} \geq (1-\varepsilon)\pi(v) \text{ for all } u, v \in V\}$. The definition ensures that $s(\cdot)$ is submultiplicative, so in particular, non-increasing [AF02], and we can define the *separation threshold time*

$$t_{\mathrm{sep}} := \min\{t \geq 0 : s(t) \leq e^{-1}\}$$

and, by [AF02, Lemma 4.11], $t_{\mathrm{sep}} \leq 4t_{\mathrm{mix}}$. We write $T_{\mathrm{hit}}(u,v)$ to denote the first time step $t \geq 0$ at which a random walk starting at u hits v. In particular, $T_{\mathrm{hit}}(u,u) = 0$. The *hitting time* $t_{\mathrm{hit}}(u,v) = \mathbb{E}[\,T_{\mathrm{hit}}(u,v)\,]$ of any pair of nodes $u, v \in V$ is the expected time required for a random walk starting at u to hit v. Thus, $t_{\mathrm{hit}}(u,v)$ is the expectation of $T_{\mathrm{hit}}(u,v)$. The hitting time of a graph

$$t_{\mathrm{hit}} := \max_{u,v} t_{\mathrm{hit}}(u,v)$$

is the maximum over all such pairs.

For $A \subseteq V$, we use $t_{\mathrm{hit}}(u, A)$, to denote the expected time required for a random walk starting to u to hit some node in the set A. Furthermore, we define $t_{\mathrm{hit}}(\pi, u) := \sum_{v \in V} t_{\mathrm{hit}}(v,u) \cdot \pi(v)$. Furthermore, we define $t_{\text{avg-hit}} := \sum_{u,v \in V} \pi(u) \cdot \pi(v) \cdot t_{\mathrm{hit}}(u,v)$.

For two random walks $(X_t)_{t\geq 0}$, $(Y_t)_{t\geq 0}$ starting at u and v let

$$t_{\text{meet}}(u,v) := \min\{t \geq 0 \colon X_t = Y_t\}$$

denote the expected *meeting time*, *i. e.,* the first time step at which both walks are on the same node. We write t^{π}_{meet} to denote the expected meeting time of two random walks starting at two independent samples from the stationary distribution. Finally, let $t_{\text{meet}} := \max_{u,v} t_{\text{meet}}(u,v)$ denoted the worst-case expected meeting time.

Consensus Processes. The processes are defined in Chapter 7 and we restrict ourselves to only introducing the important notation. In these processes we have n anonymous nodes connected by edges of a graph. Initially, each node supports one opinion from the set $[k] \coloneqq \{1,\dots,k\}$. We refer to these colors as $\mathcal{C}_1, \mathcal{C}_2, \dots, \mathcal{C}_k$. The system state after any round by an n-dimensional integral vector $\mathbf{c} = (c_i)_{i\in[n]} \in \mathbb{N}_0^n$ with $\sum_{i\in[n]} c_i = n$. Here, the i-th component $c_i \in \mathbb{N}_0$ corresponds to the number of nodes supporting opinion i. If $k < n$, then $c_i = 0$ for all $i \in \{k+1, k+2, \dots, n\}$.

Miscellaneous. A function $f\colon \mathbb{R}^d \to \mathbb{R}$ is *Schur-convex* if $\mathbf{x} \succeq \mathbf{y} \Rightarrow f(\mathbf{x}) \geq f(\mathbf{y})$.

Part I

Probabilistic Analysis of Distributed Dynamic Processes

Chapter 3

Contributions Dynamic Processes

The first part of the thesis concerns the analysis of infinite dynamic processes by means of carefully crafted potentials and the analysis of the potential through (general) random walks. We consider two processes: *Forest Fire Process* and *Balls-into-Bins with Deletions*.

Definition Forest Fire Process. The Forest Fire Process is a model for generating random graphs representing the evolution of social networks. The nodes in the generated graph represent the users and the each edges represents friendship between the adjacent nodes.

At every time step a user arrives and connects to another user chosen uniformly at random among those already present. The new user then executes a simple recursive process to determine their connections *i. e.,* their neighborhood. Formally, the Forest Fire Process is defined iteratively, starting from a seed graph G_0. Let $G_{t-1} = (V_{t-1}, E_{t-1})$ denote the graph at the end of round $t-1$. In round t, a new node u_t arrives, and chooses a node $\text{amb}(u_t) \in V_{t-1}$ uniformly at random, where we call the node $\text{amb}(u_t)$ the *ambassador* of the new node u_t. After selecting the ambassador, we *burn* the ambassador, meaning we add the edge $(u_t, \text{amb}(u_t))$ to the graph. The graph generation process then continues as follows. First choose a random subset of the edges of G_{t-1} as active edges: every edge (u, v) of G_{t-1} is active independently with probability $\min\{1, \frac{\alpha}{\deg^+(u)}\}$, where α is a parameter of the model and $\deg^+(u)$ is u's out-degree.

Secondly, add an edge to all vertices of G_{t-1}, reachable from $\text{amb}(u_t)$ by a path consisting of active edges. This construction of G_t can be obtained by executing Algorithm 1 and Algorithm 2.

We show, under mild assumptions, that if the parameter α is a large enough constant, then this models exhibits indeed the small world effect, *i. e.,* the expected distance between two users is constant. Conversely, if α is below some constant, then the expected distance is of order $\Omega(\log n)$.

Algorithm 1: Forest Fire Process (G_0)

for $t = 1, 2, \ldots$ **do**
  upon arrival of node u_t at time t:
    $\text{amb}(u_t) \leftarrow$ a node chosen u.a.r. from V_{t-1}
    $S \leftarrow \text{Burn}(G_{t-1}, \text{amb}(u_t))$
    $G_t \leftarrow (V_{t-1} \cup \{u_t\}, E_{t-1} \cup \{(u_t, w) : w \in S\})$

Algorithm 2: $\text{Burn}(G = (V, E), v)$ // Outputs a subset of V reachable from v

$H \leftarrow \emptyset$
for all $(w, x) \in E$ **do**
  with probability $\min\left\{1, \frac{\alpha}{\deg^+_{G_t}(w)}\right\}$
    $H \leftarrow H \cup \{(w, x)\}$
return $\{x \in V :$ there exists a directed path from v to x in $H\}$

Definition Balls-into-Bins with Deletions. The Balls-into-Bins with Deletions process (see Chapter 5) is a fundamental process modeling, among other things, the load distribution in distributed systems. The system consists of n bins and balls which arrive over time according to two strategies: GREEDY[1] and GREEDY[2]. The process works as follows.

At each time step, we generate up to n balls, each with probability $\lambda < 0$. Each of the spawned balls (i) chooses the target bin uniformly at random (GREEDY[1]) or (ii) chooses the target bin greedily among two bins chosen uniformly at random (GREEDY[2]). To model the load of real system more realistically, we extend the model by adding the following ingredient. At the end of each round, all non-empty bins delete one ball each modeling a finished tasks. In the following we state an algorithm summarizing the above.

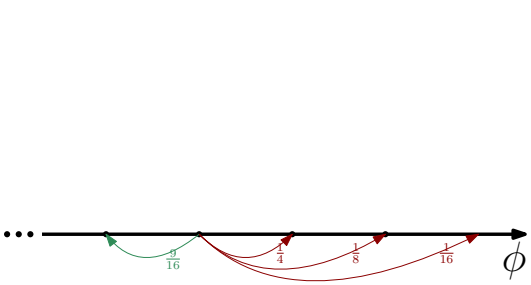
The algorithm is executed by each of the n generators.

Algorithm 3: GREEDY[d] for arrival rate λ, $d \in [1, 2]$

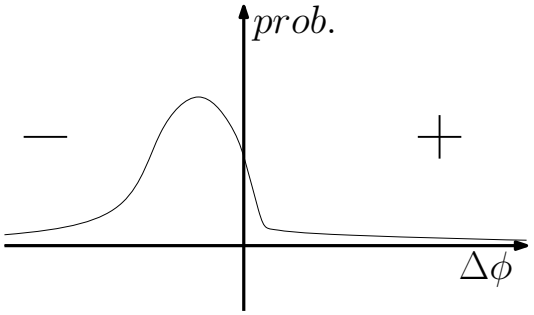
Spawn a ball w.p. $\lambda < 0$.
if a spawned is spaaned **then**
  for all choice $i \in [d]$ **do**
    $\text{SAMPLE}_i = \mathsf{Uniform}(\text{BIN}_1, \text{BIN}_2, \ldots, \text{BIN}_n)$
  Move to bin with smallest load among $\text{SAMPLE}_1, \text{SAMPLE}_2, \ldots, \text{SAMPLE}_d$

We show that both protocols are positive recurrent and that there is an exponential load difference between both protocols.

Analysis. At first glance, the Forest Fire Process and the Balls-into-Bins with Deletions process appear to be completely unrelated. The fabric connecting these processes is our analysis. We design custom-tailored potential functions for the Forest Fire Process, GREEDY[1], and GREEDY[2].



(a) A representation of the potential as a random walk: The arrow depict the transition probabilities of the walk for current node.



(b) The density function of the potential change.

Figure 3.1: The potential has a negative drift and the potential change exhibits a tail-bound.

Stating the precise definitions of the potentials would overstrain this overview (and is therefore left to Chapter 4 and Chapter 5) but their intuition is simple. We show that, whenever any of our potentials surpasses a certain threshold, then it decreases in expectation–regardless of the current state. Furthermore, the distribution of the potential change has an exponential tail-bound, *i. e.,* the probability to increase the potential by k is $2^{-\Omega(k)}$. This allows us to model the potential via a general random walk on the real numbers. See Figure 3.1 for an illustration.

It is worth mentioning that the potentials do not decrease (in expectation) in *every case*. In fact, when the potential is close to zero it increases in expectation.

Since both processes are infinite, it does happen–albeit very rarely–that the quantities of interest (*e. g.,* the maximum load) attain a value which is a function of the time t. Our potential approach shows that this is a "rare" state and whenever the system is in such a state, it quickly recovers.

The potentials are designed in such a way that the following holds. Whenever the potentials are large at a given time step t, we simply assume that system is in the worst-case state and show that the potentials decrease in expectation. This technique proves to be very useful for the problems we study since both problems are of infinite nature.

The analysis of such potentials dates back to Hajek [Haj82] (Theorem A.11) and less general versions have successfully applied to various areas and notably to evolutionary algorithms and drift theory [BFG03, DG13, PR99].

In order to harness the aforementioned general approach for the analysis of both problems, a few problem specific enhancement are required: For the Forest Fire Process we were unable to find a potential that decrease in expectation in a single time step, regardless of the current state. Instead, we consider the potential drop over two consecutive time steps: The node arriving in the first time step $t+1$ is likely to have a very "favorable" neighborhood, such that the node arriving at time step $t+2$ causes the potential to decrease. Our potential might be of general interest and refer the reader to Chapter 4 for an in-depth discussion.

As for the Balls-into-Bins process, we use three different potentials: Potential $\Phi(t)$ measuring the load difference to average load over time, potential $\Psi(t)$ counting the total load in the system, and potential $\Gamma(t)$ which interweaves both potentials and allows us use to prove positive recurrence. Because of the strong dependencies between the potential, we start by focusing solely on $\Phi(t)$ and we obtain bounds for arbitrary time steps (possibly super-exponential in the number of bins). This allows to apply the general approach described above and gives a weak bound on the maximum load at arbitrary time steps.

To derive a stronger bound we combine the bounds on Φ with combinatorial arguments. To do so use union bounds in an adaptive way to get rough bounds on the potentials at all time steps up to time step t. We then show that that there must have been a step $t - \text{poly}(n)$ where Ψ was very small. Together with our rough bounds on Φ, we are able to establish strong bounds on the maximum load at time t. Finally, we are able to analyze the third potential Γ by means of the general approach: We reduce it to a general random walk with drift towards zero whenever it's large and apply Hajek's Theorem. Using Γ we show positive recurrence of the underlying Markov chain.

Potential of the approach. It seems that the infinite nature (of the processes we considered) renders many standard approaches futile: for example, the approach of relying solely on invariants which could fail over the course of time. On the other hand, the concepts developed for Markov chains such as positive recurrence, and drift theory capture the notion of "recovery", making them very useful in the study of infinite processes. Using these techniques reduces the analysis to finding suitable potentials encapsulating the key-properties of the underlying problem. We are optimistic that this general potential approach, as well as our ideas used to craft and analyze our potentials, carry over to other dynamic processes.

Chapter 4

Social Networks: The Forest Fire Model [KLL+16]

Ten years ago, Leskovec, Kleinberg and Faloutsos introduced the Forest Fire model, a generative model to understand the dynamics of social networks over a long period [LKF07]. They examined real-world networks such as the ArXiv Citation Graph, the Patents Citation Graph, the Autonomous Systems Graph, Affiliation Graphs, the Email Network, the IMDB Actors-to-Movies Network, and a Product Recommendation Network. They observed that these social networks become denser over time. They also made the surprising observation that the effective diameter of the networks "shrinks" over time, instead of growing, as was previously thought. They suggested the *Forest Fire* model as an attempt to explain densification, shrinking diameter, and heavy-tailed distributions of vertex indegrees and outdegrees.

In this model, the evolution initially starts with a fixed seed graph. Time is discrete and at each time t a node u_t arrives, picks a random node, w, in the current graph as its "ambassador" and connects to it. The ambassador is considered *burned* and all other nodes are considered *unburnt*. Node u_t then generates two random numbers x and y and selects x outgoing edges from w and y in-coming edges to w incident to nodes that have not yet been burned. If not enough outgoing or incoming edges are available, u_t selects as many as it can. Let $w_1, w_2, ..., w_{x+y}$ denote the other endpoints of the edges selected. u_t connects to $w_1, w_2, ..., w_{x+y}$, marks them as burned, and then applies the previous step recursively to each w_i. See Figure 4.1 for an illustration. Leskovec *et al.* observed through simulation, that the Forest Fire Model appears to have the shrinking diameter property, but leave open the question of providing a rigorous proof:

> *"Rigorous analysis of the Forest Fire model appears to be quite difficult. However in simulations we find that [...] we can produce graphs that [...] have diameter that decrease."*

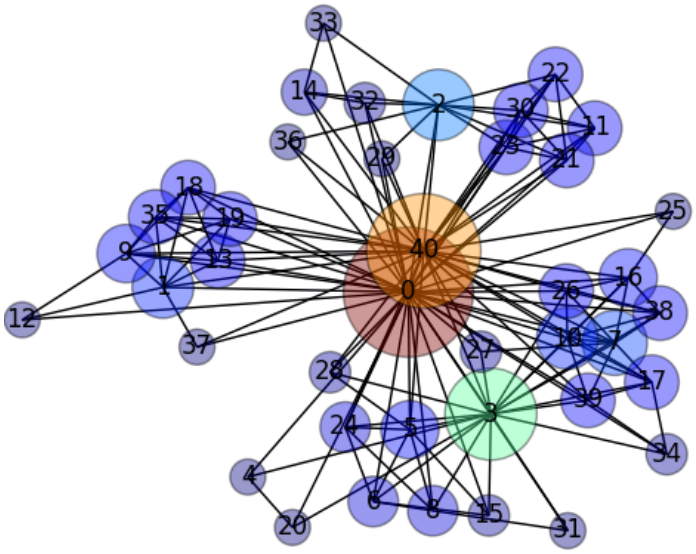


Figure 4.1: A social network generated by the Forest Fire Process. The node sizes and the colors are a function of the degrees and the node labels correspond to the arrival times.

Formal Definition

Formally, the Forest Fire Process is defined iteratively, starting from a seed graph G_0. Let $G_{t-1} = (V_{t-1}, E_{t-1})$ denote the graph at the end of round $t-1$. In round t, a new node u_t arrives, and chooses a node $\text{amb}(u_t) \in V_{t-1}$ uniformly at random, where we call the node $\text{amb}(u_t)$ the *ambassador* of the new node u_t. After selecting the ambassador, we *burn* the ambassador, *i. e.,* we add the edge $(u_t, \text{amb}(u_t))$ to the graph. This then propagates as follows.

First choose a random subset of the edges of G_{t-1} as active edges: every edge (u, v) of G_{t-1} is active independently with probability $\min\{1, \frac{\alpha}{\deg^+(u)}\}$, where α is a parameter of the model. Second, *burn* all vertices of G_{t-1}, reachable from $\text{amb}(u_t)$ by following directed active edges. Third, add an edge from u_t to every *burnt* vertex. This construction of G_t can be obtained by executing Algorithms 4 and 5. Although, it is more natural to view burning as a branching process in which we consider the burnt nodes "layer by layer", we describe the process as a percolation process[1] in order to avoid the need to define a specific order for the burning process: In a branching process, a node w could appear on several levels however we allow w to only be burnt once[2] and thus the order in which we burn nodes affects the random subtree of burnt nodes.

[1] A percolation process is a process in which every edge of the graph is present independently with a fixed probability.

[2] In Section 4.7 we discuss a model where we allow a node to burn several times

Algorithm 4: Forest Fire Process (G_0)

for $t = 1, 2, \ldots$ **do**
  upon arrival of node u_t at time t:
    $\text{amb}(u_t) \leftarrow$ a node chosen u.a.r. from V_{t-1}
    $S \leftarrow \text{Burn}(G_{t-1}, \text{amb}(u_t))$
    $G_t \leftarrow (V_{t-1} \cup \{u_t\}, E_{t-1} \cup \{(u_t, w) : w \in S\})$

Algorithm 5: $\text{Burn}(G = (V, E), v)$ // Outputs a subset of V reachable from v

$H \leftarrow \emptyset$
for all $(w, x) \in E$ **do**
  with probability $\min\left\{1, \frac{\alpha}{\deg^+_{G_t}(w)}\right\}$
    $H \leftarrow H \cup \{(w, x)\}$
return $\{x \in V :$ there exists a directed path from v to x in $H\}$

4.1 Results

We now state our two main results for the Forest Fire model. The parameters α and the input graph G_0 are fixed and we study the asymptotic properties of the graph G_t. We have not optimised the constants in the theorem statements and expect them to be far from being tight.

Theorem 4.1. *Let $\alpha \geq 100$ and let G_0 be a directed cycle such that $|G_0| \geq \alpha^{20}$, the Forest Fire Process with parameters α and G_0 has the property of non-increasing distance to G_0, i. e., for every t,*

$$\mathbb{E}[\,\text{dist}_{G_t}(u, G_0)\,] = O(1),$$

where the expectation is over a node u, which is chosen uniformly at random in G_t, and $\text{dist}(u, G_0)$ is the directed distance.[3]

Remark 4.2. *It is not critical that G_0 is a cycle. The main requirement is that conditioned on the* Burn Process *reaching G_0, a large enough constant number of vertices of G_0 will be burnt. For example, G_0 being an expander, clique, or a strongly connected graph with large girth suffices. Simulations seem to indicate that G_0 being a single node also result in a similar behaviour.*

Theorem 4.3. *Let $\alpha \leq 1/(4e)$ and let G_0 be an arbitrary graph, the Forest Fire Process with parameters α and G_0 is such that*

$$\mathbb{E}[\,\text{dist}_{G_t}(u, G_0)\,] = \Omega(\log t),$$

using the same notation as above.

[3]Note that $\text{dist}(v_t, G_0)$, once defined at time t, never changes

4.2 Approach and Technical Contributions

The main idea is as follows. We first reduce the process to the line process in which the node at time t connects to the node which arrived at time $t-1$ (see Section 4.4). In this line process we define a potential $\phi(v_t)$ which measures, intuitively speaking, the "typical path length" of node v_t arriving at time step t to the initial graph L_0. We defer the formal definition to Section 4.5. We would like to argue that no matter what happens up to time t, $\mathrm{dist}(v_{t+1}, L_0)$ is less than $\mathrm{dist}(v_t, L_0)$ in expectation whenever $\mathrm{dist}(v_t, L_0)$ is large enough. This does not seem to be possible when using distance directly; we can construct graphs where this is not true. However, these graphs are unlikely to arise under the Line Fire Process. Analysing ϕ instead gets around this issue. In fact, assuming $\phi(v_{2t}) > 2$, we show that $\phi(v_{2t+2}) - \phi(v_{2t})$ has negative expectation—irrespective of the history up to time $2t$. The potential is designed such that v_{2t+1} sets up a favourable situation such that v_{2t+2} is able to decrease the potential w.r.t. to the value $\phi((v_{2t})$. The crucial part is that $\phi(v_t)$ is designed such that it dominates $\mathrm{dist}(v_{t+1}, L_0)$ and thus assuming we can bound $\mathbb{E}[\,\phi(v)\,] = 0$, we get, by triangle inequality, that for u, v chosen uniformly at random

$$\mathbb{E}[\,\mathrm{dist}(u,v)\,] \leq \mathbb{E}[\,\mathrm{dist}(u,L_0)\,] + \mathbb{E}[\,\mathrm{dist}(v,L_0)\,] \leq \mathbb{E}[\,\phi(u)\,] + \mathbb{E}[\,\phi(v)\,] = O(1).$$

Note that even though the edges added are directed, we treat the graph as undirected when we consider the distance of nodes.

4.3 Related work

There is a extensive variety of models for generating graphs of social networks, each reproducing a subset of properties observed in real-world social networks. The first major line of research considers static graphs, where the number of nodes does not change over the course of time: For example, in *small-world* like models, there is a fixed underlying graph which is augmented by additional links between the vertices. Kleinberg proposed a particular random augmentation of links on the grid and proved that this gives rise to a decentralised greedy algorithms to find short paths among nodes [Kle01]. In a more recent paper, Chaintreau *et al.* proposed a different model, in which similar results are achieved, where the grid is augmented with links generated by random walks on the grid with occasional resets [CFL08].

Other static models focus mainly on reproducing both densification and small diameter simultaneously. One example is the model by Leskovec *et al.* which uses a matrix-operation, namely, the Kronecker product, to generate self-similar graphs recursively [LCKF05]. They reproduce a vast number of properties including heavy tails for the in- and out-degree distribution and small diameter. However, the deterministic nature of this model produces unrealistic features. To remedy this drawback, they propose the Stochastic Kronecker

Graph (SKG) model which has been very successful and is widely used in simulations. One disadvantage of SKG is that the adjustment of the parameters may have a huge influence on the properties of the resulting graphs. Recently, Seshadhri *et al.* [SPK12] showed that in fact the SKG model bears resemblance to a variant of the Chung and Lu model [CL03] which generalises classical random graph models. Here for any given collection of n weights $(w_1, w_2, \ldots, w_n)$, the probability of an edge (i, j) is given by $w_i w_j (\sum_k w_k)$.

Additionally, Pinar et al. [PKC13] introduce the Block Two-Level Erdős Rény (BTER) model, and demonstrate that it captures observable properties of many real-world social networks. Given a degree sequence, the model works in two stages: In the first stage the nodes of roughly the same degree are grouped into clusters and the edges in each cluster are generated by ER (Erdős Rény) graphs for a given using another input parameter. Finally, the "excess" edges of the node i, *i. e.,* the edges not yet used up by edges in the same cluster, are generated by randomly choosing two endpoints proportional to the excess edges of the nodes. Resulting self-loops and multi-edges are discarded.

The second major research line considers graph evolving over time where at each time step new vertices and edges are added to the evolving graph. Barabási *et al.* proposed the so called *preferential attachment model* in which new vertices attach preferentially to vertices with high degree, reproducing the power-law distribution over the in-degrees [BA99]. Building on preferential attachment, Cooper and Frieze propose a model in which exhibits a power-law of the degree as well as a shrinking diameter and densification; unfortunately, it involves many parameters [CF03]. Roughly speaking, the graph at time G_t is generated as follows. With some probability a new node is added with one or more edges to G_{t-1} and with the remaining probability an already existing vertex is selected and exta edges are added to it. Recently, Avin *et al.* extended the preferential attachment model to incorporate densification [ALNP15]: Similarly as in [CF03] either a new node arrives or new edges are added. In either case, the nodes are chosen according to preferential attachment. Krapivsky and Redner investigated the development of random networks as the attachment probability grows [KR01].

The authors of [KKR+99, KRR+00] consider an edge copying evolution in which, on arrival, a new vertex picks an existing node and copies a subset of its neighbours. Another model is the Community Guided Attachment model, in which there is a hierarchical backbone structure that determines the linkage probabilities [LKF07]. Lattanzi and Sivakumar generate random graphs according to an underlying affiliation network: Each node picks a random subset of affiliations and in each affiliation the nodes are connected as a clique (additionally, there is a process of preferential attachment) [LS09]. They show that this model exhibits shrinking diameter, densification, and a heavy-tailed degree distribution. Moreover, they connected the densification of the network to the non-linearity of the core. The recursive search model proposed by Vazquez is quite similar to the Forest-Fire model [Vaz01]. In the recursive search model, vertices are added to the graph one by one; when a new vertex

arrives it first connects to a random vertex and then recursively connects to a subset of its unvisited neighbours. The main difference is that in the Forest Fire model, the average number of neighbours visited out of the current node is constant, where as in the recursive search model this is a constant fraction. Thus, presence of high-degree nodes can make the two models quite different.

In the Random-Surfer Model (RSM), introduced by Blum *et al.* [BCR06], the nodes arrive one by one. Upon arrival, each node performs several random walks from random starting points and connects to the endpoints of the performed walks. Our Random Walk Process (RWP) share resemblance to the RSM. The main difference is that in the RWP a new node connects to all the visited nodes in the random walk (instead of just the endpoint). Chebolu and Melsted [CM08] proved that the RSM and the PageRank-based selection model, proposed by Pandurangan *et al.* [PRU06], are equivalent and also proved that the expected in-degree of vertices follows a powerlaw distribution. More recently, Mehrabian and Wormald [MW14] proved logarithmic upper bounds for the diameter in the RSM and the PageRank-based selection model as well as a logarithmic lower bound for a special case where the generated graph is a tree.

The only rigorous work thus far on the Forest Fire model is by Mehrabian [MW14] who provide a logarithmic upper bound to the diameter of the Forest Fire model as well as for other well known models, e.g., the copying model and the PageRank-based selection model.

4.4 Relating graph and line process

4.4.1 Line process

To prove the results for the Forest Fire, we study the related process which we call the *Line Fire Process* allowing us to reduce the graph process to a line process: When comparing the graph processes (Forest Fire Process), defined in Section 4.1, with the line process (Line Fire Process), the difference is that while in the graph process the first step is to select the ambassador at random; in the line process we skip this step, and force each new node to select the most recently added node as its ambassador, *i. e.,* in the line process the first step is deterministic and follows the line structure.

We state two corresponding technical lemmas for the Line process; in the next subsection, we state coupling lemmas to relate the processes and prove the results of Sec-

tion 4.1, using the corresponding related results on the line (whose proofs are deferred to later sections), together with the coupling.

Algorithm 6: Line Fire Process(L_0)

for $t = 1, 2, \ldots$ **do**
  upon arrival of node u_t at time t:
    $\text{amb}(u_t) \leftarrow u_{t-1}$ **(key difference)**
    $S \leftarrow \text{Burn}(L_{t-1}, \text{amb}(u_t))$
    $L_t \leftarrow (V_{t-1} \cup \{u_t\}, E_{t-1} \cup \{(u_t, w) : w \in S\})$

Next, we state the relevant lemmas for the Line Fire Process, that are used to prove the above theorems. The proofs of these lemmas are deferred to later sections.

Lemma 4.4. *Let $\alpha \geq 100$ and let L_0 be a directed cycle, such that $|L_0| \geq \alpha^{20}$. Then, the Line Fire Process has the property that*

$$\exists c, \exists \gamma < 1 \text{ s.t. } \forall t \ \forall j \ \mathbb{P}[\,\text{dist}_{L_t}(v_t, L_0) > j\,] < c\gamma^j.$$

To see why Theorem 4.1 follows from Lemma 4.4 observe that the subgraph of G_t by the vertices on the path from v_t to G_0 following edges to ambassadors has the same distribution as L_τ for some τ - This concept is formalized in the remainder of Section 4.4, which the reader might wish to skip.

Lemma 4.5. *There exists an $\alpha^* > 0$ such that the following holds. Let $\alpha \leq \alpha^*$ and let L_0 be an arbitrary graph. Then, the Line Fire Process with parameters α and L_0 is such that*

$$\mathbb{E}[\,\text{dist}_{L_t}(v_t, L_0)\,] = \Omega(t).$$

4.4.2 The ambassador graph

Definition 4.6. *The* ambassador graph *A_t is the subgraph of $G_t = (V_t, E_t)$, consisting of edges $(u, \text{amb}(u))$ induced by all nodes $u \notin G_0$. These edges are referred to as* ambassador edges.

A_t is a forest of directed trees, rooted at vertices of G_0. First, we observe the following fact.

Fact 4.7. *If (u, v) is an edge of $G_t \setminus G_0$, then there exists a path from u to v in A_t.*

To prove our lower bounds, we will use the following bound on the expected distance to the seed graph in the ambassador graph. The following lemma was originally proven in [Dev87, Theorem 10].

Lemma 4.8. *Let u be a vertex in the ambassador graph A_t chosen uniformly at random. Then*

$$\mathbb{E}[\,\text{dist}_{A_t}(u, G_0)\,] = \Theta(\log t)\ .$$

Proof. Let v_k denote the node which arrives at time k, where by convention the vertices of G_0 arrive at time 0. First we prove the upper bound

$$\mathbb{E}[\,\mathrm{dist}_{A_t}(u, G_0)\,] \le \frac{1}{t} \sum_{1 \le k \le t} \mathbb{E}[\,\mathrm{dist}_{A_t}(v_k, G_0)\,]\,. \tag{4.1}$$

Since $\mathrm{dist}_{A_t}(v_k, G_0)$ is at most k in the worst-case:

$$\mathbb{E}[\,\mathrm{dist}_{A_t}(v_k, G_0)\,] \le 2\log_2 k + k \cdot \mathbb{P}[\,\mathrm{dist}_{A_t}(v_k, G_0) > 2\log k\,]\,. \tag{4.2}$$

Recall that $\mathrm{dist}_{A_t}(v_k, G_0)$ is the length the path $v_k, \mathrm{amb}(v_k), \mathrm{amb}^2(v_k), \ldots$, until we reach G_0. Let X_i denote the arrival time of $\mathrm{amb}^{i-1}(v_k) = v_{k-i+1}$.

We have, by uniform choice of the ambassador of a node: $X_1 = k$ and thus $\mathbb{E}[\,X_i | X_{i-1}\,] \le X_{i-1}/2$, which implies $\mathbb{E}[\,X_i\,] \le \mathbb{E}[\,X_{i-1}\,]/2$. We deduce that $\mathbb{E}[\,X_i\,] \le k/2^{i-1}$ for all i and

$$\mathbb{E}[\,X_{2\log_2 k+1}\,] \le \frac{1}{k}.$$

Moreover, we have $X_{\mathrm{dist}_{A_t}(v_k, G_0)} = 0$. By Markov's inequality,

$$\begin{aligned} \mathbb{P}[\,\mathrm{dist}_{A_t}(v_k, G_0) > 2\log_2 k\,] &= \mathbb{P}[\,X_{2\log_2 k+1} \ge 1\,] \\ &\le \mathbb{E}[\,X_{2\log_2 k+1}\,] \le 1/k. \end{aligned} \tag{4.3}$$

Combining (4.1), (4.2), and (4.3), we obtain $\mathbb{E}[\,\mathrm{dist}_{A_t}(u, G_0)\,] = O(\log t)$, as desired. We now prove the lower bound on the expectation. Again, let v_k denote the node which arrives at time k and recall that the vertices of G_0 arrive at time 0. Consider the path p from v_k to G_0 in A_k and observe that arrival times are decreasing along p. For $i \le \log_2 k$ let Y_i denote the indicator variable of the event that some vertex of p has an arrival time in $(2^{i-1}, 2^i]$. Since those intervals are disjoint,

$$\mathbb{E}[\,\mathrm{dist}_{A_t}(v_k, G_0)\,] \ge \mathbb{E}\left[\sum_{i \le \log_2 k} Y_i\right]. \tag{4.4}$$

To analyze Y_i, let v denote the first vertex on p with arrival time in $[0, 2^i]$. By uniform choice of the ambassador and monotonicity

$$\mathbb{P}[\,Y_i = 1\,] = \mathbb{P}\Big[\,v \text{ has arrival time} > 2^{i-1}\,\Big] = \frac{2^i - 2^{i-1}}{2^i + |G_0|} \ge \frac{1}{2(1 + |G_0|)}.$$

Thus,

$$\mathbb{E}[\,\mathrm{dist}_{A_t}(v_k, G_0)\,] \ge \mathbb{E}\left[\sum_{i \le \log_2 k} Y_i\right] \ge \frac{\log_2 k}{2(|G_0| + 1)}. \tag{4.5}$$

□

4.4.3 Coupling

The following lemma shows the relation between the Line Fire and Forest Fire Processes.

Definition 4.9. *The* level *of a vertex u is its distance to G_0 in the ambassador graph, defined by:*

$$\ell(u) = \begin{cases} 0 & \text{if } u \in G_0 \\ \ell(\text{amb}(u)) + 1 & \text{otherwise.} \end{cases}$$

Lemma 4.10. *Let $t \geq \tau \geq 1$. Consider the Forest Fire Process with seed graph G_0, conditioned on $\ell(u_t) = \tau$. Then, the subgraph of G_t, consisting of G_0 and of all vertices on the path from u_t to G_0 in A_t, and of all edges out of those vertices; has the same distribution as the graph L_τ, with seed graph $L_0 = G_0$. In particular $\text{dist}_{G_t}(u_t, G_0)$ in the Forest Fire Process conditioned on $\ell(u_t) = \tau$ has the same distribution as $\text{dist}_{L_\tau}(v_\tau, G_0)$ in the Line Fire Process.*

Proof. The ambassador graph is constructed independently of the Burn Process (Algorithm 5), so we can change the order in which the edges of G_t are constructed, by generating the ambassador graph in a first phase, and then adding the other edges in a second phase. In the first phase, a node only chooses a random ambassador and connects to it. In the second phase, every node invokes the Burn Process starting with the respective ambassador.

Consider the path in the ambassador graph, going from u_t to G_0, and label its vertices

$$(u_t, \text{amb}(u_t), \text{amb}^2(u_t), \ldots, \text{amb}^{\ell(u_t)}(u_t)) = (w_{\ell(u_t)}, \ldots, w_0),$$

where amb^k denotes k iterative applications of $\text{amb}(\cdot)$.

Thus $w_{\ell(u_t)} = u_t$, $w_i = \text{amb}(w_{i+1})$ for $i < \ell(u_t)$, and $w_0 \in G_0$. We claim that the subgraph induced by $G_0 \cup \{w_0, \ldots, w_{\ell(u_t)}\}$ in the Forest Fire Process, has exactly the same distribution as the graph L_τ, produced by the Line Fire process with seed G_0, for $\tau = \ell(u_t)$.

To prove this, we couple the burning decisions of w_i and v_i. When $i = 0$, both graphs are G_0. Assume by induction that the subgraphs induced by $G_0 \cup \{w_0, \ldots, w_{i-1}\}$, in the Forest Fire Process and in the graph L_{i-1} in the Line Fire Process, are identically distributed, hence coupled. Then the Burn Process, starting at w_i, can clearly also be coupled with the Burn Process of vertex v_i, to give the desired result.

□

Corollary 4.11. *For every $b \geq 0$, we have*

$$\mathbb{P}[\,\text{dist}_{G_t}(u_t, G_0) > b \mid \ell(u_t) = \tau\,] = \mathbb{P}[\,\text{dist}_{L_\tau}(v_\tau, G_0) > b\,].$$

4.4.4 Proofs of the Graph results from the Line results - Proof of Theorem 4.1 and Theorem 4.3

Before proving Lemma 4.4, let us see how it implies Theorem 4.1.

Proof of Theorem 4.1. Let c be a large enough constant. Since $\mathrm{dist}(u_t, G_0)$, once defined at time t, never changes, it suffices to show that $\mathbb{E}[\,\mathrm{dist}_{G_t}(u_t, G_0)\,] = O(1)$. Thus, by law of total probability,

$$\begin{aligned}
\mathbb{E}[\,\mathrm{dist}_{G_t}(u_t, G_0)\,] &= \sum_{b=0}^{t} \mathbb{P}[\,\mathrm{dist}_{G_t}(u_t, G_0) > b\,] \\
&= \sum_{b=0}^{t}\sum_{\tau=1}^{t} \mathbb{P}[\,\mathrm{dist}_{G_t}(u_t, G_0) > b \mid \ell(u_t) = \tau\,] \cdot \mathbb{P}[\,\ell(u_t) = \tau\,] \\
&= \sum_{b=0}^{t}\sum_{\tau=1}^{t} \mathbb{P}[\,\mathrm{dist}_{L_\tau}(v_\tau, G_0) > b\,] \cdot \mathbb{P}[\,\ell(u_t) = \tau\,] \\
&= \sum_{\tau=1}^{t} \mathbb{P}[\,\ell(u_t) = \tau\,] \sum_{b=0}^{t} \mathbb{P}[\,\mathrm{dist}_{L_\tau}(v_\tau, G_0) > b\,]. \qquad (4.6)
\end{aligned}$$

From Lemma 4.4 for the Line Fire Process, $\mathbb{P}[\,\mathrm{dist}_{L_\tau}(v_\tau, G_0) > b\,] \le c\gamma^b$. Thus

$$\sum_{b=0}^{t} \mathbb{P}[\,\mathrm{dist}_{L_\tau}(v_\tau, G_0) > b\,] \le \sum_{b=0}^{t} c\gamma^b \le \frac{c}{1-\gamma} = O(1).$$

Since $\sum_{\tau=1}^{t} \mathbb{P}[\,\ell(u_t) = \tau\,] = 1$, the result follows. □

Proofs of Theorem 4.3. We follow the proof of Theorem 4.1 until (4.6) and get

$$\begin{aligned}
\mathbb{E}[\,\mathrm{dist}_{G_t}(u_t, G_0)\,] &= \sum_{\tau=1}^{t} \mathbb{P}[\,\ell(u_t) = \tau\,] \sum_{b=0}^{t} \mathbb{P}[\,\mathrm{dist}_{L_\tau}(v_\tau, G_0) > b\,] \\
&= \sum_{\tau=1}^{t} \mathbb{P}[\,\ell(u_t) = \tau\,] \mathbb{E}[\,\mathrm{dist}_{L_\tau}(v_\tau, G_0)\,].
\end{aligned}$$

Using Lemma 4.5, we get $\mathbb{E}[\,\mathrm{dist}_{L_\tau}(v_\tau, G_0)\,] \ge c\tau$ for a suitable constant c. Thus,

$$\begin{aligned}
\mathbb{E}[\,\mathrm{dist}_{G_t}(u_t, G_0)\,] &\ge \sum_{\tau=1}^{t} \mathbb{P}[\,\ell(u_t) = \tau\,] c \cdot \tau \\
&= c \cdot \mathbb{E}[\,\ell(u_t)\,] = \Omega(\log t),
\end{aligned}$$

by Lemma 4.8. Hence, $\mathbb{E}[\,\mathrm{dist}_{G_t}(u_{t'}, G_0)\,] = \Omega(\log t)$ for every $t' \ge t/2$, hence we obtain the desired result.

□

4.5 Analysis of the Line Fire Process

Throughout Sections 4.5.1, 4.5.2, 4.5.3, and 4.5.4 we assume that $\alpha \geq 100$, L_0 is a directed cycle with $|L_0| \geq \alpha^{20}$. In order to prove Lemma 4.4, we define a function ϕ, such that for all t, $\text{dist}_{L_t}(v_t) \leq \phi(v_t)$, and which is more amenable to analysis. The function ϕ is defined as follows: Let $\delta := \alpha^{20}$, then

$$\phi(v) = \begin{cases} 0 & \text{if } v \in L_0 \\ \max\limits_{w \in N^+(v)} \{\phi(w)\} + 1 & \text{if } \deg^+(v) < \delta \\ \max\left\{ \phi(\text{amb}(v)) - 2, \max\limits_{\substack{w \in N^+(v) \\ w \neq \text{amb}(v)}} \phi(w) + 1 \right\} & \text{otherwise.} \end{cases}$$

We now give an intuitive description and we defer the reader to Section 4.5.1 and Section 4.5.5 for an in-depth discussion about the ingredients of ϕ.

4.5.1 High-Level Proof Overview

We first give some intuition about the definition of ϕ. We would like to argue that no matter what happens up to time t, $\text{dist}(v_{t+1}, L_0)$ is less than $\text{dist}(v_t, L_0)$ in expectation whenever $\text{dist}(v_t, L_0)$ is large enough. This does not seem to be possible when using distance directly; we can construct graphs where this is not true. However, these graphs are unlikely to arise under the Line Fire Process. Analysing ϕ instead gets around this issue. In fact, assuming $\phi(v_{2t}) > 2$, we show that $\phi(v_{2t+2}) - \phi(v_{2t})$ has negative expectation — irrespective of the history up to time $2t$. A low value $\phi(v_t)$ implies that not only is there one short path from v_t to L_0, but most paths from v_t to L_0 are short. However, note that not all paths are short, in particular the path of ambassador edges $v_t, v_{t-1}, \ldots, v_0$ is of linear size.

Furthermore, while it is true for most nodes, it is not necessarily true that all nodes are well connected to the seed graph. Note that the definition of ϕ makes a special case for the ambassador when the degree is small. For an edge (v, u) if $u \neq \text{amb}(v)$, $\phi(u) < \phi(v)$. We will call edges $(v, \text{amb}(u))$ *ambassador edges*.

We start from an arbitrary history (and hence an arbitrary graph) at time $2t$. (See Figure 4.2a: the nodes are arranged by their ϕ value, ambassador edges are marked purple and may point upwards, *i. e.,* an increase in ϕ-value, all other edges point strictly downward, *i. e.,* a decrease in ϕ-value.) The good event at time $2t + 1$ involves two things: (i) the degree of v_{2t+1} is at least δ (ii) All neighbours v of v_{2t+1}, except possibly the ambassador, are such that $\phi(v) \leq \phi(v_{2t}) - 2$. We give a high-level idea why this is likely (formal proof in Lemma 4.16). The Burn Process stops at any node only with probability $\approx (1 - \alpha/d)^d \approx e^{-\alpha}$, thus it is quite likely that at least $\delta = \alpha^{20}$ nodes are burnt starting at v_{2t}. For the second part, at the very first stage, *i. e.,* neighbours of v_{2t} that are burnt, almost all neighbours

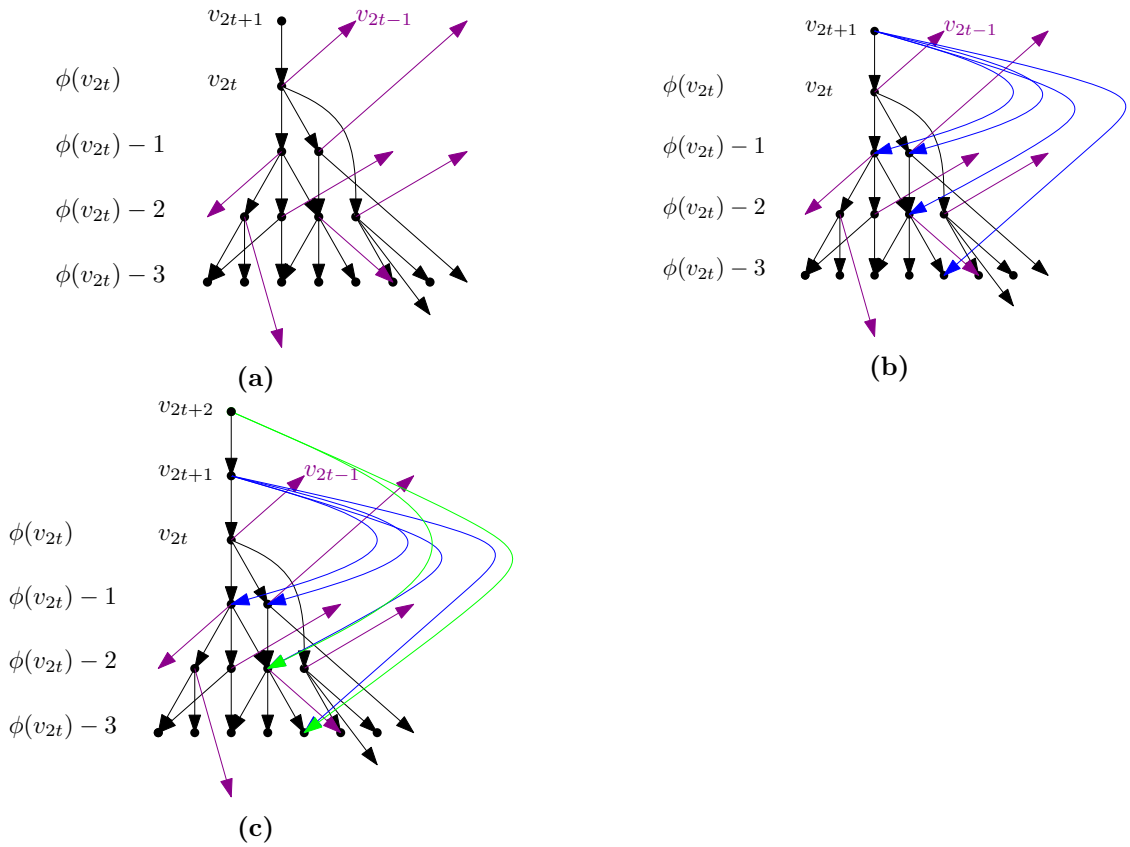


Figure 4.2: The purple edges are ambassador edges. The blue edges are the neighbours of v_{2t+1} and the green edges are the neighbours of v_{2t+2}. When v_{2t+1} arrives it is likely to only have very few edges to nodes with ϕ-values $\geq \phi(v_{2t}) - 1$. When v_{2t} arrives it is likely not to have any neighbours (except of the ambassador) with ϕ-values $\leq \phi(v_{2t}) - 2$.

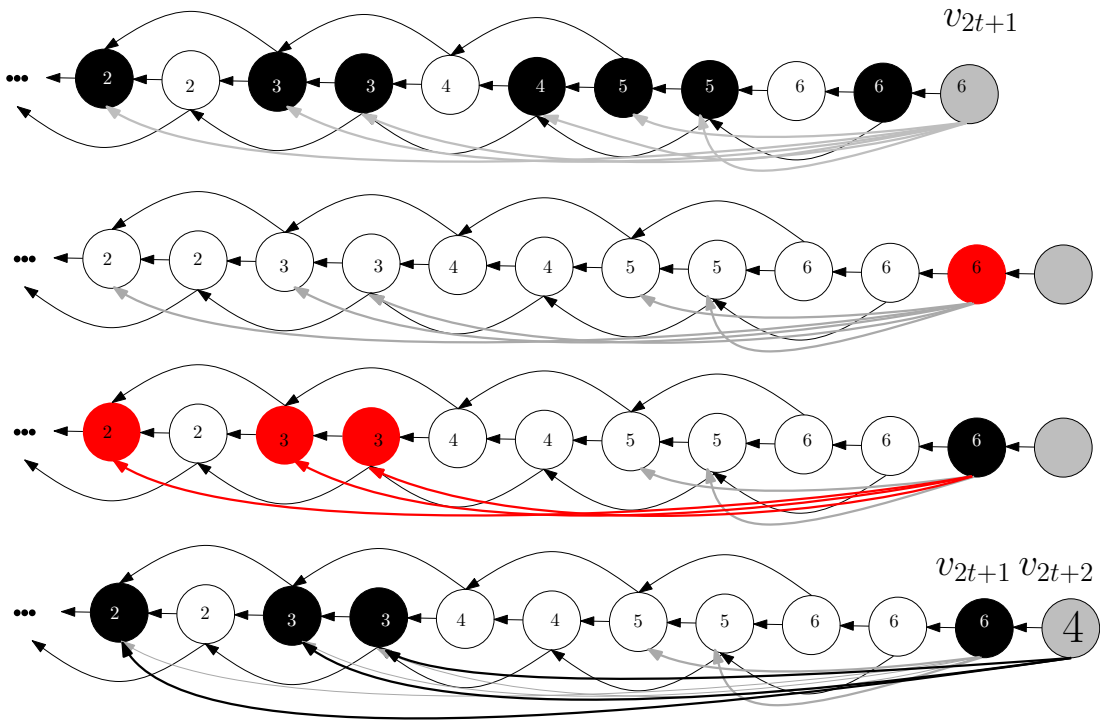


Figure 4.3: The figure depicts a typical situation for the case that $\phi(v_{2t})$ is large: When v_{2t+1} arrives it is likely to only have very few edges to nodes with ϕ-values $\geq \phi(v_{2t}) - 1$. At the arrival of v_{2t+2}, its neighbourhood (red edges) is determined by the Line Fire process. Due to the aforementioned structure of v_{2t+1}, it is likely that v_{2t+2} will not have any neighbours (except of its ambassador) with ϕ-values $\leq \phi(v_{2t})-2$.

(except possibly the ambassador if the v_{2t} has out-degree at least δ), will cause a drop in ϕ-value of at least 1. Subsequently, if we look at any path in the Burn Process, every edge traversed implies that the ϕ value dropped by at least 1, except if the edge was an ambassador (purple) edge at a high-degree node, where it may increase by 2 (see definition of ϕ). A large fraction of such purple edges are not likely to appear on any path (ambassador edges of low-degree nodes are not a problem, by definition of ϕ). The edges burnt when v_{2t+1} arrives are shown in Figure 4.2b.

Given that the good event at time $2t+1$ happens, the good event at time $2t+2$ again involves two things: (i) the degree of v_{2t+2} is at least δ (ii) *all* neighbours v of v_{2t+2} satisfy $\phi(v) \leq \phi(v_{2t}) - 2$. First, it is easily checked that if the good event happens, indeed it is the case that $\phi(v_{2t+2}) \leq \phi(v_{2t}) - 1$, *i. e.,* a decrease. The first part of the good event is similar to the previous case. For the second part, we again consider the first step of the Burn Process, *i. e.,* the burnt neighbours of v_{2t+1}. Since most neighbours of v_{2t+1} have ϕ-value at most $\phi(v_{2t}) - 2$, with high probability *all* burnt neighbours will satisfy this. Further down in the burn process, it is unlikely that ϕ-value increases as argued earlier, since a large fraction of ambassador (purple) edges would have to be followed which is unlikely (See Figure 4.2c).

Formally we can show that given any history up to time $2t$, $\phi(v_{2t+2}) - \phi(v_{2t})$ has a sub-exponential tail and negative expectation, which implies by Hajek's theorem (Theorem A.11) that $\mathbb{E}[\,\phi(v_{2t})\,] = O(1)$.

Figure 4.3 offers a slightly different point of view: Here we illustrate the essence of ϕ by focusing on the Line Fire process as opposed to focusing on the branching nature of the process.

4.5.2 Proof of Lemma 4.4

We now formalise the high-level ideas presented in the previous section. We begin by proving that ϕ indeed dominates the distance.

Fact 4.12. *If v arrives at time t, then*

$$\mathrm{dist}_{L_t}(v, L_0) \le \phi(v).$$

Proof. The straightforward proof is by induction on t. For $t = 0$, $v \in L_0$ and then $\mathrm{dist}_{L_t}(v, L_0) = 0 = \phi(0)$, so the statement holds.

Assume the statement holds for all nodes in L_{t-1}. Note that in the graph L_t all (directed) edges point to vertices that arrived earlier (*i. e.*, for any edge (v_t, v_τ), $t > \tau$). We get, by applying the induction hypothesis,

$$\begin{aligned}
\mathrm{dist}_{L_t}(v, L_0) &= \begin{cases} 0 & \text{if } v \in L_0 \\ \min\limits_{w \in N^+(v)} \{\mathrm{dist}_{L_t}(w, L_0)\} + 1 & \text{otherwise} \end{cases} \\
&\overset{ind.}{\le} \begin{cases} 0 & \text{if } v \in L_0 \\ \min\limits_{w \in N^+(v)} \{\phi(w)\} + 1 & \text{otherwise} \end{cases} \\
&\le \begin{cases} 0 & \text{if } v \in L_0 \\ \phi(\mathrm{amb}(v)) + 1 & \text{if } \deg^+(v) = 1 \\ \min\limits_{w \in N^+(v) \setminus \{\mathrm{amb}(v)\}} \{\phi(w)\} + 1 & \text{otherwise} \end{cases} \\
&\le \begin{cases} 0 & \text{if } v \in L_0 \\ \max\limits_{w \in N^+(v)} \{\phi(w)\} + 1 & \text{if } \deg^+(v) < \delta \\ \max\left\{ \phi(\mathrm{amb}(v)) - 2, \max\limits_{\substack{w \in N^+(v) \\ w \neq \mathrm{amb}(v)}} \phi(w) + 1 \right\} & \text{otherwise.} \end{cases} \\
&= \phi(v).
\end{aligned}$$

□

The proof of Lemma 4.4 relies on Hajek's theorem (Theorem A.11), which we can be found in Section A.2.4.

Let $(\mathcal{F}_t)_{t\geq 0}$ denote the history of random choices up to time t for the Line Fire process. We state the lemma that prove that $(\phi(v_{2t+2}) - \phi(v_{2t}) \mid \mathcal{F}_{2t})$ satisfies the conditions of Hajek's theorem. The proofs of this appears in subsequent subsections.

Lemma 4.13 (Majorization and Negative bias)**.** *The following holds.*

1. *Let Z be the random variable taking values over all integers greater than or equal to 4, defined by: $\mathbb{P}[\,Z = i\,] = 3^{-i/2}$ for $i \geq 5$, and $\mathbb{P}[\,Z = 4\,] = 1 - \frac{1}{9(\sqrt{3}+1)}$. Then*

$$(|\phi(v_{2t+2}) - \phi(v_{2t})| \mid \mathcal{F}_{2t}) \leq^{st} Z.$$

2. *Let α and κ be large enough constants and assume $|L_0| \geq \delta \geq \alpha^{\kappa}$. There exists a constant $\varepsilon_0 > 0$ such that for every t we have,*

$$\mathbb{E}[\,\phi(v_{2t+2}) - \phi(v_{2t}) \mid \mathcal{F}_{2t},\ \phi(v_{2t}) > 2\,] \leq -\varepsilon_0$$

We now prove Lemma 4.4.

Proof of Lemma 4.4. Let $Y_t = \phi(v_{2t})$. For any constant $\lambda' < \ln(\sqrt{3})/2$, we have,

$$\mathbb{E}\Big[\, e^{\lambda' Z} \,\Big] \leq e^{\lambda' 4} \cdot 1 + \sum_{i\geq 3} e^{\lambda' 2i} \cdot 3^{-i/2} = O(1),$$

where the last equation follows from the geometric series. Thus $\mathbb{E}\Big[\, e^{\lambda' Z} \,\Big]$ is finite for the random variable Z defined in the statement of Lemma 4.13 and hence the sequence $(Y_t)_{t\geq 0}$ with respect to the filtration $(\mathcal{F}_{2t})_{t\geq 0}$ satisfies the two conditions of Theorem A.11 by Lemma 4.13, hence, by using Fact 4.12, Lemma 4.4 follows. □

4.5.3 Proof of Lemma 4.13

In our proofs, it is useful to rephrase the process Burn(G, α) defined in Algorithm 5 as a tree process, rather than a percolation process.We define BurnBFS(G, v) in Algorithm 7. We assume that vertices have a natural order in the graph, for examples for graphs evolving in time, the vertices are ordered according to their time of arrival. Thus, when indexing a set we assume that the vertices are indexed in this order.

First, we note that if the burning decisions made for activation of edges (w, x) in Algorithm 7 are coupled with those made in Algorithm 5, the set of vertices returned by the two processes is exactly the same. Thus, this is indeed another view of the Burn Process. The Burn Process BurnBFS produces a tree T with activated vertices $\cup_{j\geq 0} M_j$, and edges (w, x) for which the *if* condition in Algorithm 7 was satisfied (see Figure 4.4). (We remark

Algorithm 7: BurnBFS(G, v)

$M_0 := \{v\}$
for $i = 1, 2, \ldots$ **do**
  $M_i := \emptyset$
  for all $w \in M_{i-1}$ **do**
    for all edges (w, x) **do**
      activate edge (w, x) with probability $\min\{1, \frac{\alpha}{\deg^+_G(w)}\}$
      if (w, x) is activated and $x \notin \bigcup_{j \le i} M_j$ **then**
        add x to M_i
        set parent$(x) := w$
return $\bigcup_{j \ge 0} M_j$

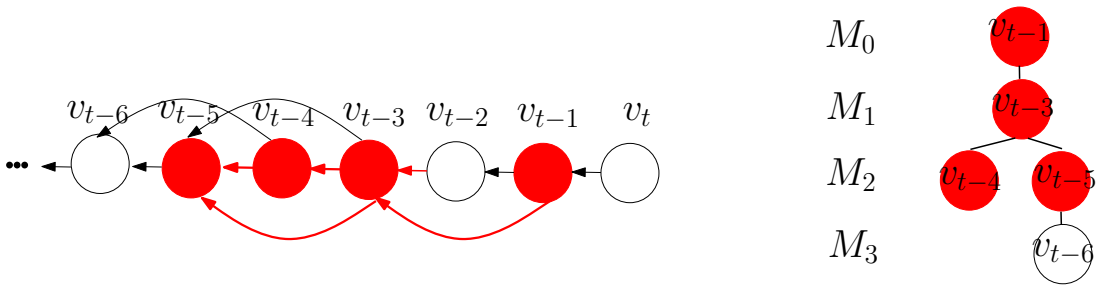


Figure 4.4: The figure depicts the percolation Burn Process (Algorithm 5) on the l.h.s. and the corresponding BFS burn process (BurnBFS(G, v) in Algorithm 7) on the r.h.s. Activated edges and burnt nodes are coloured red. In this example v_{t-5} (level M_2) is burnt by two predecessors one on level M_1 and one on level M_2; due to the definition of the BFS burn process, it is placed on M_2 (rather than M_3).

that if H_v is the induced subgraph of H (defined in Algorithm 5) consisting of all nodes reachable from v, then T is simply the unique BFS tree of H_v starting at v using the order on the vertices.)

In the present section, we will fix some graph G (say some L_t produced by the Line Fire process), and look at calls made to BurnBFS with this graph as input. Thus, the only source of randomness is the activation decision of the edges.

In order to apply Hajek's Theorem to ϕ, the main idea of the following lemma is roughly the following: Given u, let T_u denote the nodes burned by starting at u. Then, with probability at least $1 - \Omega(3^{-k})$, we have that all nodes burnt in T_u (and thus the new neighbourhood) have a ϕ-value of at most $\phi(u) + k$ which establishes a tail bound. Proposition 4.14 is proved by coupling the burning process with a Galton-Watson process (See Section A.5.1). On the Galton-Watson tree, we can finally analyze a simpler function that majorizes ϕ.

Proposition 4.14. *Let u be a vertex burnt by BurnBFS and T_u be the random subtree of T with root u. Then, for any $k \geq 0$,*

$$\mathbb{P}[\textit{for all } w \in T_u, \phi(w) \leq \phi(u) + k \mid \mathcal{H}] \geq 1 - \frac{3^{-k}}{12\alpha^4},$$

where, if i is s. t. $u \in M_i$ in BurnBFS(G, v), then $\mathcal{H}$ denotes the history of all activation decisions made by BurnBFS until all nodes belonging to M_i are added.

The proof can be found in Section 4.5.4. We continue by proving the following side lemma.

Lemma 4.15. $\mathbb{P}\Big[\deg^+_{L_t}(v_t) < \delta \mid \mathcal{F}_{t-1}\Big] \leq \frac{1}{2\alpha}$.

Proof. Observe that $\mathbb{P}[\,\mathsf{Bin}(d, \min\{1, \frac{\alpha}{d}\}) = 0\,] \leq e^{-\alpha}$. Hence, for any vertex, with probability at least $1 - e^{-\alpha}$, one or more of its outgoing edges are burned. Starting from $\mathrm{amb}(v_t) = v_{t-1}$, we do the following.

$w \leftarrow v_{t-1}$
repeat
  if at least one outgoing edge of w is burned **then**
    pick one such edges (w, x)
    $w \leftarrow x$
until w has no outgoing edges that is burned

All the vertices traversed by this process are neighbours of v_t. Note that, by assumption, the only cycles are in L_0 and we assume, that L_0 is a cycle of length $\geq \delta$. Thus, since $\delta = \alpha^{20}$ and α large enough we get,

$$\mathbb{P}\Big[\deg^+_{L_t}(v_t) \geq \delta \mid \mathcal{F}_{t-1}\Big] \geq (1 - e^{-\alpha})^{\delta - 1} \geq 1 - (\delta - 1)e^{-\alpha} \geq 1 - \frac{1}{2\alpha}.$$

This completes the proof. □

To prove the negative bias, we need to analyse the process over two consecutive steps. We start from an arbitrary history $\mathcal{F}_{2t}$. We first establish some properties that after one step hold with high probability (w.r.t. α).

Lemma 4.16. *Fix $\mathcal{F}_{2t}$ and consider the arrival of v_{2t+1}. With probability at least $1 - 1/\alpha$ the following holds: node v_{2t+1} has outdegree at least δ, and among the nodes of $N^+(v_{2t+1})$ only v_{2t} has value $\phi(v_{2t})$, at most 6α nodes have value $\phi(v_{2t}) - 1$, and all other nodes have value $\leq \phi(v_{2t}) - 2$.*

Proof. Let $z = \phi(v_{2t})$. Consider the process $\mathrm{BurnBFS}(G, v_{2t})$ executed to construct $N^+(v_{2t+1})$. The number of activated edges $|M_1|$ at v_{2t} is distributed

$$|M_1| \sim \mathsf{Bin}\Big(|\mathcal{N}(v_{2t})|, \min\Big\{1, \tfrac{\alpha}{|\mathcal{N}(v_{2t})|}\Big\}\Big) \leq^{\mathrm{st}} \mathsf{Bin}\Big(|\mathcal{N}(v)|, \tfrac{\alpha}{|\mathcal{N}(v)|}\Big)$$

and thus $\mathbb{E}[\,|M_1|\,] \leq \alpha$. By Chernoff bounds (Proposition A.3), with probability at least $1 - 2^{-6\alpha}$ we have $|M_1| \leq 6\alpha$. Assume this holds.

We consider now the out-degree of v_{2t}. If it is less than δ, then, by definition of ϕ, all nodes of $N^+(v_{2t})$, and in particular all elements of M_1, have ϕ-value at most $z - 1$. If it is greater than or equal to δ, then by definition of ϕ all but one node (v_{2t-1}) of $N^+(v_{2t})$ have value less than or equal to $z - 1$. In other words, in the former case all nodes have a small ϕ-value and in the latter case but one node have a low ϕ-value. We now consider the latter and argue that w.p. at least $1 - 6\alpha/\delta$ the node v_{2t-1} is not activated, *i. e.,* $v_{2t-1} \notin M_1$. This follows trivially since we assumed that at most 6α neighbors were activated (chosen u.a.r.) and $|N^+(v_{2t})| \geq \delta$.

Assuming this holds, all elements of M_1 have ϕ-value less than or equal to $z - 1$. Then, Proposition 4.14 (with $k = 0$) applied to each sub-tree rooted at nodes of M_1 and taking a Union bound, shows that with probability at least $1 - 6\alpha/(12\alpha^4)$, all other nodes visited by $\mathrm{BurnBFS}(G, v_{2t})$ have ϕ-value less than or equal to $z - 2$. Assume this holds.

Moreover, by Lemma 4.15, with probability at least $1 - 1/(2\alpha)$, we have $\deg^+(v_{2t+1}) \geq \delta$. Assume this holds.

Assuming all those events hold, $N^+(v_{2t+1})$ satisfies all the statements of the lemma. The probability that one of the assumptions we made along the way fails to be realised is, by Union bound, at most

$$2^{-6\alpha} + \frac{6\alpha}{\delta} + \frac{6\alpha}{12\alpha^4} + \frac{1}{2\alpha} \leq \frac{1}{\alpha}.$$

□

We now prove the majorisation Lemma 4.13.

Proof of Lemma 4.13. We start by proving the first part of the statement.

Majorization. Recall that $\mathbb{P}[\,Z = i\,] = 3^{-i/2}$ for $i \geq 5$, and $\mathbb{P}[\,Z = 4\,] = 1 - \frac{1}{9(\sqrt{3}+1)}$.

Fix the history $\mathcal{F}_{2t}$ up to time $2t$ and let

$$\begin{aligned}\Delta &:= \phi(v_{2t+2}) - \phi(v_{2t}) \\ &= \phi(v_{2t+2}) - \phi(v_{2t+1}) + \phi(v_{2t+1}) - \phi(v_{2t}) \\ &= (\phi(v_{2t+2}) - \phi(\mathrm{amb}(v_{2t+2}))) + (\phi(v_{2t+1}) - \phi(\mathrm{amb}(v_{2t+1}))).\end{aligned}$$

If $\Delta \geq i \geq 4$, then at least one of the two expressions on the right hand side exceeds $\lceil i/2 \rceil \geq 2$, so

$$\begin{aligned}\mathbb{P}[\,\Delta \geq i \mid \mathcal{F}_{2t}\,] \leq\ & \mathbb{P}[\,\phi(v_{2t+1}) - \phi(\mathrm{amb}(v_{2t+1})) \geq \lceil i/2 \rceil \mid \mathcal{F}_{2t}\,] \\ &+ \mathbb{P}[\,\phi(v_{2t+2}) - \phi(\mathrm{amb}(v_{2t+2})) \geq \lceil i/2 \rceil \mid \mathcal{F}_{2t+1}\,].\end{aligned}$$

Thus we need an upper bound on $\mathbb{P}[\,\phi(v) - \phi(\mathrm{amb}(v)) \geq j \mid \mathcal{F}\,]$ for $j \geq 2$, where $\mathcal{F}$ is the history right before the arrival of v.

Consider the process $\mathrm{BurnBFS}(G, \mathrm{amb}(v))$ with output $T_{\mathrm{amb}(v)}$ executed to construct $N^+(v)$. By definition of ϕ, the sub-tree $T_{\mathrm{amb}(v)}$ needs to contain a node with ϕ-value at least $\phi(\mathrm{amb}(v)) + j - 1$. We use Proposition 4.14 by setting $u = \mathrm{amb}(v)$, $k = \lceil i/2 \rceil - 1$, and can therefore write, by Union bound,

$$\mathbb{P}[\,\Delta \geq i\,] \leq 2\frac{3^{-(\lceil i/2 \rceil - 1)}}{12\alpha^4} = 3^{-\lceil i/2 \rceil}\tfrac{1}{2\alpha^4}. \tag{4.7}$$

Thus for any $i \geq 5$ and large enough constant α we have

$$\mathbb{P}[\,\Delta \geq i\,] \leq 3^{-\lceil i/2 \rceil}\tfrac{1}{2\alpha^4} \leq 3^{-i/2} = \mathbb{P}[\,Z = i\,] \leq \mathbb{P}[\,Z \geq i\,]. \tag{4.8}$$

Note that for $i < 5$ we have $\mathbb{P}[\,Z \geq i\,] = 1 \geq \mathbb{P}[\,\Delta \geq i\,]$. Thus, we have $\Delta \leq^{\mathrm{st}} Z$.

On the other hand, by definition of ϕ, $\phi(v_{2t+2}) - \phi(v_{2t}) \geq -4$, so $-\Delta \leq^{\mathrm{st}} Z$. Thus $|\Delta| \leq^{\mathrm{st}} Z$, which yields the first part of Lemma 4.13. We now turn to the second statement.

Negative bias. Fix $\mathcal{F}_{2t}$ and consider the arrival of v_{2t+1}. With probability at least $1 - 1/\alpha$ the situation described in Lemma 4.16 happens. Assume that to be the case, and consider the arrival of v_{2t+2}. Consider the process $\mathrm{BurnBFS}(G, v_{2t+1})$ executed to construct $N^+(v_{2t+2})$. Once again, by Chernoff bounds (Proposition A.3), with probability at least $1 - 2^{-6\alpha}$ we have $|M_1| \leq 6\alpha$. Assume this holds. By Lemma 4.16 we know that v_{2t+1} has at least δ neighbours, of which only $6\alpha + 1$ may have ϕ-value greater than or equal to $z - 1$. By Union bound, with probability at least $1 - 6\alpha(6\alpha + 1)/\delta$, none of the nodes of M_1 are in that set, and therefore all nodes of M_1 have ϕ-value less than or equal to $z - 2$. Then, Proposition 4.14 (for $k = 0$) applied to all sub-trees rooted at nodes of M_1 shows

that with probability at least $1 - 6\alpha/(12\alpha^4)$, all other nodes visited by BurnBFS(G, v_{2t+1}) have ϕ-value less than or equal to $z - 2$. Assume that holds.

By Lemma 4.15, with probability at least $1 - 1/(2\alpha)$, we have $\deg^+(v_{2t+2}) > \delta$. Assume this holds. Then by definition of ϕ and since $\phi(v_{2t}) > 2$, we obtain $\phi(v_{2t+2}) \leq z - 1$, and thus

$$\phi(v_{2t+2}) \leq \phi(v_{2t}) - 1.$$

The probability that one of the assumptions we made along the way fails to be realised is at most

$$\frac{1}{\alpha} + 2^{-6\alpha} + \frac{6\alpha(6\alpha+1)}{\delta} + \frac{6\alpha}{12\alpha^4} + \frac{1}{2\alpha} \leq \frac{2}{\alpha}.$$

To recap, if we let $\Delta = \phi(v_{2t+2}) - \phi(v_{2t})$, we have just proved that

$$\mathbb{P}[\,\Delta \leq -1\,] \geq 1 - \frac{2}{\alpha}.$$

To compute the expectation (implicitly conditioning on $\mathcal{F}_{2t}$), we now write

$$\begin{aligned}\mathbb{E}[\,\Delta\,] &\leq \sum_{k\geq 4} k \cdot \mathbb{P}[\,\Delta = k\,] + 3 \cdot \mathbb{P}[\,0 \leq \Delta \leq 3\,] - \mathbb{P}[\,\Delta \leq -1\,] \\ &\leq \sum_{k\geq 4} \mathbb{P}[\,\Delta \geq k\,] + 3 \cdot \frac{2}{\alpha} - \left(1 - \frac{2}{\alpha}\right). \end{aligned} \tag{4.9}$$

The first term on the right hand side can be bounded using (4.7):

$$\sum_{k\geq 4} \mathbb{P}[\,\Delta \geq k\,] \leq \sum_{k\geq 4} 3^{-\lceil k/2 \rceil} \frac{1}{2\alpha^4} \leq \frac{1}{\alpha^2}.$$

We finally obtain

$$\mathbb{E}[\,\Delta \mid \mathcal{F}_{2t}\,] \leq -1 + \frac{8}{\alpha} + \frac{1}{\alpha^2} < 0,$$

hence the negative bias.

□

4.5.4 Proof of Proposition 4.14

Proof. The main idea of the proof is to couple the tree process defined by BurnBFS with a Galton-Watson Process. Let i be as in the statement of the lemma, and suppose that the sets $M_0, \ldots, M_i$ have already been fixed by the activation decisions in BurnBFS. We look at $u \in M_i$, the designated vertex in the statement of the lemma.

Let w be some vertex in T, the tree generated by BurnBFS, and say $w \in M_{k-1}$. We are interested in understanding the random variable that is the number of children of w in T. Let M_k^w denote the set M_k right after the activation decisions for edges of vertices in M_{k-1} that are before w in the ordering are completed. Let $S = \{x \in N^+(w) \mid x \notin$

$\cup_{j<k} M_j$ and $x \notin M_k^w\}$ be the random variable (depending on the choices made while determining $M_0, \ldots, M_{k-1}, M_k^w \setminus \{w\}$), that is the set of potential children of w. Let $p = \min\{1, \frac{\alpha}{\deg^+(w)}\}$. Let B be obtained by adding each $x \in S$ to B with probability p. Thus,

$$Z_w := |B| \sim \mathsf{Bin}(|S|, p)$$

and B corresponds to the activated edges that lead to nodes not already in $\cup_{j<k} M_j \cup M_k^w$. We define R_w to be a random variable

$$R_w = \begin{cases} 0 & \text{if } \deg^+(w) < \delta \\ 1 & \text{if } \deg^+(w) \geq \delta \text{ and } \mathrm{amb}(w) \in B. \end{cases}$$

We will call the edge $(w, \mathrm{amb}(w))$ in T purple if $R_w = 1$. We are interested in the random variables (Z_w, R_w) (note that they are dependent on random choices made earlier in the process defined above; however, to minimise cumbersome notation we will not make this explicit).

We will now define a branching process that is completely independent of the Line Fire process. It is a Galton-Watson process (see Section A.5.1 for a definition), with some designated red edges. Let Z', R' be random variables

$$Z' \sim 1 + \lceil e\alpha \rceil + \mathsf{Poisson}(e\alpha)$$

that defines the offspring distribution of the Galton-Watson process. Furthermore,

$$R' \sim \mathsf{Bernoulli}(\alpha/\delta),$$

is the indicator variable where $R' = 1$ if and only if the edge between the node and its "first" child is marked red. We will show that this process stochastically dominates the branching process resulting from a call to Burn, in a particular technical sense. We have the following claim:

Claim 4.17. *Let w be some node in M_{k-1}, and let S, p, B be as defined above. Let (Z_w, R_w) be the random variables defined above for the burn process. Let Z', R' be as used to define the independent Galton-Watson process. Then, whenever $\alpha \leq \delta$, there exists a coupling of the random variables such that $Z_w < Z'$ and $R_w \leq R'$.*

Subproof. We distinguish between three cases.

1. If $\deg^+(w) < \delta$, then $R_w = 0 \leq R'$. So we only need to define a coupling so that $Z_w < Z'$.

2. If $\deg^+(w) \geq \delta$, but $\mathrm{amb}(w) \notin S$ (that is $\mathrm{amb}(w)$ is in some M_j for $j \leq k$ already when the activation decisions for out-edges of w were made), then $R_w = 0 \leq R'$, and again we just need to define a coupling so that $Z_w < Z'$.

3. Finally, consider the case $\deg^+(w) \geq \delta$ and $\mathrm{amb}(w) \in S$. Note that $Z_w = R_w + \tilde{Z}_w$, where $R_w \sim \mathsf{Bernoulli}(p)$ and $\tilde{Z}_w \sim \mathsf{Bin}(|S| - 1, p)$ (when $\deg^+(w) \geq \delta$, $p = \min\{1, \alpha/\deg^+(w)\} = \alpha/\deg^+(w)$, as long as $\delta \geq \alpha$). Note that (Z_w, R_w) have the exact same joint distribution as defined above, since effectively we are making the choice of whether or not $\mathrm{amb}(w)$ should be included in B independently of the other elements. Since $\alpha/\deg^+(w) \leq \alpha/\delta$, it is clear that we can couple R_w and R' so that $R_w \leq R'$. Thus, again it remains only to show a coupling such that $Z_w < Z'$.

For all $\alpha \geq 1$, it follows that $\mathsf{Bin}(n, p)$ is stochastically dominated by $\mathsf{Poisson}(e\alpha)$ whenever $p \leq \alpha/n$ and $n \geq e\alpha$ (see *e. g.,* [KM10]). When, $n < e\alpha$, clearly $\mathsf{Bin}(n, p)$ is stochastically dominated by $1 + \lceil e\alpha \rceil$. The additional 1 in the definition of Z' takes care of the strict inequality made in the claim. This completes the proof. ■

Let T' denote the (possibly infinite) Galton-Watson tree with offspring distribution Z' and some edges marked "red" as defined above. We define a coupling between the (random) sub-tree T_u generated by the Burn process (rooted at $u \in M_i$) and T' inductively below: This results in an injective map σ from $V(T_u)$ to $V(\mathrm{T}')$, where $V(T)$ denotes the vertices in tree T. Let ρ denote the root of T', then σ is defined as follows (through coupling and induction on $\mathrm{dist}_{T_u}(u, w)$). Note that u is the only vertex with $\mathrm{dist}_{T_u}(u, u) = 0$.

1. $\sigma(u) = \rho$

2. Suppose all $w \in T_u$ with $\mathrm{dist}_{T_u}(u, w) \leq \Delta$ are mapped under σ to some vertices in T'. We look at the time when activation decisions for some w such that $\mathrm{dist}_{T_u}(u, w) = \Delta$ are made. For each such w, we apply the coupling defined in Claim 4.17. Let (Z_w, R_w) be the corresponding random variables and let (Z', R') be the independent instantiation of the random variables denoting the children of $\sigma(w)$ in T'. By the coupling, we have $Z_w < Z'$ and $R_w \leq R'$. If $R_w = 1$, we set $\sigma(\mathrm{amb}(w))$ to be the "red" (first) child of $\sigma(w)$. The remaining $Z_w - 1$ children of w can be mapped to the subsequent $Z_w - 1$ children of $\sigma(w)$, which is possible by the coupling. If $R_w = 0$, all Z_w children of w are mapped to the non-"red" children of $\sigma(w)$, which again is possible since $Z' > Z_w$. This defines the map σ for all vertices w', such that $\mathrm{dist}_{T_u}(u, w') = \Delta + 1$ and completes the inductive step.

We observe that the map σ satisfies the following properties by definition:

1. If (w, x) is an edge in T_u, then $(\sigma(w), \sigma(x))$ is an edge in T', and furthermore the edge $(\sigma(w), \sigma(x))$ points away from the root.

2. If (w, x) is coloured purple in T_u, then $(\sigma(w), \sigma(x))$ is coloured purple in T'.

Finally, we define a function $\phi' : V(\mathrm{T}') \to \mathbb{N}$, on the nodes of the tree T' as follows:

1. $\phi'(\rho) = \phi(u)$
2. For w', let $\mathrm{parent}(w')$ denote the parent of w' in T'. Then,

$$\phi'(w') = \begin{cases} \phi'(\mathrm{parent}(w')) + 2 & \text{if } (\mathrm{parent}(w'), w') \text{ is red} \\ \phi'(\mathrm{parent}(w')) - 1 & \text{otherwise.} \end{cases}$$

We check the following fact:

Claim 4.18. *For every $w \in V(T_u)$, $\phi(w) \leq \phi'(\sigma(w))$*

Subproof. The proof is based on induction on $\mathrm{dist}_{T_u}(u, w)$. Clearly, when $\mathrm{dist}_{T_u}(u, w) = 0$, it must be the case that $w = u$, and we have $\phi'(\sigma(u)) = \phi'(\rho) = \phi(u)$. Suppose, this holds for all w such that $\mathrm{dist}_{T_u}(u, w) \leq \Delta$. Consider an edge (w, x) in T_u, such that $\mathrm{dist}_{T_u}(u, x) = \Delta + 1$. Then, we consider two cases:

- If (w, x) is coloured purple, $x = \mathrm{amb}(w)$. Also, in this case, the edge $(\sigma(w), \sigma(x))$ in T' is also coloured purple. Hence by definition $\phi'(\sigma(x)) = \phi'(\sigma(w)) + 2 \geq \phi(w) + 2$. On the other hand, since (w, x) is red, we have that $\deg^+(w) \geq \delta$ and $x = \mathrm{amb}(w)$. Thus, by definition of ϕ, a node cannot have potential difference of more than -2 w.r.t. its ambassador, *i. e.,* $\phi(w) \geq \phi(x) - 2$. Hence, putting everything together, we have $\phi(x) \leq \phi(w) + 2 \leq \phi'(\sigma(x))$.
- On the other hand, if (w, x) is not red, we have that $\phi'(\sigma(x)) = \phi'(\sigma(w)) - 1 \geq \phi(w) - 1$. Also by definition of ϕ, we know that $\phi(w) \geq \phi(x) + 1$ for all $x \in N^+(w) \setminus \{\mathrm{amb}(w)\}$. Hence, putting everything together, we have $\phi(x) \leq \phi(w) - 1 \leq \phi'(\sigma(x))$.

This yields the claim. ■

Using Claim 4.18 we have,

$$\mathbb{P}[\, \exists w \in T_u \; s.t. \; \phi(w) \geq \phi(u) + k \mid \mathcal{H} \,] \leq \mathbb{P}[\, \exists w \in \mathrm{T}' s.t. \; \phi'(w) \geq \phi'(\rho) + k \,]. \tag{4.10}$$

Thus, it only remains to analyze ϕ' on T' to bound the r.h.s. of (4.10) and thus yielding the proposition.

Claim 4.19. *We have, for $k \geq 0$*

$$\mathbb{P}[\, \exists w \in \mathrm{T}' \; s.t. \; \phi'(w) \geq \phi'(\rho) + k \,] \leq \frac{3^{-k}}{12\alpha^4},$$

where ρ is the root.

Subproof. Let

$$\alpha' = (1 + \lceil e\alpha \rceil + e\alpha);$$

for $i \geq 0$, let

$$b_i = (k+1)(6\alpha')^i$$

and let N_i be the random variable denoting the number of nodes of T' at distance i from the root ρ. We have the following:

$$\begin{aligned} &\mathbb{P}\big[\, \exists w \in \mathrm{T}' s.t.\ \phi'(w) \geq \phi'(\rho) + k \,\big] \leq \\ &\quad \leq \mathbb{P}\left[\exists w \in \mathrm{T}'\ s.t.\ \phi'(w) \geq \phi'(\rho) + k \;\middle|\; \bigcap_i \{N_i < b_i\} \right] \cdot 1 + \mathbb{P}\left[\bigcup_i \{N_i \geq b_i\} \right] \end{aligned} \tag{4.11}$$

We bound the two terms of (4.11) separately. To bound the first term, we use a Union bound:

$$\begin{aligned} &\mathbb{P}\left[\exists w \in \mathrm{T}'\ s.t.\ \phi'(w) \geq \phi'(\rho) + k \;\middle|\; \bigcap_i \{N_i < b_i\} \right] \leq \\ &\quad \leq \sum_{j \geq 1} b_j \max_w \mathbb{P}\big[\, \phi'(w) \geq \phi'(\rho) + k \mid \operatorname{dist}(\rho, w) = j \,\big] \end{aligned} \tag{4.12}$$

We now bound for an arbitrary (directed) path $\mathcal{P}$, with vertices $v_0 = \rho, v_1, \ldots, v_j = w$ in T', the probability that $\phi'(w) \geq \phi'(\rho) + k$. Note that as we go down the tree T', the value of ϕ' only decreases, except on red edges. Observe by definition of the tree, that the number of children of any node, distributed according to $Z' \sim 1 + \lceil e\alpha \rceil + \mathsf{Poisson}(e\alpha)$ is independent of whether or not the first node is coloured purple (since the random variable R' and $Z' \geq 1$ are drawn independently). Therefore, by assuming that every edge along the path can potentially be *red*, we are only increasing the probability that for some node w, $\phi'(w) \geq \phi'(\rho) + k$. Note that the probability that any edge out of a node is red, denoted by p_r is α/δ (for the first child of a node, which always exists since the number of children is at least $1 + \lceil e\alpha \rceil$, the probability is p_r, for the remaining it is 0). Let r denote the number of purple edges in the path $v_0 = \rho, v_1, \ldots, v_j = w$, then the number of non-purple edges is $j - r$. Thus, by definition of ϕ', $\phi'(w) = \phi'(\rho) + 2 \cdot r - 1 \cdot (j - r) = \phi'(\rho) + 3r - j$, and hence

for $\phi'(w) \geq \phi(\rho) + k$ to be true, it must be the case that $r \geq \left\lceil \frac{j+k}{3} \right\rceil$. Thus, we have

$$\begin{aligned}
\mathbb{P}\Big[\text{At least } \left\lceil \tfrac{j+k}{3} \right\rceil \text{ edges in } \mathcal{P} \text{ are red} \Big] &\leq \sum_{b=\left\lceil \frac{j+k}{3} \right\rceil}^{j} \binom{j}{b} p_r^b (1-p_r)^{(j-b)} \\
&\leq \sum_{b=\left\lceil \frac{j+k}{3} \right\rceil}^{j} \left(\frac{ej}{b}\right)^b p_r^b \\
&\leq \sum_{b \geq \left\lceil \frac{j+k}{3} \right\rceil} (3e \cdot p_r)^b \\
&\leq (3e \cdot p_r)^{\frac{j+k}{3}} \cdot \tfrac{1}{1-3ep_r}.
\end{aligned}$$

Substituting this bound in (4.12), we get using $p_r = \alpha/\delta = 1/\alpha^{19}$

$$\begin{aligned}
&\mathbb{P}\left[\exists w \in \mathrm{T}' s.t.\ \phi'(w) \geq \phi'(\rho) + k \;\middle|\; \bigcap_i \{N_i < b_i\} \right] \leq \\
&\quad \leq \sum_{j \geq 1} b_j \tfrac{1}{1-3ep_r} \Big((3ep_r)^{1/3} \Big)^{j+k} \\
&\quad \leq (k+1) \frac{1}{1-3ep_r} \Big((3ep_r)^{1/3} \Big)^{k} \sum_{j \geq 1} \Big(6\alpha' (3ep_r)^{1/3} \Big)^{j} \\
&\quad \leq (k+1) \frac{1}{1-3ep_r} \Big((3ep_r)^{1/3} \Big)^{k} \cdot \frac{6\alpha' (3ep_r)^{1/3}}{1 - 6\alpha'(3ep_r)^{1/3}} \\
&\quad \leq (k+1) 3 \frac{1}{\frac{1}{2}} \Big(p_r^{1/3} \Big)^{k} \cdot \frac{6\alpha' 3 \cdot p_r^{1/3}}{\frac{1}{2}} \\
&\quad \leq (k+1) 216 \alpha' \frac{1}{\alpha^{6(k+1)}} \\
&\quad \leq \frac{3^{-k}}{24\alpha^4}, \qquad (4.13)
\end{aligned}$$

whenever $\alpha \geq 100$. Now, we analyse the second term of (4.11). Let $\mathcal{E}_i$ denote the event that $N_i \geq b_i$. Thus, we are interested in bounding $\mathbb{P}[\bigcup_i \mathcal{E}_i]$. Observe, that:

$$\mathbb{P}\left[\bigcup_i \mathcal{E}_i \right] \leq \sum_{i \geq 1} \mathbb{P}\Big[\mathcal{E}_i \mid \overline{\mathcal{E}}_{i-1} \Big].$$

We observe that N_i is a sum of N_{i-1} independent copies of $Z' \sim 1 + \lceil e\alpha \rceil + \mathsf{Poisson}(e\alpha)$. Lemma A.10 proves that,

$$\mathbb{P}\Big[\mathcal{E}_i \mid \overline{\mathcal{E}}_{i-1} \Big] \leq 2^{-6eab_{i-1}}$$

Thus, we have:

$$\mathbb{P}\left[\bigcup_i \mathcal{E}_i\right] \leq \sum_{i\geq 1} 2^{-6eab_{i-1}} \leq 2 \cdot 2^{-6eab_0} = 2 \cdot 2^{-(k+1)(36e\alpha)} \leq \frac{3^{-k}}{24\alpha^4}. \tag{4.14}$$

Substituting (4.13) and (4.14) in (4.11) concludes the proof:

$$\mathbb{P}\big[\,\exists w \in \mathrm{T}' s.t.\ \phi'(w) \geq \phi'(\rho) + k\,\big] \leq 2 \cdot \frac{3^{-k}}{24\alpha^4} = \frac{3^{-k}}{12\alpha^4}.$$

■

This completes the proof of Proposition 4.14.

□

4.5.5 The foundations of ϕ

In order to derive a constant expected distance, we require some notion of positive recurrence: Over the course of t time steps, subgraphs with a distance of $\omega(1)$ emerge with constant probability and we need to show that the graph "recovers quickly" from this. For this reason, we show that whenever ϕ is large at time t, it decreases in expectation regardless of the structure of the graph at time t. Such an analysis seems to break if one analyzes the distance directly: One can construct worst-case graphs, where the distance does not decrease in expectation in a constant number of steps. For example one can construct line-like graphs in which the distance is likely to increase considerably. The reason why we are able to show that ϕ decreases is that in such graphs the degree of the nodes is small and thus causing ϕ to be relative large (w.r.t. the distance) allowing the potential (in contrast to the distance) to decrease in expectation. On the other side, creating such a worst-case subgraph is extremely unlikely (otherwise ϕ would increase in expectation). These worst-case graphs are the intuition behind the second line of ϕ, namely

$$\phi(v) = \max_{w \in N^+(v)} \{\phi(w)\} + 1 \text{ if } \deg^+(v) < \delta\ .$$

The third line of ϕ, namely

$$\phi(v) = \max\left\{\phi(\mathrm{amb}(v)) - 2,\ \max_{\substack{w \in N^+(v) \\ w \neq \mathrm{amb}(v)}} \phi(w) + 1\right\}$$

consists of two parts: The second part is the core of ϕ and measures essentially the longest path ignoring ambassador edges (note that the longest path from a node to G_0 is monotonically increasing due to to ambassador edges which is the reason why we disregard them). The first part (of the third line) ensure that the potential cannot increase by too much if

an ambassador edge of a node is activated; without the constraint, the potential increase caused by just activating one edge could be unbounded.

4.6 Lower bound - Proof of Lemma 4.5

We set $\alpha^* = 1/(4e)$. We majorize the original process by a process P in which the following holds. First, every arriving node performs $\mathsf{Poisson}(e\alpha)$ burns nodes upon arrival. As we show, this strictly majorizes the number of neighbours burns and hence increases a node's neighborhood and its distance to L_0. Second, all the burns happen along the shortest path which, intuitively speaking, only decreases the distance to L_0.

Let $\Delta_t = \mathrm{dist}_{L_t}(v_t, L_0) - \mathrm{dist}_{L_{t-1}}(v_{t-1}, L_0)$. We have for process P that $-\infty < \Delta_t \leq 1$. We show that it's very unlikely that the distance upon arrival of a new decreases, more precisely we show the tail bound $\mathbb{P}[\,\Delta_t = -k\,] \leq e^{-(k+1)}$ by majorizing with a Galton Watson tree with offspring distribution $\mathsf{Poisson}(e\alpha)$. From this we get, $\mathbb{E}[\,\Delta_t \mid \mathcal{F}_{t-1}\,] \geq 1/2$ which allows us to conlcude that $\mathbb{E}[\,\mathrm{dist}_{L_t}(v_t, L_0)\,] = \Omega(t)$.

Proof of Lemma 4.5. We set $\alpha^* = 1/(4e)$. As mentioned earlier, the neighbours of a node v_t in the Line Fire process can be represented by the vertices of a tree T rooted at v_{t-1}, in which every node v appears at most once. Furthermore, the number of edges percolated by v is $X_v \sim \mathsf{Bin}\Big(|\mathcal{N}(v)|, \min\Big\{1, \frac{\alpha}{|\mathcal{N}(v)|}\Big\}\Big) = \mathsf{Bin}\Big(|\mathcal{N}(v)|, \frac{\alpha}{|\mathcal{N}(v)|}\Big)$ for $\alpha < 1$. Since X_v depends on the degree of node v we will make use of Lemma A.9 and majorize X_v by the degree-independent distribution $\mathsf{Poisson}(e\alpha)$.

We define a process P in which, at the arrival of v_t, node v_{t-1} percolates $Y_{v_{t-1}} \sim \mathsf{Poisson}(e\alpha)$ outgoing edges uniformly at random. Moreover, whenever an edge (v_1, v_2) is percolated, node v_2 percolates $Y_{v_2} \sim \mathsf{Poisson}(e\alpha)$ of its outgoing edges uniformly at random. Let T'_{v_t} (T_{v_t}, respectively) be the resulting tree of percolated edges in the new process (original process, respectively).

Node v_t connects then to all nodes of T'_{v_t} – If the same node v appears several times, then we only connect v_t once to v. However, we allow v to be burnt several times: every time v is added, it chooses $\mathsf{Poisson}(e\alpha)$ children u.a.r. and independent of former choices to percolate.

Since $Pr(X_v \geq k) \leq Pr(Y_v \geq k)$ for $k \geq 1$ (by Lemma A.9), we can couple the trees T_v and T'_{v_t} such that if $v \in T$, then $v \in T'_{v_t}$. This implies that the neighborhood of v_t in the original process is a subset of the neighborhood of v_t in P. Hence, the distance of v_t in the original process is at least the distance of v_t in P.

Let $\Delta_t = \mathrm{dist}_{L_t}(v_t, L_0) - \mathrm{dist}_{L_{t-1}}(v_{t-1}, L_0)$. We have $-\infty < \Delta_t \leq 1$. The distance of a node v to L_0 in L_t equals the number of nodes on the shortest path plus one. Hence, we obtain a crude bound on $\mathbb{P}[\,\Delta_t = -k\,]$ by bounding $\mathbb{P}[\,|T'_{v_t}| \geq k+1\,]$. As we will argue in the following, $\mathbb{P}[\,|T'| \geq k+1\,]$ has an exponential tail distribution. Observe, that T'_{v_t} is

GW-tree with offspring distribution $\mathsf{Poisson}(e\alpha)$. We have, by Proposition A.33, using that $e\alpha \leq e\alpha^* = 1/4$,

$$\mathbb{P}[\,\Delta_t = -k|\mathcal{F}_{t-1}\,] \leq \mathbb{P}[\,|T'_{v_t}| \geq k+1\,] \leq e^{-h(1)\cdot(k+1)} \leq e^{-(k+1)},$$

where $h(1) \geq 1$ is the function defined in Proposition A.33. Hence,

$$\begin{aligned}
\mathbb{E}[\,\Delta_t|\mathcal{F}_{t-1}\,] &\geq \mathbf{1}\cdot\mathbb{P}[\,X_{v_{t-1}} = 0\,] + \sum_{k\geq 1}(-k)\cdot\mathbb{P}[\,\Delta_t = -k|\mathcal{F}_{t-1}\,] \\
&\geq \min_{d\geq 1}\{(1-\alpha/d)^d\} + \sum_{k\geq 1} -ke^{-(k+1)} \\
&\geq 1-\alpha-2/5 \geq 1/2.
\end{aligned}$$

Hence, $\mathbb{E}[\,\mathrm{dist}_{L_t}(v_t, L_0)|\mathcal{F}_{t-1}\,] \geq \mathrm{dist}_{L_{t-1}}(v_{t-1}, L_0) + 1/2$. We have,

$$\begin{aligned}
\mathbb{E}[\,\mathrm{dist}_{L_t}(v_t, L_0)\,] &= \mathbb{E}[\,\mathbb{E}[\,\mathrm{dist}_{L_t}(v_t, L_0)|\mathcal{F}_{t-1}\,]\,] \\
&\geq \mathbb{E}[\,\mathrm{dist}_{L_{t-1}}(v_{t-1}, L_0)\,] + 1/2.
\end{aligned}$$

Hence, by repeating this iteratively, we get $\mathbb{E}[\,\mathrm{dist}_{L_t}(v_t, L_0)\,] = \Omega(t)$, which yields the claim. □

4.7 Future Work and Conclusion

The Forest Fire model was proposed by Leskovec *et al.* to explain several properties of social networks, *shrinking diameter* being an important one, in addition to *densification* and power-law degree distributions [LKF07]. As the graphs generated are directed, we focused on distance to the seed graph, rather than diameter as the property of interest. This work shows that in a restricted version of the Forest Fire model, we can prove that this distance remains bounded, even as the graph size increases, albeit with some conditions on the seed graph.

There are several natural open questions as to how to proceed from here. The obvious one is whether one can remove the conditions on the seed graph. Our simulation results seem to suggest that starting with a single node as a seed graph should also result in similar behaviour. The next is whether one can address densification. Without backward burning, it is clear that the out-degree of any vertex in G_t can be at most logarithmic in t. This follows from the fact that the edges have to be on directed paths in the ambassador tree, which is of logarithmic depth. Thus, we cannot expect the average edge density to be more than logarithmic in the number of nodes.

Also, for this reason the out-degrees cannot have a heavy tail. In simulations, the in-degrees did exhibit power-law behaviour.

4.7.1 Further models

As mentioned before, backward burning might prove interesting to study as it might result in a super polylogarithmic number of edges as observed in the simulations of [LKF07]. Our analysis however would break: Our potential ϕ was carefully designed to ensure that for every node at most one out of many outgoing (*i. e.,* burnable) edges is "bad", *i. e.,* leading to node with a high potential. This would no longer be true in the case of backward burning since it would be possible that many nodes with a high (possibly much higher) ϕ-value have an edge to a node v and thus, when v is "burnt", v might have many neighbours with high ϕ-values. Changing the potential function alone seems to be futile, as the backward edges of v_t are only added after time step t and thus the potential of v_t might change after time step t rendering our approach inapplicable.

On the other hand, there is a multitude of variations that our techniques allow to analyze. For example the Random Surfer Model, in which at the arrival of a node a random walk is started at the ambassador, which then travers the out-edges of nodes stopping at every node with probability p. Even though this model could be analyzed with our techniques, the required bounds on p to have a constant diameter would not be sharp in contrast to [KLL+16] where we investigate tight bounds for this model. Another natural extension would be to allow "multi burning": in our model, a node is not allowed to "burn" several times, *i. e.,* to introduce the arriving nodes several times to his own friends several times. Without this restriction the process becomes much easier to analyze: The random process behaves like a Galton-Watson process with a suitable mapping from the children in the resulting Galton-Watson tree to the current graph. If the Galton-Watson tree is infinite, then we must be in the cyclic seed graph and thus the distance to the root is 1. For a suitable constant α it is well-known that this is the case. A drawback of this is the resulting infinite edges to the seed-graph; A simple way around this would be to forbid multi-edges and to simply remove them when they occur (the analysis still works). In this model, it would even be possible the drop the assumption on the seed graph: Once a Galton-Watson tree exceeds a certain depth and by making use of the underlying mapping, we know that the new node must have reached the root-node.

Further models which easily fit in our framework are those where the probability of continuing decays as function of the depth as well as models where the out-edges chosen are always the edges which lead the"oldest" neighbour of a node. Finally, the following variation also falls into our framework: each outgoing edge is activated with constant probability, *i. e.,* without normalizing by the out-degree of a node. This model behaves like the aforementioned Random Surfer model with additional edges. It is worth mentioning that it is unclear whether any of the above models exhibits the densification (of the edges) property.

4.7.2 Conclusion

To summarize, we showed that the Forest Fire model with forward burning exhibits a constant expected distance for large enough constant α and a logarithmic distance for small enough constant α. The machinery we developed and in particular the core ideas of the potential function allow the study of a wide-range of variants for models of social networks. We were able to apply the high-level ideas of our approach to the at first glance unrelated appearing problem of balls-into-bins (see Chapter 5). We believe that our techniques find application in settings beyond the realm of social networks and balls-into-bins.

Chapter 5

Balls-into-Bins with Deleting Bins [BFK+16a]

One of the fundamental problems in distributed computing is the distribution of requests, tasks, or data items to a set of uniform servers. In order to simplify this process and to avoid a single point of failure, it is often advisable to use a simple, randomized strategy instead of a complex, centralized controller to allocate the requests to the servers. In the most naive strategy (GREEDY[1]), each client sends its request to a server chosen uniformly at random. A more elaborate scheme (GREEDY[2]) chooses two servers, queries their current loads, and sends the request to the least loaded of them. Both approaches are typically modeled as balls-into-bins processes [Gon81, RS98, ABKU99, BCSV06, ACMR98, Ste96, BCN+15a], where requests are represented as balls and servers as bins. While the latter approach leads to considerably better load distributions [ABKU99, BCSV06], it loses some of its power in synchronous settings, where requests arrive in parallel and cannot take each other into account [ACMR98, Ste96].

We propose and study a novel infinite batch-based balls-into-bins process to model the client-server scenario. In a round, each server (bin) consumes one of its current tasks (balls). Afterward, in expectation λn tasks arrive and are allocated using a given distribution scheme. The *arrival rate* λ is allowed to be a function of n (e.g., $\lambda = 1 - 1/\operatorname{poly}(n)$). Standard balls-into-bins results imply that, for high arrival rates, with high probability[1] (w.h.p.) in each round there is a bin that receives $\Theta(\log n / \log \log n)$ balls. Most other infinite balls-into-bins-type processes limit the total number of concurrent balls in the system by n [ABKU99, BCN+15a] and show a fast recovery.

[1] An event $\mathcal{E}$ occurs *with high probability* (w.h.p.) if $\mathbb{P}[\mathcal{E}] = 1 - n^{-\Omega(1)}$.

5.1 Results

Since we do not limit the number of balls, our process can, in principle, result in an arbitrary high system load. In particular, if starting in a high-load situation (e.g., exponentially many balls), we cannot recover in a polynomial number of steps. Instead, we regard the system load as a Markov chain and adapt the following notion of *self-stabilization*: The system is positive recurrent (expected return time to a typical low-load situation is finite), and taking a snapshot of the load situation at an *arbitrary* (even super-exponential large) time step yields (w.h.p.) a time-independent maximum load. Positive recurrence is a standard notion for stability and basically states that the system load is time-invariant. For irreducible, aperiodic Markov chains it implies the existence of a unique stationary distribution (cf. Section 5.4). While this alone does not guarantee a good load in the stationary distribution, together with the snapshot property we can look at an arbitrary time window of polynomial size (even if it is exponentially far away from the start) and give strong load guarantees.

In particular, we give the following bounds on the load in addition to showing positive recurrence:

1-Choice Process: The maximum load at an arbitrary time is (w.h.p.) bounded by $O(\frac{1}{1-\lambda} \cdot \log \frac{n}{1-\lambda})$. We also provide a lower bound which is asymptotically tight for $\lambda \leq 1 - 1/\operatorname{poly}(n)$. While this implies that already the simple 1-Choice process is self-stabilizing, the load properties in a "typical" state are poor: even an arrival rate of only $\lambda = 1 - 1/n$ yields a superlinear maximum load.

2-Choice Process: The maximum load at an arbitrary time is (w.h.p.) bounded by $O(\log \frac{n}{1-\lambda})$. This allows to maintain an exponentially better system load compared to the 1-Choice process; for any $\lambda \leq 1 - 1/\operatorname{poly}(n)$ the maximum load remains logarithmic.

Note that the resulting processes can be seen as queuing processes.

5.2 Approach and Technical Contributions

For the analysis of Greedy[1] the main idea of the proof is to bound the maximum load for any bin i and to take union bound of all resources. The load of bin i decreases whenever it is large and, thus, performs a biased random walk towards a load of zero. However, when the load is zero, it increases in expectation, such that standard drift theorems cannot not be applied directly. Nevertheless, the increase of the load for any given state has an exponential tail, which allows us to apply Hajek's Theorem (Theorem A.11) to derive exponential tail bounds on the load of i at any (possibly super-exponential) number of time steps.

For the analysis of GREEDY[2] we define three different potentials (see Section 5.6 for further details) which measure the total load in the system, the load difference as an exponential function (which was already sucuessfully used in [TW14]) as well as a weighted combination of the first two potentials. We're able to derive strong bounds on the load difference after an arbitrary number. In order to derive bounds on the maximum load of the system after an arbitrary number of steps we essentially use union bounds in an adaptive manner: While we cannot apply the same union bound over exponentially many time steps to bound to the load difference of the system at a given time step t, we apply union bounds at every $\tau < t$ which bound the load difference as a function of $t - \tau$. This together with combinatorical properties of the potentials will allow us to derive strong bounds on the load of the system at an arbitrary point t in time. See Section 5.6.2 for further intuition.

5.3 Related Work

We will continue with an overview of related work. We start with classical results for sequential and finite balls-into-bins processes, go over to parallel settings, and give an overview of infinite and batch-based processes similar to ours. We also briefly mention some results from queuing theory (which is related but studies slightly different quality of service measures and system models).

Sequential Setting. There are many strong, well-known results for the classical, sequential balls-into-bins process. In the sequential setting, m balls are thrown one after another and allocated to n bins. For $m = n$, the maximum load of any bin is known to be (w.h.p.) $(1+o(1)) \cdot \ln(n)/\ln\ln n$ for the 1-Choice process [Gon81, RS98] and $\ln\ln(n)/\ln d + \Theta(1)$ for the d-Choice process with $d \geq 2$ [ABKU99]. If $m \geq n \cdot \ln n$, the maximum load increases to $m/n + \Theta(\sqrt{m \cdot \ln(n)/n})$ [RS98] and $m/n + \ln\ln(n)/\ln d + \Theta(1)$ [BCSV06], respectively. In particular, note that the number of balls above the average grows with m for $d = 1$ but is independent of m for $d \geq 2$. This fundamental difference is known as the *power of two choices*. A similar (if slightly weaker) result was shown by Talwar and Wieder [TW14] using a quite elegant proof technique (which we also employ and generalize for our analysis in Section 5.6). Czumaj and Stemann [CS97] study adaptive allocation processes where the number of a ball's choices depends on the load of queried bins. The authors subsequently analyze a scenario that allows reallocations.

Berenbrink et al. [BKSS13] adapt the threshold protocol from [ACMR98] (see below) to a sequential setting and $m \geq n$ bins. Here, ball i randomly chooses bins until it sees a load smaller than $1 + i/n$. While this is a relatively strong assumption on the balls, this protocol needs only $O(m)$ choices in total (allocation time) and achieves an almost optimal maximum load of $\lceil m/n \rceil + 1$.

Parallel Setting. Several papers (e.g., [ACMR98, Ste96]) investigated parallel settings of multiple-Choice games for the case $m = n$. Here, all m balls have to be allocated in parallel, but balls and bins might employ some (limited) communication. Adler et al. [ACMR98] consider a trade-off between the maximum load and the number of communication rounds r the balls need to decide for a target bin. Basically, bounds that are close to the classical (sequential) processes can only be achieved if r is close to the maximum load [ACMR98]. The authors also give a lower bound on the maximum load if r communication rounds are allowed, and Stemann [Ste96] provides a matching upper bound via a collision-based protocol.

Infinite Processes. In infinite processes, the number of balls to be thrown is not fixed. Instead, in each of infinitely many rounds, balls are thrown or reallocated and bins (possibly) delete old balls. Azar et al. [ABKU99] consider an infinite, sequential process starting with n balls arbitrarily assigned to n bins. In each round one random ball is reallocated using the d-Choice process. For any $t > cn^2 \log\log n$, the maximum load at time t is (w.h.p.) $\ln\ln(n)/\ln d + O(1)$.

Adler et al. [ABS98] consider a system where in each round $m \leq n/9$ balls are allocated. Bins have a FIFO queue, and each arriving ball is stored in the queue of two random bins. After each round, every non-empty bin deletes its frontmost ball (which automatically removes its copy from the second random bin). It is shown that the expected waiting time is constant and the maximum waiting time is (w.h.p.) $\ln\ln(n)/\ln d + O(1)$. The restriction $m \leq n/9$ is the major drawback of this process. A further study of this process, based on differential methods and experiments, was conducted in [BCFV00]. The balls' arrival times are binomially distributed with parameters n and $\lambda = m/n$. Their results indicate a stable behavior for $\lambda \leq 0.86$. A similar model was considered by Mitzenmacher [Mit01], who considers ball arrivals as a Poisson stream of rate λn for $\lambda < 1$. It is shown that the 2-Choice process reduces the waiting time exponentially compared to the 1-Choice process.

Czumaj [Czu98] presents a framework to study the recovery time of discrete-time dynamic allocation processes. In each round one of n balls is reallocated using the d-Choice process. Two models are considered: in the first, the ball to be reallocated is chosen by taking a ball from a random bin. In the second, the ball to be reallocated is chosen by selecting a random ball. From an arbitrary initial assignment, the system is shown to recover to the maximum load from [ABKU99] within $O(n^2 \ln n)$ rounds in the former and $O(n \ln n)$ rounds in the latter case. Becchetti et al. [BCN+15a] consider a similar (but parallel) process. In each round one ball is chosen from every non-empty bin and reallocated to a randomly chosen bin (one Choice per ball). The authors show that (w.h.p.) starting from an arbitrary configuration, it takes $O(n)$ rounds to reach a configuration with maximum load $O(\log n)$. Moreover, if the process starts in a configuration with maximum load $O(\log n)$, then the maximum load stays in $O(\log n)$ for $\text{poly}(n)$ rounds. An interesting connection to our work

is that the analysis of [BCN+15a] is based on an auxiliary TETRIS-process. This process can be seen a special version of our 1-Choice process and is defined as follows: starting from a state with at least $n/4$ empty bins, in each round every non-empty bin deletes one ball. Subsequently, exactly $3n/4$ new balls are allocated to the bins (one choice per ball).

Batch-Processes. Batch-based processes allocate m balls to n bins in batches of (usually) n balls each, where each batch is allocated in parallel. They lie between (pure) parallel and sequential processes. For $m = \tau \cdot n$, Stemann [Ste96] investigates a scenario with n players each having m/n balls. To allocate a ball, every player independently chooses two bins and allocates copies of the ball to both of them. Every bin has two queues (one for first copies, one for second copies) and processes one ball from each queue per round. When a ball is processed, its copy is removed from the system and the player is allowed to initiate the allocation of the next ball. If $\tau = \ln n$, all balls are processed in $O(\ln n)$ rounds and the waiting time is (w.h.p.) $O(\ln \ln n)$. Berenbrink et al. [BCE+12] study the d-Choice process in a scenario where m balls are allocated to n bins in batches of size n each. The authors show that the load of every bin is (w.h.p.) $m/n \pm O(\log n)$. As noted in Lemma 5.9, our analysis can be used to derive the same result by easier means. Batch-processes have also been studied in the operations research community [Bai54, Dow55, BDJ98] though with more practical emphasis. Bailey [Bai54] and Downton [Dow55] study the process where users arrive and are processed once a sufficiently large number (batch) are present and Berg et al. [BDJ98] studies a variant where manufactures deliver in batches and individually.

Queuing Processes. Batch arrival processes have also been considered in the context of queuing systems. A key motivation for such models stems from the asynchronous transfer mode (ATM) in telecommunication systems. Tasks arrive in batches, are stored in a FIFO queue and served by a fixed number of servers which remove the tasks from the queue and process them. Several papers [SZ92, Kam96, KCYK12, Alf03] consider scenarios where the number of arriving tasks is determined by a finite state Markov chain. Results study steady state properties of the system to determine properties of interest (e.g., waiting times or queue lengths). Sohraby and Zhang [SZ92] use spectral techniques to study a multi-server scenario with an infinite queue. Alfa [Alf03] considers a discrete-time process for n identical servers and tasks with constant service time $s \geq 1$. To ensure a stable system, the arrival rate λ is assumed to be at most n/s and tasks are assigned cyclical, allowing to study an arbitrary server (instead of the complete system). Kamal [Kam96] and Kim et al. [KCYK12] study a system with a finite capacity. The tasks which arrive when the buffer is full are lost. The authors study the steady state probability and give empirical results to show the decay of waiting times as n increases.

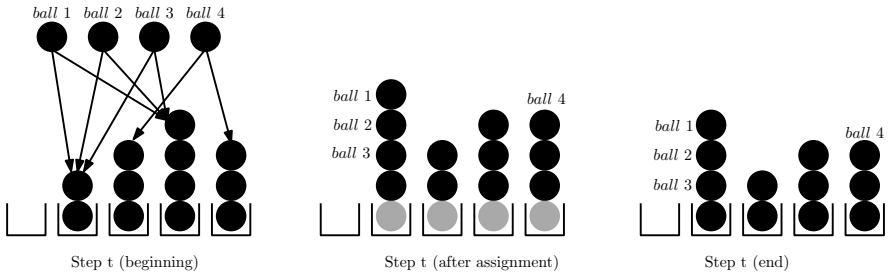


Figure 5.1: The figure depicts a typical round of Greedy[2]. In this example we have $n = 5$ and 4 balls arrive. Balls 1, 2, and 3 choose the same bin with a load of 2 and a bin with larger node and hence all move the same bin resulting in that bin having the highest load. Moreover, Ball 4 chooses two bins with equal load and chooses one of these uniformly at random. At the end of the round all non-empty bins delete one ball (marked gray).

5.4 Model & Preliminaries

We model our load balancing problem as an infinite, parallel balls-into-bins process. Time is divided into discrete, synchronous rounds. There are n bins and n generators, and the initial system is assumed to be empty. At the start of each round, every non-empty bin deletes one ball. Afterward, every generator generates a ball with a probability of $\lambda = \lambda(n) \in [0, 1]$ (the *arrival rate*). This generation scheme allows us to consider arrival rates that are arbitrarily close to one (like $1 - 1/\operatorname{poly}(n)$). Generated balls are distributed in the system using a distribution process. We analyze two specific distribution processes:

- The 1-Choice process Greedy[1] assigns every ball to a random bin.
- The 2-Choice process Greedy[2] assigns every ball to a least loaded among two randomly chosen bins.

See Figure 5.1 for an illustration. It is worth mentioning, that the maximum load in Greedy[2] does not need to be smaller than in Greedy[1] as the following (artificial) example shows. Consider two bins ($n = 2$) with different initial loads and $\lambda = 1$. In Greedy[1] each bin receives $n/2 \pm c\sqrt{n}$ new balls for some constant c. On the other side, in Greedy[2] the bin with the smaller initial load receives $3n/4 \pm c\sqrt{n}$ new balls. However, as our results indicate, this effect becomes negligible when n grows.

Notation. The random variable $X_i(t)$ denotes the load (number of balls) of the i-th fullest bin at the end of round t. Thus, the load situation (configuration) after round t can be described by the load vector $\boldsymbol{X}(t) = (X_i(t))_{i\in[n]} \in \mathbb{N}^n$. We define $\varnothing(t) \coloneqq \frac{1}{n}\sum_{i=1}^{n} X_i(t)$ as the average load at the end of round t. The value $\nu(t)$ denotes the fraction of non-empty bins after round t and $\eta(t) \coloneqq 1 - \nu(t)$ the fraction of empty bins after round t. It will be useful to define $1_i(t) \coloneqq \min(1, X_i(t))$ and $\eta_i(t) \coloneqq 1_i(t) - \nu(t)$ (which equals $\eta(t)$ if i is a non-empty bin and $-\nu(t)$ otherwise). For random variables X and Y we write $X \leq^{\text{st}} Y$ if X is stochastically dominated by Y. That is, if for all k we have $\mathbb{P}[\,X \geq k\,] \leq \mathbb{P}[\,Y \geq k\,]$.

Markov Chain Preliminaries. The random process $(\boldsymbol{X}(t))_{t\in\mathbb{N}}$ has the Markov property, since $\boldsymbol{X}(t)$ depends only on $\boldsymbol{X}(t-1)$ and the random choices during round t. We refer to this Markov chain as $\boldsymbol{X}$. Note that $\boldsymbol{X}$ is time-homogeneous (transition probabilities are time-independent), irreducible (every state is reachable from every other state[2]), and aperiodic (path lengths have no period; in fact, our chain is lazy). Recall that such a Markov chain is positive recurrent (or ergodic) if the probability to return to the start state is 1 and the expected return time is finite. In particular, this implies the existence of a unique stationary distribution. Positive recurrence is a standard formalization of the intuitive concept of stability. See [LP08] for an excellent introduction into Markov chains and the involved terminology.

5.5 The 1-Choice Process

We present two main results for the 1-Choice process: Theorem 5.1 states the stability of the system under the 1-Choice process for an arbitrary λ, using the standard notion of positive recurrence as defined above. In particular, this implies the existence of a stationary distribution for the 1-Choice process. Theorem 5.2 strengthens this by giving a high probability bound on the maximum load for an *arbitrary* round $t \in \mathbb{N}$. Together, both results imply that the 1-Choice process is self-stabilizing. That is, the system is positive recurrent and taking a snapshot of the load situation at an arbitrary time step yields (w.h.p.) a time-independent maximum load.

Theorem 5.1 (Stability)**.** *Let $\lambda = \lambda(n) < 1$. The Markov chain $\boldsymbol{X}$ of the 1-Choice process is positive recurrent.*

Theorem 5.2 (Maximum Load)**.** *Let $\lambda = \lambda(n) < 1$. Fix an arbitrary round t of the 1-Choice process. The maximum load of all bins is (w.h.p.) bounded by $O(\frac{1}{1-\lambda} \cdot \log \frac{n}{1-\lambda})$.*

Note that for high arrival rates of the form $\lambda(n) = 1 - \varepsilon(n)$, the bound given in Theorem 5.2 is inversely proportional to $\varepsilon(n)$. For example, for $\varepsilon(n) = 1/n$ the maximal load is $O(n \log n)$. Theorem 5.3 shows that this dependence is unavoidable: the bound given in Theorem 5.2 is tight for large values of λ.

Theorem 5.3. *Let n be sufficiently large. Let $\lambda = \lambda(n) \geq 3/4$ and consider step $t := 9\lambda \log(n)/(64(1-\lambda)^2)$. With probability $1 - o(1)$ there is a bin i in step t with load $\Omega(\frac{1}{1-\lambda} \cdot \log n)$.*

The proofs of these results can be found in the following subsections. We first prove a bound on the maximum load (Theorem 5.2). Afterward, we prove stability of the system (Theorem 5.1). Finally we prove the lower bound (Theorem 5.3).

[2]The state space includes all vectors with non-increasing entries over $\mathbb{N}^n$.

5.5.1 Maximum Load - Proof of Theorem 5.2

The proof idea is to apply Hajek's Theorem (Theorem A.11) to derive exponential tail bounds on the load of i at any (possibly super-exponential) number of time steps.

Proof of Theorem 5.2. We prove Theorem 5.2 using a (slightly simplified) "drift theorem" from Hajek [Haj82] (cf. Theorem A.11). As mentioned in Section 5.4, our process is a Markov chain, such that we need to condition only on the previous state (instead of the full filtration from Theorem A.11). Our goal is to bound the load of a fixed bin i at time t using Theorem A.11 and, subsequently, to use this with a union bound to bound the maximum load over all bins. To apply Theorem A.11, we have to prove that the maximum load difference of bin i between two rounds is exponentially bounded (Majorization) and that, given a high enough load, the system tends to lose load (Negative Bias). We start with the majorization. Recall that for random variables X and Y we write $X \leq^{\text{st}} Y$ if X is stochastically dominated by Y, i.e., for all k it holds $\mathbb{P}[\,X \geq k\,] \leq \mathbb{P}[\,Y \geq k\,]$. The load difference $|X_i(t+1) - X_i(t)|$ is bounded by $\max(1, B_i(t)) \leq 1 + B_i(t)$, where $B_i(t)$ is the number of tokens bin i receives during round $t+1$. In particular, $(|X_i(t+1) - X_i(t)| \mid \boldsymbol{X}(t)) \leq^{\text{st}} 1 + B_i(t)$. Note that $B_i(t)$ is binomially distributed with parameters n and λ/n since each of the potential n balls has probability λ to spawn and, given that it spawned, with probability $1/n$ it ends up in bin i. Using standard inequalities we bound

$$\mathbb{P}[\,B_i(t) = k\,] \leq \binom{n}{k} \cdot \left(\frac{\lambda}{n}\right)^k \leq \left(\frac{e \cdot n}{k}\right)^k \cdot \left(\frac{1}{n}\right)^k = \frac{e^k}{k^k} \tag{5.1}$$

and calculate

$$\begin{aligned} \mathbb{E}\left[\, e^{B_i(t)+1} \,\right] &= e \cdot \sum_{k=0}^{n} e^k \cdot \frac{e^k}{k^k} = e \cdot \sum_{k=0}^{\lceil e^3 - 1 \rceil} \frac{e^{2k}}{k^k} + e \cdot \sum_{k=e^3}^{\infty} \frac{e^{2k}}{k^k} \\ &\leq \Theta(1) + \sum_{k=1}^{\infty} e^{-k} = \Theta(1). \end{aligned} \tag{5.2}$$

This shows that the Majorization condition from Theorem A.11 holds (with $\lambda' = 1$ and $D = \Theta(1)$). To see that the Negative Bias condition is also given, note that if bin i has non-zero load, it is guaranteed to delete one ball and receives in expectation $n \cdot \lambda/n = \lambda$ balls. We get $\mathbb{E}[\,X_i(t+1) - X_i(t) | X_i(t) > 0\,] \leq \lambda - 1 < 0$, establishing the Negative Bias condition (with $\varepsilon_0 = 1 - \lambda$). Thus, we can apply Theorem A.11 with $\eta := \min(1, (1-\lambda)/(2D), 1/(2-2\lambda)) = (1-\lambda)/(2D)$ and get for $b \geq 1$

$$\begin{aligned} \mathbb{P}[\,X_i(t) \geq b\,] &\leq e^{-b \cdot \eta} + \frac{2D}{\eta \cdot (1-\lambda)} \cdot e^{\eta \cdot (-b)} \leq \frac{2 \cdot (2D)^2}{(1-\lambda)^2} \cdot e^{\frac{(1-\lambda)\cdot(-b)}{2D}} \\ &\leq \frac{(4D)^2}{(1-\lambda)^2} \cdot e^{\frac{-b\cdot(1-\lambda)}{(4D)^2}} \leq \frac{c}{(1-\lambda)^2} \cdot e^{-\frac{b\cdot(1-\lambda)}{c}}, \end{aligned} \tag{5.3}$$

where $c \geq (4D)^2$ denotes a suitable constant. Applying the Union bound to all n bins and choosing $b \coloneqq \frac{c}{1-\lambda} \cdot \ln(\frac{c \cdot n^{h+1}}{(1-\lambda)^2})$, where $h > 2$ is a constant, yields $\mathbb{P}\Big[\max_{i \in [n]} X_i(t) \geq b\Big] \leq n^{-h}$. Since

$$\begin{aligned} b &= \frac{c}{1-\lambda} \cdot \ln\left(\frac{c \cdot n^{h+1}}{(1-\lambda)^2}\right) \leq \frac{c^2 \cdot (h+1)}{1-\lambda} \cdot \ln\left(\frac{n}{1-\lambda}\right) \\ &= O\left(\frac{1}{1-\lambda} \cdot \ln\left(\frac{n}{1-\lambda}\right)\right), \end{aligned} \tag{5.4}$$

we get the desired statement. □

5.5.2 Stability – Proof of Theorem 5.1

In the following, we provide an auxiliary result that will prove useful for deriving the stability of the 1-Choice process.

Corollary 5.4. *Let $\lambda = \lambda(n) < 1$. Fix an arbitrary round t of the 1-Choice process and a bin i. There is a constant $c > 1$ such that the expected load of bin i is bounded by $\frac{6c}{1-\lambda} \cdot \ln(\frac{n}{1-\lambda})$.*

Proof. By Theorem 5.2, the maximum load of all bins is with high probability bounded by $c \cdot \frac{1}{1-\lambda} \cdot \log \frac{n}{1-\lambda}$, for a sufficiently large constant c. Let

$$\gamma \coloneqq \frac{c}{1-\lambda} \cdot \ln\left(\frac{e \cdot cn}{(1-\lambda)^2}\right). \tag{5.5}$$

Partitioning time into windows of γ rounds and with (5.3), we calculate

$$\begin{aligned} \mathbb{E}[\,X_i(t)\,] &= \sum_{b=1}^{\gamma} b \cdot \mathbb{P}[\,X_i(t) = b\,] + \sum_{k=1}^{\infty} \sum_{b=k\cdot\gamma+1}^{(k+1)\gamma} b \cdot \mathbb{P}[\,X_i(t) = b\,] \\ &\leq \gamma + \sum_{k=1}^{\infty} (k+1)\gamma \cdot \mathbb{P}[\,X_i(t) > k \cdot \gamma\,] \\ &\leq \gamma + \sum_{k=1}^{\infty} (k+1)\gamma \cdot \frac{c}{(1-\lambda)^2} \cdot e^{-\frac{k\cdot\gamma\cdot(1-\lambda)}{c}} \\ &\leq \gamma + \sum_{k=1}^{\infty} (k+1)\gamma \cdot \frac{c}{(1-\lambda)^2} \cdot e^{-k} \cdot e^{-\ln(cn/(1-\lambda)^2)} \\ &\leq \gamma + \sum_{k=1}^{\infty} (k+1)\gamma \cdot e^{-k} \leq 3\gamma \leq \frac{6c}{1-\lambda} \cdot \ln\left(\frac{e \cdot cn}{1-\lambda}\right). \end{aligned} \tag{5.6}$$

This finishes the proof. □

Proof of Theorem 5.1. We prove Theorem 5.1 using a result from Fayolle et al. [FMM95] (cf. Theorem A.22). Note that $\boldsymbol{X}$ is a time-homogeneous irreducible Markov chain with a

countable state space. In the following, let

$$\Delta \coloneqq \frac{12e^2 \cdot c^2 n^2}{(1-\lambda)^3}, \tag{5.7}$$

where c is the constant from Corollary 5.4. For a configuration $\boldsymbol{x}$, we define the auxiliary potential $\Psi(\boldsymbol{x}) \coloneqq \sum_{i=1}^{n} x_i$ as the total system load of configuration $\boldsymbol{x}$. Consider the (finite) set $C \coloneqq \{\, \boldsymbol{x} \mid \Psi(\boldsymbol{x}) \leq n \cdot \Delta \,\}$ of all configurations with not too much load. To prove positive recurrence, it remains to show that item 1 (expected potential drop if not in a high-load configuration) and item 2 (finite potential) of Theorem A.22 hold. Let us start with item 1. Fix a round t and let $\boldsymbol{x} = \boldsymbol{X}(t) \notin C$. By definition of C, we have $\Psi(\boldsymbol{x}) > n \cdot \Delta$. Hence, there is at least one bin i with load $x_i \geq \Psi(\boldsymbol{x})/n > \Delta$. Thus, by definition of the process, during each of the next Δ rounds bin i deletes exactly one ball. On the other hand, bin i receives in expectation $\Delta \cdot \lambda n \cdot \frac{1}{n} = \lambda\Delta$ balls during the next Δ rounds. We get

$\mathbb{E}[\, X_i(t+\Delta) - x_i | \boldsymbol{X}(t) = \boldsymbol{x} \,] = \lambda\Delta - \Delta = -(1-\lambda) \cdot \Delta$. For any bin $j \neq i$, we assume pessimistically that no ball is deleted. Note that the expected load increase of each of these bins can be majorized by the load increase in an empty system running for Δ rounds. Thus, we can use Corollary 5.4 to bound the expected load increase in each of these bins by $\frac{6c}{1-\lambda} \cdot \ln\left(\frac{2 \cdot cn}{1-\lambda}\right) \leq \frac{6e^2 \cdot c^2 \cdot n}{(1-\lambda)^2} = \frac{(1-\lambda)\Delta}{2n}$, by definition of Δ. We get

$$\begin{aligned} \mathbb{E}[\, \Psi(\boldsymbol{X}(t+\Delta)) | \boldsymbol{X}(t) = \boldsymbol{x} \,] &\leq -(1-\lambda) \cdot \Delta + (n-1) \cdot \frac{(1-\lambda)\Delta}{2n} \\ &\leq -\frac{1-\lambda}{2} \cdot \Delta. \end{aligned} \tag{5.8}$$

This proves item 1 of Theorem A.22. For item 2, assume $\boldsymbol{x} = \boldsymbol{X}(t) \in C$. We bound the system load after Δ rounds trivially by

$$\mathbb{E}[\, \Psi(\boldsymbol{X}(t+\Delta)) | \boldsymbol{X}(t) = \boldsymbol{x} \,] \leq \Psi(\boldsymbol{x}) + \Delta \cdot n \leq n \cdot \Delta + \Delta \cdot n < \infty, \tag{5.9}$$

(note that the finiteness in Theorem A.22 is with respect to time, not n). This finishes the proof. □

5.5.3 Lower Bound on Maximum Load - Proof of Theorem 5.3

In expectation, the load of any non-empty bin decreases. Thus, to derive a meaningful lower bound, we need to make use of the variance of the number of balls that are assigned to a bin over a period of suitable length. To do so, we make use of Theorem A.34 (due to Raab and Steger [RS98]; see appendix), which lower-bounds the maximum number of balls a bin receives when m balls are allocated into n bins.

Proof of Theorem 5.3. We assume that we start at an empty system and apply Theorem A.34 to $m \coloneqq \lambda t n$ many balls. The theorem states that, due to the variance, one

of the bins is likely to get more than $c_1\lambda t + c_2\sqrt{t\lambda \log n}$ many balls for suitable constants c_1 and c_2. This allows us to show that the load of this bin is large, even if we assume, pessimistically, that it deletes a ball during each of the t time steps.

Let $M(t')$ be the number of balls allocated during the first $t' \in \mathbb{N}$ steps, and let $Y_{\max}(t')$ be the maximum number of balls allocated to any bin. Set

$$t \coloneqq \frac{9\lambda \log(n)}{64(1-\lambda)^2} \tag{5.10}$$

and let $\varepsilon \coloneqq (1-\lambda)/\lambda$. Since all balls are independent and $\mathbb{E}[\,M(t)\,] = t \cdot \lambda n \geq n \log n$ (due to $\lambda \geq 3/4$), it follows by Chernoff's inequality that

$$\mathbb{P}[\,M(t) \leq (1-\varepsilon)\cdot t \cdot \lambda n\,] \leq e^{-\varepsilon^2 \mathbb{E}[\,M(t)\,]/2} \leq \frac{1}{n^2}. \tag{5.11}$$

By Theorem A.34 Cases 3 and 4 (depending on the size of $1-\lambda$) we get for $\alpha \coloneqq \sqrt{8/9}$ (w.h.p.)

$$\begin{aligned} &Y_{\max}(t) \geq \\ &\geq (1-\varepsilon)\cdot t\cdot\lambda + \sqrt{2(1-\varepsilon)\cdot t\cdot\lambda\log n}\cdot\min\left\{\,\alpha, \sqrt{1-\frac{\log\log n}{2\alpha\log n}}\,\right\} \\ &= (1-\varepsilon)\cdot t\cdot\lambda + \alpha\sqrt{2(1-\varepsilon)\cdot t\cdot\lambda\log n}. \end{aligned} \tag{5.12}$$

Let $X_{\max}(t)$ denote the load of the bin of maximum load. We derive,

$$\begin{aligned} X_{\max}(t) &\geq (1-\varepsilon)\cdot t\cdot\lambda + \sqrt{(1-\varepsilon)\cdot\frac{16}{9}t\cdot\lambda\log n} - t \\ &= (1-\varepsilon)\cdot t\cdot\lambda + \sqrt{\frac{1-\varepsilon}{4}}\cdot\frac{\lambda\log n}{(1-\lambda)} - t \\ &= \sqrt{\frac{1-\varepsilon}{4}}\cdot\frac{\lambda\log n}{(1-\lambda)} - 2(1-\lambda)t \\ &= \sqrt{\frac{1-\varepsilon}{4}}\cdot\frac{\lambda\log n}{(1-\lambda)} - \frac{9\lambda\log(n)}{32(1-\lambda)} \\ &= \left(\sqrt{\frac{1-\frac{1-\lambda}{\lambda}}{4}} - \frac{9}{32}\right)\cdot\frac{\lambda\log n}{(1-\lambda)} = \Omega\Big(\frac{\lambda\log n}{1-\lambda}\Big), \end{aligned} \tag{5.13}$$

where the last inequality holds since $\lambda \geq 3/4$. □

5.6 The 2-Choice Process

We continue with the study of the 2-Choice process. Here, new balls are distributed according to GREEDY[2] (cf. description in Section 5.4). Our main results are the following theorems, which are equivalents to the corresponding theorems for the 1-Choice process.

Theorem 5.5 (Stability). *Let $\lambda = \lambda(n) \in [1/4, 1)$. The Markov chain $\boldsymbol{X}$ of the 2-Choice process is positive recurrent.*

Theorem 5.6 (Maximum Load). *Let $\lambda = \lambda(n) \in [1/4, 1)$. Fix an arbitrary round t of the 2-Choice process. The maximum load of all bins is (w.h.p.) bounded by $O(\log \frac{n}{1-\lambda})$.*

Note that Theorem 5.6 implies a much better behaved system than we saw in Theorem 5.2 for the 1-Choice process. In particular, it allows for an exponentially higher arrival rate: for $\lambda(n) = 1 - 1/\operatorname{poly}(n)$ the 2-Choice process maintains a maximal load of $O(\log n)$. In contrast, for the same arrival rate the 1-Choice process results in a system with maximal load $\Omega(\operatorname{poly}(n))$.

Our analysis of the 2-Choice process relies to a large part on a good bound on the *smoothness* (the maximum load difference between any two bins). This is stated in the following proposition. This result is of independent interest, showing that even if the arrival rate is $\lambda(n) = 1 - e^{-n}$, where we get a polynomial system load, the maximum load difference is still logarithmic.

Proposition 5.7 (Smoothness). *Let $\lambda = \lambda(n) \in [1/4, 1]$. Fix an arbitrary round t of the 2-Choice process. The load difference of all bins is (w.h.p.) bounded by $O(\ln n)$.*

Analysis Overview. To prove these results, we combine three different potential functions: For a configuration $\boldsymbol{x}$ with average load $\varnothing$ and for a suitable constant $\alpha < 1$ (to be fixed later), we define

$$\begin{aligned} \Phi(\boldsymbol{x}) &\coloneqq \sum_{i\in[n]} e^{\alpha\cdot(x_i-\varnothing)} + \sum_{i\in[n]} e^{\alpha\cdot(\varnothing-x_i)}, \qquad \Psi(\boldsymbol{x}) \coloneqq \sum_{i\in[n]} x_i, \quad \text{and} \\ \Gamma(\boldsymbol{x}) &\coloneqq \Phi(\boldsymbol{x}) + \tfrac{n}{1-\lambda}\cdot\Psi(\boldsymbol{x}). \end{aligned} \tag{5.14}$$

The potential Φ measures the *smoothness* (the maximum load difference to the average) of a configuration and is used to prove Proposition 5.7 (Section 5.6.1). The proof is based on the observation that whenever the load of a bin is far from the average load, it decreases in expectation. The potential Ψ measures the *total load* of a configuration and is used, in combination with our results on the smoothness, to prove Theorem 5.6 (Section 5.6.2). The potential Γ entangles the smoothness and total load, allowing us to prove Theorem 5.5 (Section 5.6.3). The proof is based on the fact that whenever Γ is large (i.e., the configuration is not smooth or it has a huge total load), it decreases in expectation.

Before we continue with our analysis, let us make a simple but useful observation concerning the smoothness: For any configuration $\boldsymbol{x}$ and value $b \geq 0$, the inequality $\Phi(\boldsymbol{x}) \leq e^{\alpha \cdot b}$ implies (by definition of Φ) $\max_i |x_i - \varnothing| \leq b$. That is, the load difference of any bin to the average is at most b and, thus, the load difference between any two bins is at most $2b$.

Observation 5.8. *Consider a configuration $\boldsymbol{x}$ with average load $\varnothing$ and let $b \geq 0$. If $\Phi(\boldsymbol{x}) \leq e^{\alpha \cdot b}$, then $|x_i - \varnothing| \leq b$ for all $i \in [n]$. In particular, $\max_i(x_i) - \min_i(x_i) \leq 2b$.*

5.6.1 Bounding the Smoothness – Proof of Proposition 5.7

The goal of this section is to prove Proposition 5.7. The key ingredient for its proof is the following statement: There are values $0 < c < 1$ and $\gamma > 0$ such that

$$\mathbb{E}[\,\Phi(\boldsymbol{X}(t+1))|\boldsymbol{X}(t)\,] \leq c \cdot \Phi(\boldsymbol{X}(t)) + \gamma \tag{5.15}$$

holds for all rounds $t \geq 0$. Once (5.15) is proven, taking the expected value on both sides yields $\mathbb{E}[\,\Phi(\boldsymbol{X}(t+1))\,] \leq c \cdot \mathbb{E}[\,\Phi(\boldsymbol{X}(t))\,] + \gamma$. This recursion is solved by $\mathbb{E}[\,\Phi(\boldsymbol{X}(t))\,] \leq \gamma \cdot (1-c)^{-1}$. In the rest of this section, we prove that (5.15) holds for a constant c and $\gamma = O(n)$, such that we immediately get the following bound on the expected smoothness (potential Φ) at an arbitrary time t:

Lemma 5.9. *Let $\lambda \in [1/4, 1]$. Fix an arbitrary round t of the 2-Choice process. There is a constant $\varepsilon > 0$ such that $\mathbb{E}[\,\Phi(\boldsymbol{X}(t))\,] \leq n/\varepsilon$.*

In Lemma 5.9, we chose $\lambda \in [1/4, 1]$ for convenience; the proof works with minor modifications for any $\lambda = \Theta(1)$ (i.e., for any constant λ, no matter whether $\lambda < 1$ or $\lambda > 1$). Also, our analysis easily adapts to the process without deletions by setting $\lambda = 1$ and $\eta_i(t) = 0$. This yields the same results as [BCE+12] using a simpler analysis.

Proposition 5.7 emerges by combining Observation 5.8, Lemma 5.9, and Markov's inequality:

$$\mathbb{P}\left[\max_i X_i(t) - \min_i X_i(t) \geq \frac{4}{\alpha} \cdot \ln\left(\frac{n}{\varepsilon}\right)\right] \leq \mathbb{P}\left[\Phi(\boldsymbol{X}(t)) \geq \frac{n^2}{\varepsilon^2}\right] \leq \frac{\varepsilon}{n}.$$

It remains to prove (5.15). Our proof follows the lines of [PTW10, TW14][3]. We start by splitting the potential $\Phi(\boldsymbol{x})$ in two parts:

$$\Phi(\boldsymbol{x}) = \Phi_+(\boldsymbol{x}) + \Phi_-(\boldsymbol{x}), \tag{5.16}$$

with the *upper potential* $\Phi_+(\boldsymbol{x}) := \sum_i e^{\alpha \cdot (x_i - \varnothing)}$ and with the *lower potential* $\Phi_-(\boldsymbol{x}) := \sum_i e^{\alpha \cdot (\varnothing - x_i)}$. For a fixed bin i, we use $\Phi_{i,+}(\boldsymbol{x}) := e^{\alpha \cdot (x_i - \varnothing)}$ and $\Phi_{i,-}(\boldsymbol{x}) := e^{\alpha \cdot (\varnothing - x_i)}$ to

[3] Talwar and Wieder [TW14] use the same potential function to analyze variants of the sequential d-Choice process without deletions. Our analysis turns out a bit more involved, since we have to consider deletions and argue over whole batches (of random size) instead of single balls.

denote i's contribution to the upper and lower potential, respectively. When we consider the effect of a fixed round $t+1$, we will sometimes omit the time parameter and use prime notation to denote the value of a parameter at the end of round $t+1$. For example, we write X_i and X_i' for the load of bin i at the beginning and at the end of round $t+1$, respectively.

Two simple but useful identities regarding the potential drops $\Delta_{i,+}(t+1) \coloneqq \Phi_{i,+}(\boldsymbol{X}(t+1)) - \Phi_{i,+}(\boldsymbol{X}(t))$ and $\Delta_{i,-}(t+1) \coloneqq \Phi_{i,-}(\boldsymbol{X}(t+1)) - \Phi_{i,-}(\boldsymbol{X}(t))$ due to a fixed bin i during round $t+1$ are as follows:

Observation 5.10. *Fix a bin i, let K denote the number of balls that are placed during round $t+1$ and let $k \leq K$ be the number of these balls that fall into bin i. Then,*

1. $\Delta_{i,+}(t+1) = \Phi_{i,+}(\boldsymbol{X}(t)) \cdot \left(e^{\alpha\cdot(k-\eta_i(t)-K/n)} - 1\right)$ *and*
2. $\Delta_{i,-}(t+1) = \Phi_{i,-}(\boldsymbol{X}(t)) \cdot \left(e^{-\alpha\cdot(k-\eta_i(t)-K/n)} - 1\right)$.

Proof. Remember that $\mathbf{1}_i$ is an indicator value which equals 1 if and only if the i-th bin is non-empty in configuration $\boldsymbol{X}$. Bin i looses exactly $\mathbf{1}_i$ balls and receives exactly k balls, such that $X_i' - X_i = -\mathbf{1}_i + k$. Similarly, we have $\varnothing' - \varnothing = -\nu + K/n$ for the change of the average load. With the identity $\eta_i = \mathbf{1}_i - \nu$ (see Section 5.4), this yields

$$\begin{aligned} \Delta_{i,+}(t+1) &= e^{\alpha\cdot\left(X_i'-\varnothing'\right)} - e^{\alpha\cdot\left(X_i-\varnothing\right)} \\ &= e^{\alpha\cdot\left(X_i-\varnothing\right)} \cdot \left(e^{\alpha\cdot\left(-\mathbf{1}_i+k+\nu-K/n\right)} - 1\right) = \Phi_{i,+} \cdot \left(e^{\alpha\cdot(k-\eta_i-K/n)} - 1\right), \end{aligned} \tag{5.17}$$

proving the first statement. The second statement follows similarly. □

Preliminaries to Bound the Potential Drop

We now derive the main technical lemma that states general bounds on the expected upper and lower potential change during one round. This will be used to derive different bounds on the potential change depending on the situation (Section 5.6.1). For this, let $p_i \coloneqq \left(\frac{i}{n}\right)^2 - \left(\frac{i-1}{n}\right)^2 = \frac{2i-1}{n^2}$ (the probability that a ball thrown with Greedy[2] falls into the i-th fullest bin). We also define

$$\hat{\alpha} \coloneqq e^{\alpha} - 1 \qquad \text{and} \qquad \check{\alpha} \coloneqq 1 - e^{-\alpha}. \tag{5.18}$$

Note that $\hat{\alpha} \in (\alpha, \alpha+\alpha^2)$ and $\check{\alpha} \in (\alpha-\alpha^2, \alpha)$ for $\alpha \in (0, 1.7)$. This follows from the Taylor approximation $e^x \leq 1 + x + x^2$, which holds for $x \in (-\infty, 1.7]$ (we will use this approximation several times in the analysis). Finally, let

$$\hat{\delta}_i \coloneqq \lambda n \cdot ({}^1\!/_n \cdot 1^- - p_i \cdot {}^{\hat{\alpha}}\!/_{\alpha}) \qquad \text{and} \qquad \check{\delta}_i \coloneqq \lambda n \cdot ({}^1\!/_n \cdot 1^+ - p_i \cdot {}^{\check{\alpha}}\!/_{\alpha}), \tag{5.19}$$

where $1^- \coloneqq 1 - \alpha/n < 1 < 1^+ \coloneqq 1 + \alpha/n$. These $\hat{\delta}_i$ and $\check{\delta}_i$ values can be thought of as upper/lower bounds on the expected difference in the number of balls that fall into bin i

under the 1-Choice and 2-Choice process, respectively (note that 1^+, 1^-, $\hat{\alpha}/\alpha$, and $\check{\alpha}/\alpha$ are all close to 1).

Lemma 5.11. *Consider a bin i after round t and a constant $\alpha \leq 1$.*

1. *For the expected change of i's upper potential during round $t+1$ we have*

$$\frac{\mathbb{E}[\,\Delta_{i,+}(t+1)|\boldsymbol{X}(t)\,]}{\Phi_{i,+}(\boldsymbol{X}(t))} \leq -\alpha \cdot \left(\eta_i + \hat{\delta}_i\right) + \alpha^2 \cdot \left(\eta_i + \hat{\delta}_i\right)^2. \tag{5.20}$$

2. *For the expected change of i's lower potential during round $t+1$ we have*

$$\frac{\mathbb{E}[\,\Delta_{i,-}(t+1)|\boldsymbol{X}(t)\,]}{\Phi_{i,-}(\boldsymbol{X}(t))} \leq \alpha \cdot \left(\eta_i + \check{\delta}_i\right) + \alpha^2 \cdot \left(\eta_i + \check{\delta}_i\right)^2. \tag{5.21}$$

Proof. For the first statement, we use Observation 5.10 to calculate

$$\begin{aligned}
&\mathbb{E}[\,\Delta_{i,+}(t)|\boldsymbol{X}\,]/\Phi_{i,+} = \\
&= \sum_{K=0}^{n}\sum_{k=0}^{K}\binom{n}{K}\binom{K}{k}(p_i\lambda)^k \cdot ((1-p_i)\lambda)^{K-k} \cdot (1-\lambda)^{n-K} \cdot \left(e^{\alpha\cdot(k-\eta_i-K/n)}-1\right) \\
&= \sum_{K=0}^{n}\binom{n}{K}(1-\lambda)^{n-K}\lambda^K \sum_{k=0}^{K}\binom{K}{k} \cdot p_i^k \cdot (1-p_i)^{K-k} \cdot \left(e^{\alpha\cdot(k-\eta_i-K/n)}-1\right) \\
&= \sum_{K=0}^{n}\binom{n}{K}(1-\lambda)^{n-K}\lambda^K \cdot \left(e^{-\alpha(\eta_i+K/n)}\sum_{k=0}^{K}\binom{K}{k}(e^\alpha \cdot p_i)^k(1-p_i)^{K-k}-1\right) \\
&= \sum_{K=0}^{n}\binom{n}{K}(1-\lambda)^{n-K}\lambda^K \cdot \left(e^{-\alpha(\eta_i+K/n)} \cdot (1+\hat{\alpha}\cdot p_i)^K-1\right),
\end{aligned}$$

where we first apply the law of total expectation together with Observation 5.10 and, afterward, twice the binomial theorem. Continuing the calculation using the aforementioned Taylor approximation $e^x \leq 1+x+x^2$ (which holds for any $x \in (-\infty, 1.7]$), and the definition of $\hat{\delta}_i$ yields

$$\begin{aligned}
&= e^{-\alpha\eta_i} \cdot \left(1-\lambda+\lambda e^{-\alpha/n} \cdot (1+\hat{\alpha}\cdot p_i)\right)^n - 1 \\
&\leq e^{-\alpha\eta_i} \cdot \left(1-\lambda(1-e^{-\alpha/n})+\lambda\cdot\hat{\alpha}\cdot p_i\right)^n - 1 \\
&\leq e^{-\alpha\eta_i} \cdot \left(1-\frac{\lambda\cdot\alpha}{n}\cdot(1-\alpha/n)+\lambda\cdot\hat{\alpha}\cdot p_i\right)^n - 1 \\
&\leq e^{-\alpha\eta_i} \cdot \left(1-\frac{\alpha}{n}\cdot\hat{\delta}_i\right)^n - 1 \\
&\leq e^{-\alpha\cdot\left(\eta_i+\hat{\delta}_i\right)} - 1.
\end{aligned}$$

Now, the claim follows by another application of the Taylor approximation. The second statement follows similarly. □

Before we apply Lemma 5.11 to derive different bounds on the potential drop for various situations, we provide three auxiliary claims:

Claim 5.12. *Consider a bin i and the values $\hat{\delta}_i$ and $\check{\delta}_i$ as defined before Lemma 5.11. If $\alpha \leq \ln(10/9)$, then $\max(|\hat{\delta}_i|, |\check{\delta}_i|) \leq 5\lambda/4$.*

Proof. Remember that $\hat{\delta}_i = \lambda n \cdot ({}^1\!/_n \cdot 1^- - p_i \cdot {}^{\hat{\alpha}}\!/_\alpha)$ and $\check{\delta}_i = \lambda n \cdot ({}^1\!/_n \cdot 1^+ - p_i \cdot {}^{\check{\alpha}}\!/_\alpha)$, where $1^- = 1 - \alpha/n < 1 < 1 + \alpha/n = 1^+$ (see proof of Lemma 5.11). Note that if $\alpha \leq \ln(10/9)$, we have $1^+ < 5/4$ and $1^- > 8/9$. Since the p_i are non-decreasing in i, it is sufficient to consider the extreme cases $i = 1$ and $i = n$.

The claims hold trivially for $i = 1$, since $p_1 = 1/n^2$ and both $|{}^1\!/_n \cdot 1^- - p_i \cdot {}^{\hat{\alpha}}\!/_\alpha| \leq 1/n$ and $|{}^1\!/_n \cdot 1^+ - p_i \cdot {}^{\check{\alpha}}\!/_\alpha| \leq 1^+/n$. For the other extreme, $i = n$, we have $p_n \leq 2/n$. From this and the definition of $\hat{\alpha} = e^\alpha - 1$, we get $|\hat{\delta}_i| \leq \frac{5}{4}\lambda$, since $\frac{2}{n} \cdot \frac{\hat{\alpha}}{\alpha} - \frac{1}{n} \cdot 1^- \leq \frac{2}{n} \frac{10/9-1}{\ln(10/9)} - \frac{1}{n} \cdot 1^- < \frac{5}{4n}$. Similarly, $|\check{\delta}_i| \leq \frac{5}{4}\lambda$ follows together with $\frac{2}{n}\frac{\check{\alpha}}{\alpha} - \frac{1}{n} \cdot 1^+ < \frac{1}{n}$ (which holds for any $\alpha > 0$). □

Claim 5.13. *There is a constant $\varepsilon > 0$ such that*

1. $\sum_{i \leq \frac{3}{4}n} p_i \cdot \Phi_{i,+} \leq (1 - 2\varepsilon) \cdot \frac{\Phi_+}{n}$ *and*
2. $\sum_{i \in [n]} p_i \cdot \Phi_{i,-} \geq (1 + 2\varepsilon) \cdot \frac{\Phi_- - \sum_{i \leq \frac{n}{4}} \Phi_{i,-}}{n}$.

Proof. For part1, note that the $\Phi_{i,+}$ are non-increasing in i, that they sum up to Φ_+, and that the p_i are non-decreasing in i. Thus, the left hand side of the claim's first statement is maximized if $\Phi_{i,+} = \frac{4\Phi_+}{3n}$ for all i. Now note that there is a constant ε such that[4] $\sum_{i > 3n/4} p_i \geq \frac{1}{4} + \varepsilon$. We get $\sum_{i \leq 3n/4} p_i \leq \frac{3}{4} - \varepsilon$. With this, the result follows by

$$\sum_{i \leq \frac{3}{4}n} p_i \cdot \Phi_{i,+} \leq \left(\frac{3}{4} - \varepsilon\right) \frac{4\Phi_+}{3n} = \left(1 - \frac{4\varepsilon}{3n}\right) \cdot \Phi_+ \leq (1 - 2\varepsilon) \cdot \frac{\Phi_+}{n}. \tag{5.22}$$

Part 2 follows similarly. □

Claim 5.14. *Consider a round t and a constant $\alpha \geq 0$. Then:*

1. $\sum_{i \in [n]} \alpha\eta_i(\alpha\eta_i - 1) \cdot \Phi_{i,+}(\boldsymbol{X}(t)) \leq \alpha^2\eta\nu \cdot \min(n, \Phi_+(\boldsymbol{X}(t)))$ *and*
2. $\sum_{i \in [n]} \alpha\eta_i(\alpha\eta_i + 1) \cdot \Phi_{i,-}(\boldsymbol{X}(t)) \leq \alpha^2\eta\nu \cdot \Phi_-(\boldsymbol{X}(t))$.

[4] This is easily verified by hand. Alternatively, [TW14, Appendix A] gives $\sum_{i \geq 3n/4} p_i \geq \frac{1}{4} + \varepsilon'$ and the statement follows by noting that $p_{3n/4} = o(1)$.

Proof. For the first statement, we calculate

$$\begin{aligned}
&\sum_{i\in[n]} \alpha\eta_i(\alpha\eta_i - 1)\cdot\Phi_{i,+}(\boldsymbol{X}(t)) \\
&= \sum_{i\le\nu n} \alpha\eta_i(\alpha\eta_i - 1)\cdot\Phi_{i,+}(\boldsymbol{X}(t)) + \sum_{i>\nu n} \alpha\eta_i(\alpha\eta_i - 1)\cdot\Phi_{i,+}(\boldsymbol{X}(t)) \\
&= \alpha\eta(\alpha\eta - 1)\cdot\sum_{i\le\nu n}\Phi_{i,+}(\boldsymbol{X}(t)) + \alpha\nu(1+\alpha\nu)\cdot\sum_{i>\nu n}\Phi_{i,+}(\boldsymbol{X}(t)) \\
&\le \alpha\eta(\alpha\eta - 1)\cdot\nu\cdot\Phi_+(\boldsymbol{X}(t)) + \alpha\nu(1+\alpha\nu)\cdot\eta\cdot\min\bigl(n,\Phi_+(\boldsymbol{X}(t))\bigr) \\
&\le \alpha^2\eta\nu\cdot\min\bigl(n,\Phi_+(\boldsymbol{X}(t))\bigr),
\end{aligned} \tag{5.23}$$

where the first inequality uses that $\Phi_{i,+}(\boldsymbol{X}(t))$ is non-increasing in i and that $\Phi_{i,+}(\boldsymbol{X}(t)) \le 1$ for all $i > \nu n$. The claim's second statement follows by a similar calculation, using that $\Phi_{i,-}(\boldsymbol{X}(t))$ is non-decreasing in i (note that we cannot apply the same trick as above to get $\min\bigl(n,\Phi_-(\boldsymbol{X}(t))\bigr)$ instead of $\Phi_-(\boldsymbol{X}(t))$). □

Bounding the Potential Drop in Different Situations

With these tools in place, we can derive the bounds on the potential drop in different situations. We start with a relative bound on the upper potential change $\Delta_+(t+1) \coloneqq \sum_{i\in[n]}\Delta_{i,+}(t+1)$ and lower potential change $\Delta_-(t+1) \coloneqq \sum_{i\in[n]}\Delta_{i,-}(t+1)$ during round $t+1$, respectively.

Lemma 5.15. *Consider a round t and a constant $\alpha \le \ln(10/9)$ $(< 1/8)$. Let $R \in \{+,-\}$ and $\lambda \in [1/4, 1]$. For the expected upper and lower potential drop during round $t+1$ we have*

$$\mathbb{E}[\,\Delta_R(t+1)|\boldsymbol{X}(t)\,] < 2\alpha\lambda\cdot\Phi_R(\boldsymbol{X}(t)). \tag{5.24}$$

Proof. We prove the statement for $R = +$. The case $R = -$ follows similarly. Using Lemma 5.11 and summing up over all $i \in [n]$ we get

$$\begin{aligned}
\mathbb{E}[\,\Delta_+(t+1)|\boldsymbol{X}\,] &\le \sum_{i\in[n]}\Bigl(-\alpha\cdot(\eta_i+\hat\delta_i) + \alpha^2\cdot(\eta_i+\hat\delta_i)^2\Bigr)\cdot\Phi_{i,+} \\
&= \sum_{i\in[n]}\Bigl(\eta_i\alpha(\eta_i\alpha - 1) + \alpha^2\cdot(2\eta_i\hat\delta_i + \hat\delta_i^2) - \alpha\cdot\hat\delta_i\Bigr)\cdot\Phi_{i,+} \\
&\le \sum_{i\in[n]}\Bigl(\eta_i\alpha(\eta_i\alpha - 1) + 5\alpha^2\lambda + \frac{5}{4}\alpha\lambda\Bigr)\cdot\Phi_{i,+}.
\end{aligned}$$

Here, the last inequality uses $\lambda \le 1$ and $|\hat\delta_i| \le \frac{5}{4}\lambda$ (Claim 5.12). We now apply Claim 5.14, $\nu\eta \le 1/4 \le \lambda$, and $\alpha < 1/8$ to get

$$\mathbb{E}[\,\Delta_+(t)|\boldsymbol{X}\,] \le \Bigl(\alpha^2\lambda + 5\alpha^2\lambda + \frac{5}{4}\alpha\lambda\Bigr)\cdot\Phi_+ < 2\alpha\lambda\cdot\Phi_+, \tag{5.25}$$

the desired statement. □

The next two lemmas derive bounds that are used to bound the upper/lower potential change in reasonably balanced configurations.

Lemma 5.16. *Consider a round t and the constants ε (from Claim 5.13) and $\alpha \leq \min(\ln(10/9), \varepsilon/4)$. Let $\lambda \in [1/4, 1]$ and assume $X_{\frac{3}{4}n}(t) \leq \varnothing(t)$. For the expected upper potential drop during round $t+1$ we have*

$$\mathbb{E}[\,\Delta_+(t+1)|\boldsymbol{X}(t)\,] \leq -\varepsilon\alpha\lambda \cdot \Phi_+(\boldsymbol{X}(t)) + 2\alpha\lambda n. \tag{5.26}$$

Proof. To calculate the expected upper potential change, we use Lemma 5.11 and sum up over all $i \in [n]$ (using similar inequalities as in the proof of Lemma 5.15 and the definition of $\hat{\delta}_i$):

$$\begin{aligned}\mathbb{E}[\,\Delta_+(t+1)|\boldsymbol{X}\,] &\leq 6\alpha^2\lambda \cdot \Phi_+ - \sum_{i\in[n]} \alpha \cdot \hat{\delta}_i \cdot \Phi_{i,+} \\ &= \left(6\alpha^2\lambda - \alpha\lambda \cdot 1^-\right) \cdot \Phi_+ + \hat{\alpha}\lambda n \sum_{i\in[n]} p_i \cdot \Phi_{i,+}.\end{aligned} \tag{5.27}$$

We now use that $\Phi_{i,+} = e^{\alpha\cdot(X_i - \varnothing)} \leq 1$ for all $i > \frac{3}{4}n$ (by our assumption on $X_{\frac{3}{4}n}$). This yields

$$\mathbb{E}[\,\Delta_+(t+1)|\boldsymbol{X}\,] \leq \left(6\alpha^2\lambda - \alpha\lambda \cdot 1^-\right) \cdot \Phi_+ + \hat{\alpha}\lambda n \sum_{i\leq\frac{3}{4}n} p_i \cdot \Phi_{i,+} + 2\alpha\lambda n. \tag{5.28}$$

Finally, we apply Claim 5.13 and the definition of 1^- and $\hat{\alpha}$ to get

$$\begin{aligned}\mathbb{E}[\,\Delta_+(t+1)|\boldsymbol{X}\,] &\leq \left(6\alpha^2\lambda - \alpha\lambda \cdot 1^- + (1-2\varepsilon) \cdot \hat{\alpha}\lambda\right) \cdot \Phi_+ + 2\alpha\lambda n \\ &\leq \left(4\alpha^2\lambda - 2\varepsilon \cdot \alpha\lambda\right) \cdot \Phi_+ + 2\alpha\lambda n.\end{aligned} \tag{5.29}$$

Using $\alpha \leq \varepsilon/4$ yields the desired result. □

Lemma 5.17. *Consider a round t and the constants ε (from Claim 5.13) and $\alpha \leq \min(\ln(10/9), \varepsilon/8)$. Let $\lambda \in [1/4, 1]$ and assume $X_{\frac{n}{4}}(t) \geq \varnothing(t)$. For the expected lower potential drop during round t we have*

$$\mathbb{E}[\,\Delta_-(t+1)|\boldsymbol{X}(t)\,] \leq -\varepsilon\alpha\lambda \cdot \Phi_-(\boldsymbol{X}(t)) + \frac{\alpha\lambda n}{2}. \tag{5.30}$$

Proof. To calculate the expected lower potential change, we use Lemma 5.11 and sum up over all $i \in [n]$ (as in the proof of Lemma 5.16):

$$\begin{aligned}\mathbb{E}[\,\Delta_-(t+1)|\boldsymbol{X}\,] &\leq 6\alpha^2\lambda\cdot\Phi_- + \sum_{i\in[n]}\alpha\cdot\check{\delta}_i\cdot\Phi_{i,-}\\ &= \left(6\alpha^2\lambda+\alpha\lambda\cdot 1^+\right)\cdot\Phi_- - \check{\alpha}\lambda n\sum_{i\in[n]} p_i\cdot\Phi_{i,-}.\end{aligned} \tag{5.31}$$

We now use that $\Phi_{i,-} = e^{\alpha\cdot(\varnothing - X_i)} \leq 1$ for all $i \leq \frac{n}{4}$ (by our assumption on $X_{\frac{n}{4}}$) and apply Claim 5.13 to get

$$\begin{aligned}\mathbb{E}[\,\Delta_-(t)|\boldsymbol{X}\,] &\leq \left(6\alpha^2\lambda+\alpha\lambda\cdot 1^+\right)\cdot\Phi_- - (1+2\varepsilon)\cdot\check{\alpha}\lambda n\cdot\frac{\Phi_- - \frac{n}{4}}{n}\\ &= \left(6\alpha^2\lambda+\alpha\lambda\cdot 1^+ - (1+2\varepsilon)\cdot\check{\alpha}\lambda\right)\cdot\Phi_- + (1+2\varepsilon)\cdot\frac{\check{\alpha}\lambda n}{4}\\ &\leq \left(8\alpha^2\lambda - 2\varepsilon\cdot\alpha\lambda\right)\cdot\Phi_- + \frac{\alpha\lambda n}{2},\end{aligned} \tag{5.32}$$

where the last inequality used the definitions of 1^+, $\check{\alpha}$, as well as $\check{\alpha} > \alpha - \alpha^2$. Using $\alpha \leq \varepsilon/8$ yields the desired result. □

The following two lemmas bound the potential drop in configurations with many balls far below the average to the right and with many balls far above the average to the left.

Lemma 5.18. *Consider a round t and constants $\alpha \leq 1/46$ $(< \ln(10/9))$ and $\varepsilon \leq 1/3$. Let $\lambda \in [1/4, 1]$ and assume $X_{\frac{3}{4}n}(t) \geq \varnothing(t)$ and $\mathbb{E}[\,\Delta_+(t+1)|\boldsymbol{X}(t)\,] \geq -\frac{\varepsilon\alpha\lambda}{4}\cdot\Phi_+(\boldsymbol{X}(t))$. Then, $\Phi_+(\boldsymbol{X}(t)) \leq \frac{\varepsilon}{4}\cdot\Phi_-(\boldsymbol{X}(t))$ or $\Phi(\boldsymbol{X}(t)) = \varepsilon^{-8}\cdot O(n)$.*

Proof. Let $L \coloneqq \sum_{i\in[n]}\max(X_i - \varnothing, 0) = \sum_{i\in[n]}\max(\varnothing - X_i, 0)$ be the "excess load" above and below the average. First note that the assumption $X_{\frac{3}{4}n} \geq \varnothing$ implies $\Phi_- \geq \frac{n}{4}\cdot\exp(\frac{\alpha L}{n/4})$ (using Jensen's inequality). On the other hand, we can use the assumption $\mathbb{E}[\,\Delta_+(t+1)|\boldsymbol{X}\,] \geq -\frac{\varepsilon\alpha\lambda}{4}\cdot\Phi_+$ to show an upper bound on Φ_+. To this end, we use Lemma 5.11 and sum up over all $i \in [n]$ (as in the proof of Lemma 5.16):

$$\begin{aligned}\mathbb{E}[\,\Delta_+(t+1)|\boldsymbol{X}\,] &\leq 6\alpha^2\lambda\cdot\Phi_+ - \sum_{i\in[n]}\alpha\cdot\hat{\delta}_i\cdot\Phi_{i,+}\\ &= 6\alpha^2\lambda\cdot\Phi_+ - \sum_{i\leq\frac{n}{3}}\alpha\cdot\hat{\delta}_i\cdot\Phi_{i,+} - \sum_{i>\frac{n}{3}}\alpha\cdot\hat{\delta}_i\cdot\Phi_{i,+}.\end{aligned} \tag{5.33}$$

For $i \leq n/3$ we have $p_i = \frac{2i-1}{n^2} \leq \frac{2}{3n}$ and, using the definition of 1^- and $\hat{\alpha}$, $\hat{\delta}_i = \lambda n\cdot(1/n\cdot 1^- - p_i\cdot\hat{\alpha}/\alpha) \geq (1-5\alpha)\lambda/3$. Setting $\Phi_{\leq n/3,+} \coloneqq \sum_{i\leq n/3}\Phi_{i,+}$ and $\Phi_{>n/3,+} \coloneqq \sum_{i>n/3}\Phi_{i,+}$,

together with Claim 5.12 this yields

$$\begin{aligned}
&\mathbb{E}[\,\Delta_+(t+1)|\boldsymbol{X}\,] \leq \\
&\leq 6\alpha^2\lambda\cdot\Phi_+ - \frac{\alpha(1-5\alpha)\lambda}{3}\cdot\Phi_{\leq n/3,+} + \frac{5}{4}\alpha\lambda\cdot\Phi_{>n/3,+} \\
&= \left(6\alpha^2\lambda - \frac{\alpha(1-5\alpha)\lambda}{3}\right)\cdot\Phi_+ + \left(\frac{5}{4}\alpha\lambda + \frac{\alpha(1-5\alpha)\lambda}{3}\right)\cdot\Phi_{>n/3,+} \\
&\leq -\frac{\varepsilon\alpha\lambda}{2}\cdot\Phi_+ + 2\alpha\lambda\cdot\Phi_{>n/3,+},
\end{aligned} \tag{5.34}$$

where the last inequality uses $\alpha \leq 1/46 \leq \frac{1}{23} - \frac{3}{46}\varepsilon$. With this, the assumption $\mathbb{E}[\,\Delta_+(t+1)|\boldsymbol{X}\,] \geq -\frac{\varepsilon\alpha\lambda}{4}\cdot\Phi_+$ implies $\Phi_+ \leq \frac{8}{\varepsilon}\cdot\Phi_{>n/3,+} \leq \frac{8}{\varepsilon}\cdot\frac{2n}{3}e^{\frac{\alpha L}{n/3}} = \frac{16n}{3\varepsilon}e^{\frac{3\alpha L}{n}}$ (the last inequality uses that none of the $2n/3$ remaining bins can have a load higher than $L/(n/3)$). To finish the proof, assume $\Phi_+ > \frac{\varepsilon}{4}\cdot\Phi_-$ (otherwise the lemma holds). Combining this with the upper bound on Φ_+ and with the lower bound on Φ_-, we get

$$\frac{16n}{3\varepsilon}e^{\frac{3\alpha L}{n}} \geq \Phi_+ > \frac{\varepsilon}{4}\cdot\Phi_- \geq \frac{\varepsilon n}{16}\cdot e^{\frac{4\alpha L}{n}}. \tag{5.35}$$

Thus, the excess load can be bounded by $L < \frac{n}{\alpha}\cdot\ln\left(\frac{256}{3\varepsilon^2}\right)$. Now, the lemma's statement follows from $\Phi = \Phi_+ + \Phi_- < \frac{5}{\varepsilon}\cdot\Phi_+ \leq \frac{80n}{3\varepsilon^2}e^{\frac{3\alpha L}{n}} = \varepsilon^{-8}\cdot O(n)$. □

Lemma 5.19. *Consider a round t and constants $\alpha \leq 1/32$ ($< \ln(10/9)$) and $\varepsilon \leq 1$. Let $\lambda \in [1/4, 1]$ and assume $X_{\frac{n}{4}}(t) \leq \varnothing(t)$ and $\mathbb{E}[\,\Delta_-(t+1)|\boldsymbol{X}(t)\,] \geq -\frac{\varepsilon\alpha\lambda}{4}\cdot\Phi_-(\boldsymbol{X}(t))$. Then, $\Phi_-(\boldsymbol{X}(t)) \leq \frac{\varepsilon}{4}\cdot\Phi_+(\boldsymbol{X}(t))$ or $\Phi(\boldsymbol{X}(t)) = \varepsilon^{-8}\cdot O(n)$.*

Proof. Let $L \coloneqq \sum_{i\in[n]}\max(X_i - \varnothing, 0) = \sum_{i\in[n]}\max(\varnothing - X_i, 0)$ be the "excess load" above and below the average. First note that the assumption $X_{\frac{n}{4}} \leq \varnothing$ implies $\Phi_+ \geq \frac{n}{4}\cdot e^{\frac{\alpha L}{n/4}}$ (using Jensen's inequality). On the other hand, we can use the assumption $\mathbb{E}[\,\Delta_-(t+1)|\boldsymbol{X}\,] \geq -\frac{\varepsilon\alpha\lambda}{4}\cdot\Phi_-$ to show an upper bound on Φ_-. To this end, we use Lemma 5.11 and sum up over all $i \in [n]$ (as in the proof of Lemma 5.17):

$$\begin{aligned}
\mathbb{E}[\,\Delta_-(t+1)|\boldsymbol{X}\,] &\leq 6\alpha^2\lambda\cdot\Phi_- + \sum_{i\in[n]}\alpha\cdot\check{\delta}_i\cdot\Phi_{i,-} \\
&= 6\alpha^2\lambda\cdot\Phi_- + \sum_{i\leq\frac{2n}{3}}\alpha\cdot\check{\delta}_i\cdot\Phi_{i,-} + \sum_{i>\frac{2n}{3}}\alpha\cdot\check{\delta}_i\cdot\Phi_{i,-}.
\end{aligned} \tag{5.36}$$

For $i \geq 2n/3$ we have $p_i = \frac{2i-1}{n^2} \geq \frac{4}{3n} - \frac{1}{n^2}$. Using this with $p_i \leq p_n \leq 2/n$ and $\check{\alpha} \geq \alpha - \alpha^2$, we can bound $\check{\delta}_i = \lambda n\cdot\left(1/n\cdot 1^+ - p_i\cdot\check{\alpha}/\alpha\right) \leq \lambda\cdot(-1/3 + \frac{1+\alpha}{n}) + 2\alpha\lambda \leq -\lambda/6 + 2\alpha\lambda$. Setting $\Phi_{\leq 2n/3,-} \coloneqq \sum_{i\leq 2n/3}\Phi_{i,-}$ and $\Phi_{>2n/3,-} \coloneqq \sum_{i>2n/3}\Phi_{i,-}$, together with Claim 5.12

this yields

$$\begin{aligned}
&\mathbb{E}[\,\Delta_-(t+1)|\boldsymbol{X}\,] \leq \\
&\leq 6\alpha^2\lambda\cdot\Phi_- + \frac{5}{4}\alpha\lambda\cdot\Phi_{\leq 2n/3,-} - \frac{\alpha\lambda}{6}\cdot\Phi_{>2n/3,-} + 2\alpha^2\lambda\cdot\Phi_{>2n/3,-} \\
&\leq \Big(8\alpha^2\lambda - \alpha\lambda/6\Big)\cdot\Phi_- + \Big(\frac{5}{4}\alpha\lambda + \alpha\lambda/6\Big)\cdot\Phi_{\leq 2n/3,-} \\
&\leq -\frac{\varepsilon\alpha\lambda}{2}\cdot\Phi_- + 2\alpha\lambda\cdot\Phi_{\leq 2n/3,-},
\end{aligned} \tag{5.37}$$

where the last inequality uses $\alpha \leq 1/32 \leq \frac{1}{16} - \frac{1}{48}\varepsilon$. With this, the assumption $\mathbb{E}[\,\Delta_-(t+1)|\boldsymbol{X}\,] \geq -\frac{\varepsilon\alpha\lambda}{4}\cdot\Phi_-$ implies that $\Phi_- \leq \frac{8}{\varepsilon}\cdot\Phi_{\leq 2n/3,-} \leq \frac{8}{\varepsilon}\cdot\frac{2n}{3}e^{\frac{\alpha L}{n/3}} = \frac{16n}{3\varepsilon}e^{\frac{3\alpha L}{n}}$ (the last inequality uses that none of the $2n/3$ remaining bins can have a load higher than $L/(n/3)$). To finish the proof, assume $\Phi_- > \frac{\varepsilon}{4}\cdot\Phi_+$ (otherwise the lemma holds). Combining this with the upper bound on Φ_- and with the lower bound on Φ_+, we get

$$\frac{16n}{3\varepsilon}e^{\frac{3\alpha L}{n}} \geq \Phi_- > \frac{\varepsilon}{4}\cdot\Phi_+ \geq \frac{\varepsilon n}{16}\cdot e^{\frac{4\alpha L}{n}}. \tag{5.38}$$

Thus, the excess load can be bounded by $L < \frac{n}{\alpha}\cdot\ln\Big(\frac{256}{3\varepsilon^2}\Big)$. Now, the lemma's statement follows from $\Phi = \Phi_+ + \Phi_- < \frac{5}{\varepsilon}\cdot\Phi_- \leq \frac{80n}{3\varepsilon^2}e^{\frac{3\alpha L}{n}} = \varepsilon^{-8}\cdot O(n)$. □

Proving (5.15)

With the lemmas from Section 5.6.1, we are finally ready to prove (5.15). More exactly, we argue that for the constant ε from Claim 5.13 and $\alpha \leq \min(1/32, \varepsilon/8)$, for any $\lambda \in [1/4, 1]$ we have

$$\mathbb{E}[\,\Phi(\boldsymbol{X}(t+1))|\boldsymbol{X}(t)\,] \leq \Big(1 - \frac{\varepsilon\alpha\lambda}{4}\Big)\cdot\Phi(\boldsymbol{X}(t)) + \varepsilon^{-8}\cdot O(n). \tag{5.39}$$

This follows via a case analysis analogously to [TW14]:

Case 1: $x_{\frac{n}{4}} \geq \varnothing$ and $x_{\frac{3n}{4}} \leq \varnothing$

The bound follows from Lemma 5.16 and Lemma 5.17.

Case 2: $x_{\frac{n}{4}} \geq x_{\frac{3n}{4}} > \varnothing$

For $\mathbb{E}[\,\Delta_+(t+1)|\boldsymbol{X}(t)\,] \leq \frac{-\varepsilon\alpha\lambda}{4}\cdot\Phi_+$ the results follows from Lemma 5.17. Otherwise, $\mathbb{E}[\,\Delta_+(t+1)|\boldsymbol{X}(t)\,] > \frac{-\varepsilon\alpha\lambda}{4}\cdot\Phi_+$ and Lemma 5.18 yields two subcases:

Case 2.1: $\Phi_+(\boldsymbol{X}(t)) \leq \frac{\varepsilon}{4}\cdot\Phi_-(\boldsymbol{X}(t))$

Using Lemma 5.15 and Lemma 5.17 we obtain

$$\begin{aligned}
&\mathbb{E}[\,\Delta(t+1)|\boldsymbol{X}(t)\,] \leq \\
&\leq 2\alpha\lambda\cdot\Phi_+(\boldsymbol{X}(t)) - \varepsilon\alpha\lambda\cdot\Phi_-(\boldsymbol{X}(t)) + \frac{\alpha\lambda n}{2} \\
&\leq -\frac{\varepsilon\alpha\lambda}{2}\cdot\Phi_-(\boldsymbol{X}(t)) + \frac{\alpha\lambda n}{2} \\
&\leq -\frac{\varepsilon\alpha\lambda}{4}\cdot\Phi(\boldsymbol{X}(t)) + \varepsilon^{-8}\cdot O(n).
\end{aligned} \tag{5.40}$$

Case 2.2: $\Phi(\boldsymbol{X}(t)) = \varepsilon^{-8}\cdot O(n)$

Using Lemma 5.15 we get $\mathbb{E}[\,\Delta(t+1)|\boldsymbol{X}(t)\,] \leq 2\alpha\lambda\varepsilon^{-8}\cdot O(n)$. Our choice of α ($< 1/8$), λ (< 1), and ε ($\ll 1$) yields $2\alpha\lambda \leq (1-\varepsilon\alpha\lambda/4)$. Using the case assumption, we compute

$$\begin{aligned}
\mathbb{E}[\,\Delta(t+1)|\boldsymbol{X}(t)\,] \leq 2\alpha\lambda\varepsilon^{-8}\cdot O(n) &\leq \left(1-\frac{\varepsilon\alpha\lambda}{4}\right)\cdot\varepsilon^{-8}\cdot O(n) \\
&\leq -\frac{\varepsilon\alpha\lambda}{4}\cdot\Phi(\boldsymbol{X}(t)) + \varepsilon^{-8}\cdot O(n).
\end{aligned} \tag{5.41}$$

Case 3: $x_{\frac{3n}{4}} \leq x_{\frac{n}{4}} \leq \varnothing$

Similar to the previous case, for $\mathbb{E}[\,\Delta_-(t+1)|\boldsymbol{X}(t)\,] \leq \frac{-\varepsilon\alpha n}{4}\cdot\Phi_-$ the result follows from Lemma 5.16. For $\mathbb{E}[\,\Delta_-(t+1)|\boldsymbol{X}(t)\,] \geq \frac{-\varepsilon\alpha n}{4}\cdot\Phi_-$, Lemma 5.19 yields two subcases that are proven analogously to Cases 2.1 and 2.2 (using Lemma 5.16 instead of Lemma 5.17).

Thus, all cases lead to (5.39).

5.6.2 Maximum Load – Proof of Theorem 5.6

The goal of this section is to prove Theorem 5.6. Recall the definitions of $\Phi(\boldsymbol{x})$ and $\Psi(\boldsymbol{x})$ from (5.14). For any fixed round t, we will prove that (w.h.p.) $\Psi(\boldsymbol{X}(t)) = O(n\cdot\ln n)$ and that the average load is $\varnothing = O(\ln n)$. Using Union bounds and Proposition 5.7, we see that (w.h.p.) the maximum load at the end of round t is bounded by $\varnothing + O(\ln n) = O(\ln n)$.

It remains to prove a high probability bound on $\Psi(\boldsymbol{X}(t))$ for arbitrary t. To get an intuition for our analysis, consider the toy case $t = \text{poly}(n)$ and assume that exactly $\lambda\cdot n \leq n$ balls are thrown each round. Here, we can combine Observation 5.8 and Lemma 5.9 to bound (w.h.p.) the load difference between any pair of bins and for all $t' < t$ by $O(\ln n)$ (via a union bound over $\text{poly}(n)$ rounds). Given this bound on load difference, we can use the following combinatorial observation (formally stated in Lemma 5.20). If the load distance to the average is bounded by some $b \geq 0$, the bound on the number of balls $\Psi \leq 2b\cdot n$ is invariant under the 2-Choice process, since under our assumptions all bins are non-empty

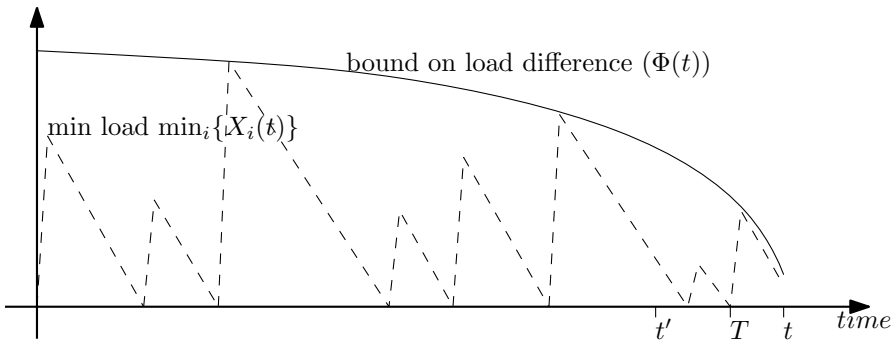


Figure 5.2: To bound the system load at time t, consider the minimum load and our bound on the load difference over time. We consider the last time T when there was an empty bin. The system load can only increase if there is an empty bin, and this increase is bounded by our bound on the load difference. Using that the system load decreases linearly in time while every increase is bounded by our logarithmic bound on the load difference, we find a small interval $[t', t]$ containing T. Due to the monotonic of our bound on Ψ, this will allow us to derive strong bounds on $\Psi(t)$ and on the maximum load.

and thus at least as many balls are deleted as spawn. In particular, we get for $b = O(\ln n)$ that $\Psi(\boldsymbol{X}(t)) \leq 2b \cdot n = O(n \cdot \ln n)$, as required.

The case $t = \omega(\text{poly}(n))$ is considerably more involved. In particular, the fact that the number of balls in the system is only guaranteed to decrease when the total load is high *and* the load distance to the average is low makes it challenging to design a suitable potential function that drops fast enough when it is high. Thus, we deviate from this standard technique and elaborate on the idea of the toy case: Instead of bounding (w.h.p.) the load difference between any pair of bins by $O(\ln n)$ for all $t' < t$ (which is not possible for $t \gg \text{poly}(n)$), we prove (w.h.p.) an *adaptive bound* of $O(\ln(t - t') \cdot f(\lambda))$ for all $t' < t$, where f is a suitable function (Lemma 5.21 and Lemma 5.22). Then we consider the last round $T < t$ with an empty bin. Observation 5.8 yields a bound of $\Psi(\boldsymbol{X}(T)) = 2 \cdot O(\ln(t - T) \cdot f(\lambda)) \cdot n$ on the total load at time T. Using the same combinatorial observation as in the toy case, we get that (w.h.p.) $\Psi(\boldsymbol{X}(t)) \leq \Psi(\boldsymbol{X}(T)) = 2 \cdot O(\ln(t - T) \cdot f(\lambda)) \cdot n$. The final step is to show that the load at time T (the load is is logarithmic in $t - T$) decreases *linearly* in $t - T$, showing that the time interval $[t - T, t]$ cannot be too large (or we would get a negative load at time t). Since the interval $[t - T, t]$ is short, we get a good bound on $\Psi(T)$. Using $\Psi(t) \leq \Psi(T)$ (due to the definition of T) together with the smoothness bounds of Lemma 5.21 yields the claim. See Figure 5.2 for an illustration.

Lemma 5.20. *Let $b \geq 0$ and consider a configuration $\boldsymbol{x}$ with $\Psi(\boldsymbol{x}) \leq 2b \cdot n$ and $\Phi(\boldsymbol{x}) \leq e^{\alpha \cdot b}$. Let $\boldsymbol{x'}$ denote the configuration after one step of the 2-Choice process. Then, $\Psi(\boldsymbol{x'}) \leq 2b \cdot n$.*

Proof. We distinguish two cases: if there is no empty bin, then all n bins delete one ball. Since the maximum number of new balls is n, the number of balls cannot increase. That is, we have $\Psi(\boldsymbol{x'}) \leq \Psi(\boldsymbol{x}) \leq 2b \cdot n$. Now consider the case that there is at least one empty bin. Let $\eta \in (0, 1]$ denote the fraction of empty bins (i.e., there are exactly $\eta \cdot n > 0$ empty bins). Since the minimal load is zero, Observation 5.8 implies $\max_i x_i \leq 2b$. Thus, the

total number of balls in configuration $\boldsymbol{x}$ is at most $(1-\eta)n \cdot 2b$. Exactly $(1-\eta)n$ balls are deleted (one from each non-empty bin) and at most n new balls enter the system. We get $\Psi(\boldsymbol{x'}) \leq (1-\eta)n \cdot 2b - (1-\eta)n + n = (1-\eta)n \cdot (2b-1) + n \leq 2b \cdot n$. □

The next lemma bounds the probability of two events: First, it bounds Φ over an *arbitrary* time interval $[0, t)$ using a union bound over all past rounds $t' < t$. Note that t can be arbitrary large. Thus, in order to get a high probability bound, we must make the bound adaptive and allow for larger errors the further back in time we go. Second, the lemma shows that (w.h.p.) not too many balls are created.

Lemma 5.21. *Let $\lambda \in [1/4, 1)$. Fix a round t. For $i \in \mathbb{N}$ with $t - i \cdot \frac{8 \log n}{1-\lambda} \geq 0$ define $\mathcal{I}_i \coloneqq [t - i \cdot \frac{8 \ln n}{1-\lambda}, t]$. Let Y_i be the number of balls which spawn in $\mathcal{I}_i$.*

1. *Define the (good) smooth event $\mathcal{S}_t \coloneqq \bigcap_{t'<t} \{\, \Phi(\boldsymbol{X}(t')) \leq |t-t'|^2 \cdot n^2 \,\}$. Then, $\mathbb{P}[\,\mathcal{S}_t\,] = 1 - O(n^{-1})$.*
2. *Define the (good) bounded balls event $\mathcal{B}_t \coloneqq \bigcap_i \{\, Y_i \leq \frac{1+\lambda}{2} \cdot |\mathcal{I}_i| \cdot n \,\}$. Then, $\mathbb{P}[\,\mathcal{B}_t\,] = 1 - O(n^{-1})$.*

Proof. Consider an arbitrary time $t' < t$. By Lemma 5.9 we have $\mathbb{E}[\,\Phi(t')\,] \leq n/\varepsilon$. Using Markov's inequality, this implies

$$\mathbb{P}\Big[\,\Phi(t') \geq |t-t'|^2 \cdot n^2\,\Big] \leq 1/(\varepsilon \cdot |t-t'|^2 \cdot n). \tag{5.42}$$

Using the union bound over all $t' < t$ we calculate

$$\mathbb{P}\Big[\,\bar{\mathcal{S}}_t\,\Big] \leq \sum_{t'<t} \mathbb{P}\Big[\,\Phi(t') \geq |t-t'|^2 \cdot n^2\,\Big] \leq \frac{1}{\varepsilon n} \cdot \sum_{t'<t} \frac{1}{|t-t'|^2} \leq \frac{\pi^2}{6\varepsilon \cdot n} = O\Big(n^{-1}\Big),$$

where the last inequality uses the solution to the Basel problem. This proves the first statement.

For the second statement, let $Z_i \coloneqq |\mathcal{I}_i| \cdot n - Y_i$ be the number of balls that did not spawn during $\mathcal{I}_i$. Note that Z_i is a sum of $|\mathcal{I}_i| \cdot n$ independent indicator variables with $\mathbb{E}[\,Z_i\,] = (1-\lambda) \cdot |\mathcal{I}_i| \cdot n = 8i \cdot \ln n$. Chernoff yields $\mathbb{P}[\,Z_i \leq (1-\lambda) \cdot |\mathcal{I}_i| \cdot n/2\,] \leq e^{-8i \cdot \ln n/8} = n^{-i}$. The desired statement follows from applying the identity $Z_i = |\mathcal{I}_i| \cdot n - Y_i$ and taking the union bound. □

Lemma 5.22. *Fix a round t and assume that both $\mathcal{S}_t$ and $\mathcal{B}_t$ hold. Then,*

$$\Psi(\boldsymbol{X}(t)) \leq \frac{9n}{\alpha} \cdot \ln\left(\frac{n}{1-\lambda}\right). \tag{5.43}$$

Proof. Let $T < t$ be the last time when there was an empty bin and set $\Delta \coloneqq t - T$. Note that T is well defined, as we have $X_i(0) = 0$ for all $i \in [n]$. Since $\mathcal{S}_t$ holds, we have

$$\Phi(\boldsymbol{X}(T)) \leq \Delta^2 \cdot n^2 = \exp\Big(\ln(\Delta^2 \cdot n^2)\Big). \tag{5.44}$$

By definition of T, we have $\min_i X_i(T) = 0$. Together with Observation 5.8 we get

$$\max_i X_i(T) \leq 2\ln(\Delta^2 \cdot n^2)/\alpha. \tag{5.45}$$

Summing up over all bins (and pulling out the square), this implies that $\Psi(\boldsymbol{X}(T)) \leq 4n \cdot \ln(\Delta \cdot n)/\alpha$. Applying Lemma 5.20 yields

$$\Psi(\boldsymbol{X}(T+1)) \leq 4n \cdot \ln(\Delta \cdot n)/\alpha. \tag{5.46}$$

By the definition of T, is must be the case that there is no empty bin in $\boldsymbol{X}(t'')$ for all $t'' \in \{T+1, T+2, \ldots, t-1\}$. Thus, during each of these rounds exactly n balls are deleted. To bound the number of deleted balls, let i be maximal with $\mathcal{I}_i \subseteq [T,t]$ (as defined in Lemma 5.21). Recall that $\mathcal{I}_i = [t - i \cdot \frac{8\ln n}{1-\lambda}, t]$. Since $\mathcal{B}_t$ holds and using the maximality of i, the number of balls Y that spawn during $[T,t]$ is bounded by

$$(1+\lambda)|\mathcal{I}_i| \cdot n/2 + \frac{8\ln n}{1-\lambda} \cdot n \leq (1+\lambda)\Delta \cdot n/2 + \frac{8\ln n}{1-\lambda} \cdot n. \tag{5.47}$$

We calculate

$$\begin{aligned}
\Psi(\boldsymbol{X}(t)) &\leq \Psi(\boldsymbol{X}(T+1)) - \Delta \cdot n + Y \\
&\leq \frac{4n}{\alpha}\ln(\Delta \cdot n) - \frac{1-\lambda}{2}\Delta \cdot n + \frac{8\ln n}{1-\lambda} \cdot n \\
&= \frac{1-\lambda}{2} \cdot n \cdot \left(\frac{8}{\alpha(1-\lambda)} \cdot \ln(\Delta \cdot n) - \Delta + \frac{16\ln n}{(1-\lambda)^2}\right) \\
&\leq \frac{1-\lambda}{2} \cdot \Delta \cdot n \cdot \left(\frac{24}{\alpha(1-\lambda)^2} \cdot \frac{\ln(\Delta \cdot n)}{\Delta} - 1\right).
\end{aligned} \tag{5.48}$$

With $f = f(\lambda) \coloneqq 24/(\alpha(1-\lambda)^2)$ the last factor becomes $f \cdot \ln(\Delta \cdot n)/\Delta - 1$. It is negative if and only if $\Delta > f \cdot \ln(\Delta \cdot n)$. This inequality holds for any $\Delta > -f \cdot W_{-1}(-\frac{1}{f \cdot n})$, where W_{-1} denotes the lower branch of the Lambert W function[5]. This implies that $\Delta \leq -f \cdot W_{-1}(-\frac{1}{f \cdot n})$, since otherwise we would have $\Psi(\boldsymbol{X}(t)) < 0$, which is clearly a contradiction. Using the Taylor approximation $W_{-1}(x) = \ln(-x) - \ln(\ln(-1/x)) - o(1)$ as $x \to -0$, we get

$$\Delta \leq -f \cdot W_{-1}\left(-\frac{1}{f \cdot n}\right) \leq f \cdot \ln(f \cdot n) + f \cdot \ln(\ln(f \cdot n)) + f \leq 2f \cdot \ln(f \cdot n). \tag{5.49}$$

[5]Note that $-\frac{1}{f \cdot n} \geq -1/e$, so that $W_{-1}(-\frac{1}{f \cdot n})$ is well defined.

Finally, we use this bound on Δ to get

$$\begin{aligned}\Psi(\boldsymbol{X}(t)) \leq \Psi(\boldsymbol{X}(T+1)) &\leq \frac{4n}{\alpha} \cdot \ln(\Delta \cdot n) \leq \frac{4n}{\alpha} \cdot \ln(2fn \cdot \ln(fn)) \\ &\leq \frac{4n}{\alpha} \cdot \ln\left(\frac{48n}{\alpha(1-\lambda)^2} \cdot \ln\left(\frac{24n}{\alpha(1-\lambda)^2}\right)\right) \leq \frac{9n}{\alpha} \cdot \ln\left(\frac{n}{1-\lambda}\right).\end{aligned} \tag{5.50}$$

□

Now, by combining Lemma 5.22 with the fact that the events $\mathcal{S}_t$ and $\mathcal{B}_t$ hold with high probability (Lemma 5.21), we immediately get that (w.h.p.) $\Psi(\boldsymbol{X}(t)) = O(n \cdot \ln n)$. As described at the beginning of this section, combining this with Proposition 5.7 proves Theorem 5.6.

5.6.3 Stability – Proof of Theorem 5.5

This section proves Theorem 5.5. In order to do so, we consider the potential Γ (defined in (5.14)) and show that, for a sufficiently high value of, this potential decreases (Lemma 5.23).[6] To show this drop, we argue along the following lines. For the potential to be large and since the potential is the sum of two potentials Φ and Ψ, one of must have size at least $\Gamma(\boldsymbol{x})/2$. If $\Phi(\boldsymbol{x})$ is large, then we can even assume a worst-case increase of Ψ and invoke (5.39) to show that Φ drops considerably resulting in an overall potential drop of Γ. Similarly, if $\Psi(\boldsymbol{x}) \geq \Gamma(\boldsymbol{x})/2$, then, due to the careful construction of Γ, we can show that all bins are non-empty, and the overall potential decreases in expectation. This overall potential decrease of Γ allows to apply Theorem A.22 yielding stability.

Lemma 5.23 (Negative Bias Γ)**.** *Let $\lambda \in [1/4, 1)$. If $\Gamma(\boldsymbol{X}(t)) \geq 2\frac{n^4}{(1-\lambda)^2\lambda}$, then*

$$\mathbb{E}[\,\Gamma(\boldsymbol{X}(t+1)) - \Gamma(\boldsymbol{X}(t)) | \boldsymbol{X}(t)\,] \leq -1. \tag{5.51}$$

Proof. Assume $\boldsymbol{X}(t) = x$ is fixed. By definition of $\Gamma(\cdot)$, we have $\Phi(x) \geq \Gamma(x)/2$ or $\Psi(x) \geq \Gamma(x)/2$. We now show that in both cases

$$\mathbb{E}[\,\Gamma(\boldsymbol{X}(t+1)) - \Gamma(x) | \boldsymbol{X}(t) = \boldsymbol{x}\,] \leq -1. \tag{5.52}$$

1. If $\Phi(\boldsymbol{x}) \geq \Gamma(\boldsymbol{x})/2$, then we have, by (5.39), a potential drop of

$$\begin{aligned}\mathbb{E}[\,\Phi(\boldsymbol{X}(t+1)) - \Phi(\boldsymbol{x}) | \boldsymbol{X}(t) = \boldsymbol{x}\,] &\leq -(\varepsilon\alpha\lambda/4) \cdot \Phi(\boldsymbol{x}) + n \log n \\ &\leq -(\varepsilon\alpha\lambda/8) \cdot \Gamma(\boldsymbol{x}) + n \log n.\end{aligned} \tag{5.53}$$

[6] It might look tempting to use Γ together with Hajek's theorem to bound the maximum system load. However, this would require (exponentially) sharper bounds on Φ. Furthermore, it might be tempting to use the stability of Greedy[1] to prove stability of Greedy[2], however, as discussed earlier, it is not clear to achieve this, as it seems challenging to couple or majorize the processes.

Note that, by definition of Ψ, $\Psi(\boldsymbol{X}(t+1)) - \Psi(\boldsymbol{x}) \leq n$. Together with $\Gamma(\boldsymbol{x}) \geq \frac{8(n\log n + n^2/(1-\lambda)+1)}{e\alpha\lambda}$,

$$\begin{aligned} &\mathbb{E}[\,\Gamma(\boldsymbol{X}(t+1)) - \Gamma(\boldsymbol{x}) | \boldsymbol{X}(t) = \boldsymbol{x}\,] \\ \leq\ & -\frac{\varepsilon\alpha\lambda}{8}\Gamma(\boldsymbol{x}) + n\log n + (n/(1-\lambda))\cdot n \leq -1. \end{aligned} \tag{5.54}$$

2. Otherwise, i.e., if $\Phi(\boldsymbol{x}) < \Gamma(\boldsymbol{x})/2$, we have that

   (i) the load difference is, by Observation 5.8, bounded by $2\ln(\Gamma(\boldsymbol{x})/2)/\alpha$, and

   (ii) $\Psi(\boldsymbol{x}) \geq \Gamma(\boldsymbol{x})/2$ must hold. This implies that $\varnothing \geq \frac{1}{n}\left(\frac{\Gamma(\boldsymbol{x})/2}{\frac{n}{1-\lambda}}\right) = \frac{(1-\lambda)\cdot\Gamma(\boldsymbol{x})}{2n^2}$.

   From (i) and (ii) we have that the minimum load is at least $\frac{(1-\lambda)\cdot\Gamma(\boldsymbol{x})}{2n^2} - \ln(\Gamma(\boldsymbol{x})/2)/\alpha$. From Lemma 5.24 and $\Gamma(\boldsymbol{x}) \geq 2\frac{n^4}{(1-\lambda)^2\lambda}$, it follows that every bin has load at least load 1. Thus each bin will delete one ball and the number of balls arriving is λn in expectation. Hence,

$$\mathbb{E}[\,\Psi(\boldsymbol{X}(t+1)) - \Psi(\boldsymbol{x}) | \boldsymbol{X}(t) = \boldsymbol{x}\,] = -\frac{n}{1-\lambda}(1-\lambda)n. \tag{5.55}$$

   Now,

$$\begin{aligned} &\mathbb{E}[\,\Gamma(\boldsymbol{X}(t+1)) - \Gamma(\boldsymbol{x}) | \boldsymbol{X}(t) = \boldsymbol{x}\,] \\ &= \mathbb{E}[\,\Phi(\boldsymbol{X}(t+1)) - \Phi(\boldsymbol{x}) | \boldsymbol{X}(t) = \boldsymbol{x}\,] - \frac{n}{1-\lambda}(1-\lambda)n \\ &\leq n\log n - \frac{n}{1-\lambda}(1-\lambda)n \leq -1. \end{aligned} \tag{5.56}$$

Thus, $\mathbb{E}[\,\Gamma(\boldsymbol{X}(t+1)) - \Gamma(\boldsymbol{x}) | \boldsymbol{X}(t) = \boldsymbol{x}\,] \leq -1$, which yields the claim. □

We now proceed with a technical result.

Lemma 5.24. *For all $x \geq 2\frac{n^4}{(1-\lambda)^2\lambda}$ it holds that $\frac{(1-\lambda)\cdot x}{2n^2} - 2\ln(x/2)/\alpha \geq 1$.*

Proof. Define $f(x) = \frac{(1-\lambda)\cdot x}{2n^2} - 2\ln(x/2)/\alpha$. We have $f\left(2\frac{n^4}{(1-\lambda)^2\lambda}\right) \geq \frac{n^2}{(1-\lambda)\lambda} - \frac{2}{\alpha}\ln\left(\frac{n^4}{(1-\lambda)^2\lambda}\right) \geq 1$, where the last inequality holds for large enough of n since α is a constant. Moreover, for all $x \geq 2\frac{n^4}{(1-\lambda)^2\lambda}$ we have $f'(x) = \frac{1-\lambda}{n^2} - \frac{2}{\alpha x} \geq 0$. Thus, the claim follows. □

We are ready to prove Theorem 5.5.

Proof of Theorem 5.5. The proof proceeds by applying Theorem A.22. We now define the parameters of Theorem A.22. Let $\zeta(t) = \boldsymbol{X}(t)$ and hence Ω is the state space of X. First we observe that Ω is countable since there are a constant number of bins (n is consider a constant in this matter) each having a load which is a natural number. We define $\phi(\boldsymbol{X}(t))$

to be $\Gamma(\boldsymbol{X}(t))$. We define $C = \{\, x \mid \Gamma(x) \leq 2\frac{n^4}{(1-\lambda)^2\lambda} \,\}$. Define $\beta(x) = 1$ and $\eta = 1$. We now show that the preconditions (a) and (b) of Theorem A.22 are fulfilled.

- Let $x \notin C$. By definition of C and $\phi(\boldsymbol{X}(t))$, and from Lemma 5.23 we have

$$\begin{aligned} &\mathbb{E}[\,\phi(X(t+1)) - \phi(x) | \boldsymbol{X}(t) = x\,] \\ \leq\ &\mathbb{E}[\,\Gamma(\boldsymbol{X}(t+1)) - \Gamma(x) | \boldsymbol{X}(t) = x\,] \leq -1. \end{aligned} \tag{5.57}$$

- Let $x \in C$. Recall that $\Gamma(\boldsymbol{X}(t)) = \Phi(\boldsymbol{X}(t)) + \Psi(\boldsymbol{X}(t))$. By Lemma 5.19 and the fact that the number of balls arriving in one round is bounded by n, we derive,

$$\begin{aligned} &\mathbb{E}[\,\phi(X(t+1)) | \boldsymbol{X}(t) = x\,] = \\ =\ &\mathbb{E}[\,\Phi(\boldsymbol{X}(t+1)) | \boldsymbol{X}(t) = x\,] + \mathbb{E}[\,\Psi(\boldsymbol{X}(t+1)) | \boldsymbol{X}(t) = x\,] \\ \leq\ &\left(\left(1 - \frac{\varepsilon\alpha\lambda}{4}\right)2\frac{n^4}{(1-\lambda)^2\lambda}\right) + \frac{n}{1-\lambda}n < \infty. \end{aligned} \tag{5.58}$$

The claim follows by applying Theorem A.22 with (5.57) and (5.58). □

Chapter 6

Future Work - Dynamic Processes

Apart from the variants of the Forest Fire Process discussed in Section 4.7 we suspect that there are many social networks and other dynamic processes that can be studied by using the potential approach of Chapter 3 and similar ideas and methods as we used for the study of Forest Fire Process and Balls-into-Bins with Deletions.

It would be interesting to develop some further notion of what it means that a process is *self-stabilizing* in the sense that whenever the process is in a "bad" state (*e. g.,* the distance is super-constant), then it quickly recovers. This notion of self-stabilization is different than the related the notion of positive recurrence of the underlying Markov chain we use in Chapter 5 since we would like the notion to include the distance in the Forest Fire Process—however, the underlying state space is growing since the graph is growing and the underlying Markov chain is not positive recurrent. Nevertheless, as our analysis of the Forest Fire Process shows, whenever the potential maximizing the distance is large, then it decreases in expectation. It would be interesting to identify necessary properties of such self-stabilizing processes—what's the common thread about self-stabilizing processes?

Another interesting research direction consists of deriving lower bounds complementing Theorem A.11 and to understand how tight the results of Theorem A.11 are—can the first condition (the bound on all moments), be relaxed?

One of the the technically most challenging task consists of deriving tools to deal with potentials that change in more convoluted ways: We studied the Forest Fire Process for the case without backward burning since with backward burning the ϕ of the node arrive at time t changes by nodes arriving after time t. We are not aware of any tools that would permit the analysis of such a potential and we believe that such tools would allow us to study much more involved processes—rendering the potential analysis of dynamic processes even more powerful.

Part II

Probabilistic Analysis of Consensus Dynamics and Protocols

Chapter 7

Contributions Consensus Processes

Since reaching consensus is impossible for deterministic distributed algorithm in many settings of interest (*e. g.,* [Ang80, Lyn89]), we focus on randomized *consensus* processes executing simple algorithms. The system consists of n anonymous nodes connected by edges of a graph. Initially, each node supports one opinion from the set $[k] \coloneqq \{1, \ldots, k\}$. We refer to these opinions as *colors*. We describe the system state after any round by an n-dimensional integral vector $\mathbf{c} = (c_i)_{i \in [n]} \in \mathbb{N}_0^n$ with $\sum_{i \in [n]} c_i = n$. Here, the i-th component $c_i \in \mathbb{N}_0$ corresponds to the number of nodes supporting opinion i. If $k < n$, then $c_i = 0$ for all $i \in \{k+1, k+2, \ldots, n\}$. A consensus process is specified by an *update rule* that is executed by each node. The goal is to reach a state in which all nodes support the same color; the special case where nodes start with pairwise distinct colors is *leader election*, an important primitive in distributed computing. The quantity of interest is the *consensus time* which is the expected time of a consensus process to reach consensus. In some cases we are also interested in *plurality consensus* in which the goal is that all nodes agree on the initially most dominant color.

We assume a severely restricted and simple communication mechanism known as *Uniform Pull* [DGH+87, KSSV00, KDG03]. Here, in each discrete round, nodes independently pull information from some (typically constant) number of randomly sampled nodes. We are interested in protocols with low memory and low message size.

We distinguish between *Consensus Dynamics* and *Consensus Protocols*: Consensus Dynamics are a subset of Consensus Protocols in which the message size and memory is bounded by $O(\log k)$–the memory is enough to share a constant number of opinions but no more. Typically, consensus dynamics are simple and memoryless. Consensus protocols on the other side can be more complex: they are trading off additional memory and larger message sizes against faster consensus or better guarantees on the plurality consensus (*i. e.,* they require a smaller initial bias to ensure that plurality color wins).

7.1 Consensus Dynamics

We start by giving an overview over the most studied protocols in this class. The Voter process (also known as Polling) uses the most naïve update rule: In every round, with probability 1/2, each node samples one neighbor independently and uniformly at random, and it adopts that node's color. Otherwise, the node keeps its current color. We will also consider the non-lazy version of the model, where every node samples one neighbor in every round.

Two further natural and prominent consensus processes are the *2-Choices* and the 3-*Majority* processes. Their corresponding update rules, executed synchronously by every node, are as follows:

- 2-Choices: Sample two neighboring nodes independently and uniformly at random. If the samples have the same color, adopt it. Otherwise, ignore them and keep the current color.
- 3-Majority: Sample three neighboring nodes independently and uniformly at random. If a color is supported by at least two samples, adopt it. Otherwise, adopt the color of one of them at random[1].

See Figure 7.1 for an illustration.

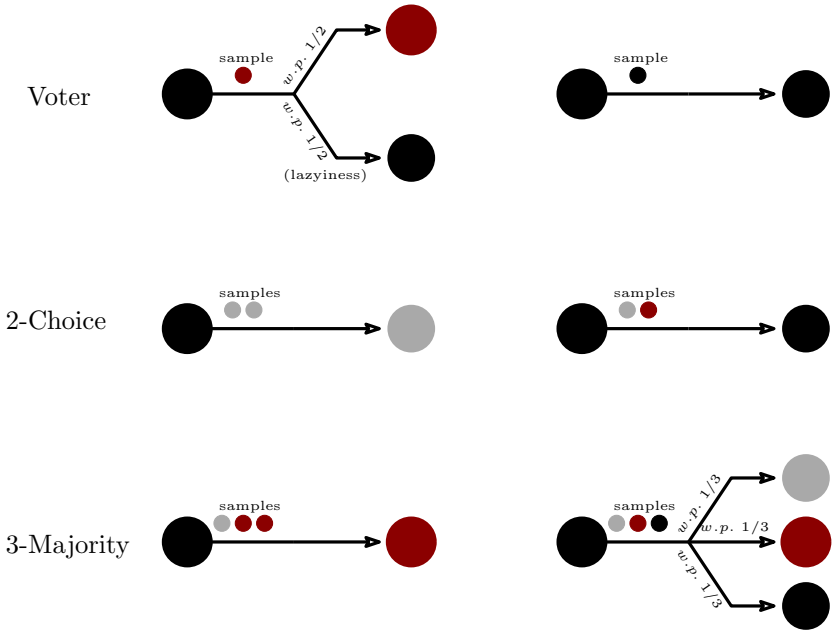


Figure 7.1: The figure depicts the three models with two examples each.

In the following we give an overview of our results in these models.

[1]Equivalently, the node may adopt the color of a fixed sample (the first, or second, or third).

7.1.1 Overview of Results

We start by considering the Voter model on general undirected graphs.

Results Voter model. The Voter is the dual[2] of the so-called *coalescing random walk* model, which is a fundamental stochastic process on *connected* and *undirected* graphs: At the beginning of the process there is one particle on each node in the graph. At discrete time steps, every particle performs independently one step of a random walk.[3] Whenever two or more particles arrive at the same node at the same time step, they merge into a single particle and continue as a single random walk. We are interested in the (expected) first time step when only one particle remains, to which we refer to as *coalescence time*.

The aforementioned duality (see [HP01, AF02] and Proposition 9.11 (more rigorously)) states that the Voter process viewed backwards is exactly the same as the coalescence process starting with a random walk on every node; thus, the coalescence time t_{coal} and consensus time t_{V} have the same distribution. In other words, a bound on the coalescence time yields a bound on consensus time and vice versa.

When starting with two particles, the coalescence time is referred to as the *meeting time*: we denote by t_{meet} the worst-case expected meeting time over all pairs of starting nodes and let t_{coal} denote the expected coalescence time starting from one particle on every node. Surprisingly, little is known about the relationship between t_{meet} and t_{coal}.

Question 1: What are the bounds for the expected consensus time t_{coal} in terms of the meeting time t_{meet}?

Question 2: For which graphs do we have $t_{\text{coal}} = \Theta(t_{\text{meet}})$?

The first question can partially be answered by

$$t_{\text{coal}} \in [t_{\text{meet}}, O(t_{\text{coal}} \log n)]. \tag{7.1}$$

The lower bound follows my means of a coupling argument stating that it takes for all n walks at least as long to meet as for 2 random walks. The upper bound follows by dividing time into periods of length $2t_{\text{meet}}$: Say we are left with k random walks, then any pair of random walks meets after $2t_{\text{meet}}$ time steps w.p. at least $1/2$, by Markov inequality.

[2] The meaning of "dual" in this field differs from usage in other fields. As we elaborate later, the duality states that either of the processes can be viewed as the other the exception that time runs "backwards". This is made rigorously in Proposition 9.11.

[3] Throughout this thesis, we use random walk and particle interchangeably, assuming that every random walk has an identifier.

Repeating this argument shows that after $O(\log n)$ periods only one walk prevails (see Proposition 8.6).

In Chapter 8, we generalize (7.1) and give at the same time sufficient conditions in answer to the second question. More precisely, we relate t_{coal} to the ratio of the mixing time t_{mix} and t_{meet}: Whenever t_{meet} is marginally larger than t_{mix} (a factor of $\log^2 n$), then $t_{\text{coal}} = \Theta(t_{\text{meet}})$. This result as well as (7.1) are special cases of our more general bound covering the entire spectrum $t_{\text{meet}}/t_{\text{mix}} \in [1, \log^2 n]$. We show that (Theorem 8.1), using the duality $t_{\text{V}} = t_{\text{coal}}$

$$\boxed{t_{\text{V}} = t_{\text{coal}} = O\left(t_{\text{meet}}\left(1 + \sqrt{\frac{t_{\text{mix}}}{t_{\text{meet}}}} \cdot \log n\right)\right).}$$

We complement this by giving a matching lower bound (Theorem 8.2).

While the Voter model is very natural and useful in many settings–due to its simplicity, the Voter model has two downsides: 1) the consensus time on the clique is fairly large $\Theta(n)$ and 2) the probability for an opinion to win is proportional to the sum of the degrees of nodes of that color ([HP01]) and thus even with a large bias the second most dominant color can easily win, which is in certain settings intolerable. To circumvent both issues two other simple dynamics were proposed: 3-Majority and 2-Choices.

Results 3-Majority. All known results for 3-Majority (and 2-Choices) apply only if the number of opinions is small thus not solving the leader election problem. Most results require–in addition to a limited number of opinions–a sufficient bias between the largest and second largest color. The reason for the lack of more general results is the following. While it it easy to apply first moment analysis, it seems challenging to analyze the true behavior of the process, due to the considerable variance. Intuitively, in the difficult regime where the number of opinions exceeds $n^{2/3}$, the process behaves similarly to the Voter in the sense that majorities shift easily, rendering many standard approaches futile. We embrace this resemblance and show a stochastic majorization between the processes (Proposition 9.10) which allows us to analyze the challenging regime by reducing to the easy-to-analyze Voter model. By doing so we obtain the first unconditional results on the clique: We show (Theorem 9.8) that the expected consensus time is bounded by

$$\boxed{t_{\text{3M}} = O(n^{3/4} \log^{7/8} n).}$$

This stochastic majorization is a consequence of our general result (Theorem 9.4) whose essence is the following. We define a potential ϕ measuring the progress towards consensus, and a family of memoryless processes (AC-processes defined in Definition 9.2) comprising 3-Majority and the non-lazy voter model. For any two process P, P' of this family we show

that P' reaches consensus faster than P, *i. e.,* $T_P \leq^{\text{st}} T_{P'}$ if the following holds. For every pair of color distributions $\mathbf{c}, \mathbf{c}'$ such that $\phi(\mathbf{c}') \leq \phi(\mathbf{c})$ after one-step P remains closer towards consensus than P', *i. e.,* $\mathbb{E}[\,\phi(P'(\mathbf{c}'))\,] \leq \mathbb{E}[\,\phi(P(\mathbf{c}))\,]$. These requirements are satisfied by Voter and 3-Majority (Proposition 9.10) establishing the stochastic majorization between the two processes. Therefore, we use the following two-state approach: We can use our reduction from 3-Majority to Voter in order to reduce the number of opinion until we are in a regime where the known results apply. From there one we simply use the known results to reduce to one opinion.

Results 2-Choices. For the 2-Choices dynamics we generalize to the setting of $k > 2$ colors and highlight the advantage of 2-Choices (in comparison to 3-Majority) when it comes to plurality consensus (converging towards the initially most dominant color) in the self-stabilizing setting, *i. e.,* in presence of adversaries. We show for the complete graph (Theorem 10.2) starting with $k = \mathrm{O}(n^{\varepsilon})$ opinions for some small constant $\varepsilon > 0$ and for a sufficient bias (between the most and second most dominant color) of $\Omega(\sqrt{n \log n})$ that 2-Choices converges with high probability[4] to the initially most dominant color within

$$\boxed{t_{2\mathrm{C}} = \mathrm{O}(n/c_1 \cdot \log n)}$$

rounds even in presence of an adversary which can change up to $c_1(c_1 - c_2)/n$ nodes per round, where c_i is the size of the i'th largest opinion.

Furthermore, we prove a series of lower bounds which show that leader election takes $\Omega(n/\log n)$ rounds (Theorem 10.5), and that plurality consensus is likely to fail if the initial bias is smaller (w.r.t. the bias of our upper bound) by a factor of $\sqrt{\log n}$ (Theorem 10.6).

At first glance 2-Choices and 3-Majority appear to very similar: Under the first moment method both processes behave identically meaning that when started from the same configuration the expected sized of all colors after one round are identical. Nevertheless, we show stark differences between processes resulting in an interesting trade-off: 2-Choices trades a (considerably) worst-case slower consensus time for better guarantees on the plurality color. These strong plurality guarantees prove to be very useful in the design of consensus protocols as we show by using 2-Choices as the core of our fast protocol RapidPluralityConsensus.

7.2 Consensus Protocols

The goal of our consensus protocols is not only to reach consensus more efficiently, but also to relax the synchronicity requirement, *i. e.,*, the requirement that all nodes perform

[4]Throughout this thesis, the expression *with high probability* means a probability of at least $1 - n^{-\Omega(1)}$.

synchronous rounds. Instead, we would like our algorithm to work even in settings such the asynchronous model, sequential model.

7.2.1 Overview of Results

We consider three protocols RapidPluralityConsensus as well as two protocols inspired by load balancing: Balance and Shuffle.

Results RapidPluralityConsensus. The key to our fast plurality protocol is the combination of the 2-Choices process with an information dissemination process. We divide time into phases: A 2-Choices sub-phase and a dissemination sub-phase. The first sub-phase is just one round of 2-Choices and every node that changed its color propagates the adopted color in the second sub-phase via a process which is essentially pull rumor spreading with multiple rumors (representing the adopted colors).

While this process is not too difficult to analyze in the synchronous model it becomes very challenging in the asynchronous (continuous) setting or sequential settings in which the nodes are selected u.a.r. to perform a tick (see Chapter 11). The reason for the arising difficulties is that the aforementioned protocol relies heavily on simultaneously execution of the phases which is no longer the case in the asynchronous realm.

Nevertheless, we are able to adapt the aforementioned protocol to obtain (Theorem 11.1) a plurality consensus time[5] of

$$\boxed{t_{\text{async}} = O(\log n).}$$

Incidentally, this is the best possible consensus time since some nodes will not have even ticked once after $\Omega(\log n)$ and have therefore not even queried a single. To make our algorithm work, we develop the following weaker notion of synchronicity. At any time we only require a $(1 - \mathrm{o}(1))$ fraction of the nodes to be *almost synchronous*. This relaxes full synchronicity in three ways: First, nodes are only "almost synchronous", meaning that for any two nodes their clocks (adjusted over time) may differ by up to $\Delta = \Theta(\log n / \log \log n)$. Secondly, we allow $\mathrm{o}(n)$ nodes to be poorly synchronized. Finally, we require this to hold only with high probability.

The above notion does not require the nodes to synchronize actively per se, since their number of ticks is to some extent concentrated even without active synchronization. However, it turns out that without synchronizing perpetually, the number of poorly synchronized nodes in each phase will become larger than the initial bias (we are interested in) towards the plurality opinion $c_1 - c_2$ and could therefore influence the consensus significantly. We thus synchronize actively (see Algorithm 10) nodes at the end of each phase to decrease the

[5] To allow for an easier comparison with the synchronous model, we will normalize the run time of all sequential algorithms and continuous processes throughout this thesis by dividing their run time by n [AGV15].

fraction of poorly synchronized nodes such that their number is in $o(c_1 - c_2)$, resulting in a negligible influence of those nodes.

Results Consensus via Load Balancing. To the best of our knowledge, no effort has been made to ensure plurality consensus in arbitrary undirected graphs. We develop two protocols Balance and Shuffle which are heavily inspired by distributed load balancing. In the spirit of generalization, our protocols work in a multitude of different environments: Asynchronous, sequential, synchronous, and random matchings. The beauty of our protocols is that they're transparent to the underlying communication environment and use the results from load balancing as a black box.

Protocol Balance (Algorithm 13) uses load balancing in the most natural way possible: Each node creates a polynomial number of balls of its own color and then balances the balls of all colors separately for

$$\boxed{t_{\text{Balance}} = O\left(\frac{\log n}{1-\lambda_2}\right)}$$

rounds, where λ_2 is the second largest eigenvalue. Afterwards, each node simply chooses the color of which it has the most number of balls. This will with high probability be the plurality opinion (Theorem 12.9). While this is very efficient in the settings with few colors (small k), the required memory per node is $\Theta(k \log n)$. Reducing this memory is the goal of our algorithm Shuffle (Algorithm 12), which essentially performs "blind" load-balancing as opposed to color separated load balancing as done by Balance: In the "diffusion setting" where all nodes can communicate to all of the neighbors in every round the algorithm essential just chooses half of its tokens and randomly sends them in equal shares to each of the neighbors. Our results (Theorem 12.9) show a trade-off between memory usage and consensus time of Shuffle. The challenge in analyzing Shuffle is the dependencies between the tokens: The number of tokens per node is invariant and thus if we reveal the positions of some tokens, we affect the probability distribution of the remaining tokens. Fortunately, it turns out that the the dependencies work in the right direction: the tokens are negatively associated (Lemma 12.6) allowing us to apply concentration inequalities.

7.3 Related Work

7.3.1 Voter Model

Due to the duality, one easily obtains the bound $O(t_{\text{meet}} \log n)$ on the coalescence time. This bound appears implicitly in the work of Hassin and Peleg [HP01]. In recent work, Cooper *et al.* [CEOR13] provide results that are better than $O(t_{\text{meet}} \log n)$ for several interesting graph classes, notably expanders and power-law graphs. They show that $t_{\text{coal}} = O((\log^4 n + \|\pi\|_2^{-2}) \cdot (1-\lambda_2)^{-1})$, where λ_2 is the second largest eigenvalue of the transition matrix of the random walk and $\boldsymbol{\pi}$ is the stationary distribution. Berenbrink *et al.* [BGKM16]

show that $t_{\text{coal}} = O(m/(d_{\min} \cdot \Phi))$, where m is the number of edges, $d_{\min}$ is the minimum degree and Φ is the conductance. Their result improves on that of Cooper *et al.* for certain graph classes, *e. g.,* cycles. Their bounds hold in dynamic graphs where edges change, in a restricted way, over time.

Despite the recent progress due to Cooper *et al.* [CEOR13] as well as Berenbrink *et al.* [BGKM16], for many fundamental graphs such as the hypercube and the (d-dimensional) torus, the coalescing time in the discrete setting remains unsettled.

The coalescing random walk process has also been studied in *continuous time*; in this case, particles jump to a random neighboring node when activated according to a Poisson clock with mean 1. As Cooper and Rivera [CR16] recently pointed out

> *"It is however, not clear whether the continuous-time results apply to the discrete-time setting."*

and to the best of our knowledge, there is no general way in which results in *continuous time* can be transferred to *discrete time* or vice versa, even when the random walks in discrete-time are *lazy*. In the continuous time setting, Cox [Cox89] show that the coalescence time is bounded by $\Theta(t_{\text{hit}})$ for tori. Oliveira [Oli12] showed that the coalescence time is $O(t_{\text{hit}})$ in general. In a different work, Oliveira [Oli13] proved so-called mean field conditions, which are sufficient conditions for the coalescing process on a graph to behave similarly to that on a complete graph. His main result in [Oli13], Theorem 1.2, implies that $t_{\text{coal}} = O(t_{\text{meet}})$ whenever $t_{\text{mix}} \cdot \pi_{\max} = O(1/\log^4 n)$.

7.3.2 2-Choices

The expected convergence time of the Voter process is at least $\Omega(n)$ on many graphs, such as regular expanders and complete graphs. Taking into account that solutions to many other fundamental problems in distributed computing, such as information dissemination [KSSV00] or aggregate computation [KDG03], are known to run much more efficiently, Cooper et al. noted that there is room for improvement. To address this issue, Cooper et al. [CER14] introduced the 2-Choices voting process. In this modified process, one is given a graph $G = (V, E)$ where each node has one of two possible opinions. The process runs in discrete rounds during which, unlike in the Voter process, every node is allowed to contact two neighbors chosen uniformly at random. As mentioned earlier, if both neighbors have the same opinion, then this opinion is adopted, otherwise the calling vertex retains its current opinion in this round.

They show that in random d-regular graphs, with high probability all nodes agree after $\mathrm{O}(\log n)$ steps on the initially most frequent opinion, provided that $c_1 - c_2 = K \cdot (n\sqrt{1/d + d/n})$ for K large enough, where c_1 and c_2 denote the support of the initially most frequent and second-most frequent colors. For an arbitrary d-regular graph G, they need $c_1 - c_2 = K \cdot \lambda_2 \cdot n$, where λ_2 is the second largest eigenvalue. In the more recent

work by Cooper et al. [CER+15], the results from [CER14] have been extended to general expander graphs, cutting out the restrictions on the node degrees but nevertheless proving that the convergence time remains in $O(\log n)$. Recently, the authors of [CRRS16] showed the following bound on the consensus time in regular expanders. If the initial bias between the largest and second-largest opinion is at least $c_1 - c_2 \geq Cn \max\{\sqrt{\log n / c_1}, \lambda_2\}$, where λ_2 is the second largest eigenvalue and C is a suitable constant, then the largest opinion wins in $O((n \log n)/c_1)$ steps, with high probability. Very recently Ghaffari and Lengler [GL17] show for 2-Choices that for $k = O(\sqrt{n \log n})$ the consensus time is $O(k \log n)$ which is known to be tight. The authors suggest that their results extend to 3-Majority.

7.3.3 3-Majority

All theoretical results for 3-Majority consider the complete graph. The authors of [BCN+14b] assume that the bias is $\Omega(\min\{\sqrt{2k}, (n/\log n)^{1/6}\} \cdot \sqrt{n \log n})$. Under this assumption, they prove that consensus is reached with high probability in $O(\min\{k, (n/\log n)^{1/3}\} \cdot \log n)$ rounds, and that this is tight if $k \leq (n/\log n)^{1/4}$. The only result without bias [BCN+16] restricts the number of initial colors to $k = o(n^{1/3})$. Under this assumption, they prove that 3-Majority reaches consensus with high probability in $O((k^2(\log n)^{1/2} + k \log n) \cdot (k + \log n))$ rounds. Their analysis considers phases of length $O(k^2 \log n)$ and shows that, at the end of each phase, one of the initial colors disappears with high probability. Note that this approach is so far the only one not assuming any bias cannot yield sublinear bounds with respect to k.

7.3.4 Further Consensus Dynamics

A related consensus process is 2-Median [DGM+11]. Here, every node updates its color (a numerical value) to the median of its own value and two randomly sampled nodes. Without assuming any initial bias, the authors show that this process reaches consensus with high probability in $O(\log k \cdot \log \log n + \log n)$ rounds. This is seemingly stronger than the bounds achieved for 3-Majority and 2-Choices without bias. However, it comes at the price of a complete order on the colors (our processes require colors only to be testable for identity). Moreover, 2-Median is not self-stabilizing for Byzantine agreement (unlike 3-Majority and 2-Choices [BCN+16, EFK+16]): it cannot guarantee *validity*[6] [BCN+16]. Another consensus process is the UndecidedDynamics. Here, each node randomly samples one neighbor and, if the sample has a different color, adopts a special "undecided" color. In subsequent rounds, it tries to find a new (real) color by sampling one random neighbor. The most recent results [BCN+15b] show that, for a large enough bias, consensus is reached with high probability in at most $O(k \log n)$ rounds. Slightly more involved variants yield improved

[6] Byzantine agreement requires that the system does not converge to a color that was initially not supported by at least one non-corrupted node.

bounds of $O(\log k \cdot \log n)$ [BFGK16, GP16, EFK+16]. However, observe that for $k = n$ all nodes become undecided with constant probability instead of agreeing on a color.

Another natural variant is five-sample voting in d-regular graphs with $d \geq 5$, where in each round at least five distinct neighbors are consulted. Abdullah and Draief showed for the case $k = 2$ an $O(\log_d \log_d n)$ bound [AD15], which is tight for a wider class of voting protocols. A more general analysis of multi-sample voting has been conducted by Cruise and Ganesh [CG14] on the complete graph.

7.3.5 Consensus Protocols

There is a diverse body of literature that analyzes consensus problems under various models and assumptions. Results differ in the considered network topology (e.g., arbitrary or complete), the restrictions on model parameters (e.g., the number of opinions or the *initial bias*[7]), the time model (synchronous or asynchronous), or the required knowledge (e.g., n, maximal degree, or spanning tree). To capture this diverse spectrum, we classify[8] results into *population protocols* and *sensor networks*. We will not discuss work whose focus is too far away from this paper's, e.g., consensus on some arbitrary opinion, leader election, robustness concerns, or Byzantine models.

Population Protocols. The first area of work we consider comes from *population protocols*. Population protocols model interactions between large populations of very simple entities (like molecules). Entities are modeled as finite state machines with a small state space and communicate asynchronously. In each step, an edge is chosen uniformly at random and only the two connected nodes communicate. We refer to this communication model as the *sequential model*. See [AR07, AAER07] for detailed introductions.

Angluin et al. [AAE08] propose a 3-state population protocol for majority voting (i.e., $k = 2$) on the clique. If the initial bias α is $\omega(\log n/\sqrt{n})$, their protocol agrees (w.h.p.) [9] on the majority opinion in $O(n \cdot \log n)$ steps. Mertzios et al. [MNRS14] suggest a 4-state protocol for *exact* majority voting, which always returns the majority opinion (independent of α) in time $O(n^6)$ in arbitrary graphs and in time $O(n^2 \cdot \log(n)/\alpha)$ in the clique. This is optimal in that no population protocol for exact majority can have fewer than four states [MNRS14].

Alistarh et al. [AGV15] gives a protocol for $k = 2$ in the clique that allows for a speed-memory trade-off. It solves exact majority and has expected *parallel running time*[10]

[7] The bias is $\alpha \coloneqq (n_1 - n_2)/n$, n_1 and n_2 being the support of the most and second most common opinions.

[8] This classification is neither unique nor injective but merely an attempt to make the overview more accessible.

[9] We say an event happens with high probability (w.h.p.) if its probability is at least $1 - 1/n^c$ for $c \in \mathbb{N}$.

[10] The number of steps divided by n. A typical measure for population protocols, based on the intuition that each node communicates roughly once in n steps.

Table 7.1: Summary of plurality consensus results.

	Arbitrary Graph	Number of Opinions	Required Bias α O-notation	Time O-notation	Model	Space O-notation
SHUF.	✓	arbitrary	arbitrary	$T \cdot t_{\text{mix}}$ $T \cdot \log n/(1-\lambda_2)$ for (d-regular graph)	sync & async	see Theorem 12.1
BALA.	✓	arbitrary	arbitrary	τ $\log n/(1-\lambda_2)$ for (d-regular graph)	sync & async	$k \cdot \log n$
[KLS08]	✓	arbitrary	arbitrary	$D + \frac{F_2}{n_1^2} \cdot \log(k)$	broadcast	–
[MNRS1	✓	2	arbitrary	n^5	async	1
[DV12]	✓	2	arbitrary	$\log n/\delta(G, n_1/n)$	async	1
[CER+1	expand	2	$vol(1) - vol(2) \geq 4\lambda_2^2\lvert E\rvert$	$\log n$	sync	1
[CER14]	random d-reg	2	$\sqrt{1/d + d/n}$	$\log n$	sync	1
[BCN+1	✗	$\leq n$	$\sqrt{\min\left\{k, \sqrt[3]{\frac{n}{\log n}}\right\} \cdot \frac{\log n}{n}}$	$\min\left\{k, \sqrt[3]{\frac{n}{\log n}}\right\} \cdot \log n$	sync	$\log(k)$
[BCN+15b]	✗	$O(\sqrt[3]{\frac{n}{\log n}})$	$\varepsilon \cdot n_2/n$	$md(c) \cdot \log n$	sync	$\log(k)$
[EFK+1	✗	$O(n^\epsilon)$	$\sqrt{\log n/n}$	$k + \log n$	sync	$\log(k)$
[BFGK16]	✗	$o(\sqrt{\frac{n}{\log n}})$	$\gg \sqrt{\log n/n}$	$\log n \cdot \log\log n$	sync	$\log(k)$
[AAG17	✗	2	arbitrary	$O(\log^2(n))$	async	$s = O(\log n)$ states
[AGV15	✗	2	arbitrary	$\frac{\log^2(n)}{s\alpha} + \log^2(n)$	async	$s = O(n)$ states
[AAG17]	✗	2	arbitrary	$O(\log^2 n)$	async	$O(\log n)$ states
[GS17]	✗	2	arbitrary	$O(\log n)$	async	$O(\log^{(2)} n)$ states
[AAE08]	✗	2	$\gg \log n/\sqrt{n}$	$\log n$	async	1

SHUFFLE assumes rough bounds on t_{mix} and n. Bounds on α can reduce the space requirements of our protocols. [KLS08] requires a spanning tree and a common set of quasi-random hash functions. Time in the async model use parallel time. All results, except for [DV12], hold w.p. $1 - o(1)$. [AGV15] also gives an expected time of $o(\log(n)/(s\alpha) + \log(n) \cdot \log(s))$.

$O(\frac{\log n}{s \cdot \alpha} + \log n \cdot \log s)$ and (w.h.p.) $O(\frac{\log^2 n}{s \cdot \alpha} + \log^2 n)$. Here, s is the number of states and must be in the range $s = O(n)$ and $s = \Omega(\log n \cdot \log \log n)$.

In contrast to these results, our protocols consider the case of arbitrary number $k \geq 2$ of opinions. Also, with the notable exception of [MNRS14], the above results are restricted to the complete graph. These restrictions are not surprising, given that these protocols operate on a very constrained state space. Moreover, [GS17] provided a protocol for leader election which uses $O(\log \log n)$ states and reaches consensus in $O(\log^2 n)$ rounds. Very recently [AAG17] showed that for a protocol converging to the majority opinion in $O(n^c)$, $c \leq 1$ time steps regardless of the initial configuration, $\Omega(\log n)$ states are necessary. Moreover, they give a protocol requiring $O\Big(\log^2(n)\Big)$ time steps to converge and $O(\log n)$ states.

Sensor Networks. Another line of work has a background in sensor networks. *Quantized interval consensus* draws its motivation from signal processing. Initially, nodes measure quantized values (signals) and then communicate through a network to agree on the quantized values that enclose the average. This can be used to solve majority consensus ($k = 2$). The communication model is typically sequential.

Bénézit et al. [BTV09] propose a protocol that is equivalent to the 4-state population protocol of [MNRS14] and prove that with probability 1 it converges in finite time, but without bounds on that convergence time.

A more recent result by Draief and Vojnovic [DV12] shows that this protocol (and thus [MNRS14]) needs $O(\frac{\log n}{\delta(Q_S, \alpha)})$ steps in expectation. Here, $\delta(Q_S, \alpha)$ depends on the bias α and on the spectrum of a set of matrices Q_S related to the underlying graph. The authors give concrete bounds for several specific graphs (e.g., in the complete graph the consensus time is of order[11] $O(\log n/\alpha)$). The only related result for $k > 2$ we are aware of is [BTV11] which again proves only convergence in finite time.

Another consensus variant is *mode computation*. For example, Kuhn et al. [KLS08] consider a graph of diameter D where each node has one or several of k distinct elements. The authors use a protocol based on a complex hashing scheme to compute the *mode* (the most frequent element) w.h.p. in time $O(D + F_2/n_1^2 \cdot \log k)$. Here, $F_2 = \sum_i n_i^2$ is the second frequency moment and n_i the frequency of the i-th most common element. $F_2/n_1^2 \in [1, k]$ can be seen as an alternative bias measure. Nodes communicate via synchronous broadcasts and need a precomputed spanning tree and hash functions. [KLS08] can also be used for aggregate computation as done by Kempe et al. [KDG03] (where the authors provide an elegant protocol to compute sums or averages in complete graphs).

Overall, [DV12] and [KLS08] are probably the most closely related to our work since they consider arbitrary graphs. However, we our work consider more general communication models, including dynamic graphs. Similar to [DV12], our results for $k = 2$ rely on spectral properties of the underlying graph (and are asymptotically the same for their concrete

[11] We state their bound in terms of our $\alpha = (n_1 - n_2)/n$; their definition of α differs slightly.

examples). However, our bounds are related to well-studied load balancing bounds and mixing times of random walks (which we believe are easier to get a handle on than their $\delta(Q_S, \alpha)$).

Further Consensus Protocols.

The authors of [FHK15, FN16] give efficient protocols for plurality consensus in the setting where transmission are subject the noise.

Chapter 8

Voter Model [KMS16]

In the *voter model*, each node of a graph has an opinion, and in every round each node chooses independently a random neighbor and adopts its opinion. We are interested in the *consensus time*, which is the first point in time where all nodes have the same opinion. In order to derive strong bound on the voter model we will study the dual of this problem which is called Coalescing random walks. Coalescing random walks is a fundamental stochastic process on *connected* and *undirected* graphs. The process begins with particles on some subset of the nodes in the graph. At discrete time steps, every particle performs one step of a random walk.[1] Whenever two or more particles arrive at the same node at the same time step, they merge into a single particle and continue as a single random walk. The *coalescence time* is defined as the first time step when only one particle remains. The coalescence time depends on the number and starting positions of the particles.

When starting with two particles, the coalescence time is referred to as the *meeting time*. Let t_{meet} denote worst-case expected meeting time over all pairs of starting nodes and let t_{coal} denote the expected coalescence time starting from one particle on every node. It is clear that $t_{\text{meet}} \leq t_{\text{coal}}$; as for an upper bound, it can be shown that $t_{\text{coal}} = O(t_{\text{meet}} \log n)$, where n is the number of nodes in the graph. The main idea used to obtain the bound is that the number of surviving random walks halves roughly every t_{meet} steps. A proof of the result appears implicitly in the work of Hassin and Peleg [HP01].

8.1 Results

In this work, we provide several results relating the coalescence too two fundamental quantities related to random walks: the mixing time and the meeting time. In particular, our focus is on understanding for which graphs the coalescence time is the same as meeting time, as we know that t_{coal} is always in the rather narrow interval of $[\Omega(t_{\text{meet}}), O(t_{\text{meet}} \cdot \log n)]$. As a

[1] Throughout this thesis, we use random walk and particle interchangeably, assuming that every random walk has an identifier.

consequence of our results, we derive new bounds on the coalescence times for several graph families of interest. Formal definitions of all quantities used below appear in Chapter 2 and Section 8.3.1.

Our first main result relates t_{coal} to t_{meet} and t_{mix}. As already mentioned in the introduction, the crude bound $t_{\text{coal}} = O(t_{\text{meet}} \log n)$ is well-known. However, this bound is not in general tight, as demonstrated by our result below.

Theorem 8.1. *For any graph G, we have*

$$t_{\text{coal}} = O\left(t_{\text{meet}}\left(1 + \sqrt{\frac{t_{\text{mix}}}{t_{\text{meet}}}} \cdot \log n\right)\right)$$

Consequently, when $t_{\text{meet}} \geq t_{\text{mix}} \log^2 n$, $t_{\text{coal}} = O(t_{\text{meet}})$.

The proof of Theorem 8.1 appears in Section 8.3. One interesting aspect about this bound is that it can be used to establish $t_{\text{coal}} = \Theta(t_{\text{meet}})$ even without having to know the quantities t_{meet} or t_{mix}. This flexibility turns out to particularly useful when dealing with random graph models for "real world" networks. From this we immediately derive the coalescence time for Erdős Rény graphs, random regular graphs, hypercubes, tours for any $d > 2$ as well as for many "real word" networks:

Common features of real world graph models are (i) a power-law degree distribution with exponent $\beta \in (2, 3)$ and (ii) high expansion, *i. e.*, $1 - \lambda_2$ is not too large, and hence $t_{\text{mix}} = O(\log n)$. Notice that (i) $\beta \in (2, 3)$ implies that w.h.p. we have $\Delta = O(n^{1-\epsilon})$, and hence $\|\boldsymbol{\pi}\|_2^2 \leq \max_{u \in V} \pi(u) \leq n^{-\epsilon}$, for $\varepsilon > 0$. It is easy to see that $t_{\text{coal}} \geq t_{\text{meet}} = \Omega(\|\boldsymbol{\pi}\|_2^{-2})$ which implies for the above defined networks that $t_{\text{coal}} = \Omega(n^{\epsilon})$. This implies that for a large range of parameters we have that $t_{\text{coal}} = \Theta(t_{\text{meet}})$.

Our next main result shows that the bound in Theorem 8.1 is tight up to a constant factor, which we establish by constructing an explicit family of graphs. Interestingly, for this family of almost-regular graphs we also have $t_{\text{hit}} \gg t_{\text{meet}}$, thus showing that t_{hit} may be a rather loose upper bound for t_{coal} in some cases.[2]

Theorem 8.2. *For any sequence $(\alpha_n)_{n \geq 0}$, $\alpha_n \in [1, \log^2 n]$ there exists a family of almost-regular graphs (G_n), with G_n having $\Theta(n)$ nodes and satisfying $\frac{t_{\text{meet}}}{t_{\text{mix}}} = \Theta(\alpha_n)$ such that*

$$t_{\text{coal}} = \Omega\left(t_{\text{meet}} \cdot \left(1 + \sqrt{\frac{t_{\text{mix}}}{t_{\text{meet}}}} \cdot \log n\right)\right).$$

The above two results show that that $t_{\text{meet}}/t_{\text{mix}}$ should be $\Omega(\log^2 n)$ to guarantee that $t_{\text{coal}} = O(t_{\text{meet}})$.

[2]Note that the star also exhibits $t_{\text{hit}} \gg t_{\text{meet}}$. However, the star is not almost-regular.

8.2 Approach and Technical Contributions

When dealing with processes involving concurrent random walks, a significant challenge is often to understand the behavior of "short" random walks. This challenge appears in several settings, *e. g.,* in the context of cover time of multiple random walks [AAK+11, ER09], where Efremenko and Reingold [ER09, Section 6] highlight the difficulty in analyzing the hitting time distribution before its expectation. In the context of concentration inequalities for Markov chains, Lezaud [Lez89, p. 863] points out the requirement to spend at least mixing time steps before taking any samples. Related to that, in property testing, dealing with graphs that are far from expanders has been mentioned as one of the major challenges to test the expansion of the graph by Czumaj and Sohler [CS10].

In our setting, we also face these generic problems and devise different methods to get a handle on the meeting time distribution before its expectation.

Bounds on t_{coal} in terms of t_{mix} and t_{meet}

The key ingredient in the proof of Theorem 8.1, where we express t_{coal} as a trade-off between t_{meet} and t_{mix} is the following: a tight bound on the probability p_ℓ that two random walks meet before ℓ time steps, for ℓ in the range $[t_{\text{mix}}, t_{\text{meet}}]$. Arguing about meeting probabilities of walks that are much shorter than t_{meet} allows us to understand the rate at which the number of *alive*, random walks is decreasing.

Optimistically, one may hope that starting with k random walks, as there are $\binom{k}{2}$ possible meeting events, roughly $\binom{k}{2} \cdot p_\ell$ meetings may have occurred after ℓ time steps. However, the non-independence of these events turns out to be a serious issue and we require a significantly more sophisticated approach to account for the dependencies. We divide the k random walks into disjoint groups $\mathcal{G}_1$ and $\mathcal{G}_2$ (with $|\mathcal{G}_1|$ usually being much smaller than $|\mathcal{G}_2|$) and walks of $\mathcal{G}_1$ can't be eliminated. The domination of the real process by the group-restricted one is established by introducing a formal concept called immortal process at the beginning of Section 8.3.2. In this stochastic process, we can expose the random walks of $\mathcal{G}_1$ first and consider meetings with random walks in $\mathcal{G}_2$ (for an illustration, see Figure 8.2 on page 103). Conditioning on a specific exposed walk in $\mathcal{G}_1$, the events of the different walks in $\mathcal{G}_2$ meeting this exposed walk are indeed independent. In fact, we will also use the symmetric case where the roles of $\mathcal{G}_1$ and $\mathcal{G}_2$ are switched. Thus, the problem then reduces to calculating the likelihood of a random walk in $\mathcal{G}_2$ having a 'good trajectory', *i. e.,* one which many random walks in $\mathcal{G}_1$ would meet with large enough probability.

Surprisingly, it suffices to divide trajectories into only two categories (Lemma 8.5). Although, one may expect that a more fine-grained classification of trajectories would result in better bounds, it turns out not to be the case. In fact, the bound that we derive on the coalescing time in Theorem 8.1 is tight, and this is precisely due to the tightness of Lemma 8.5. The tightness is established by the following construction (cf. Figure 8.1).

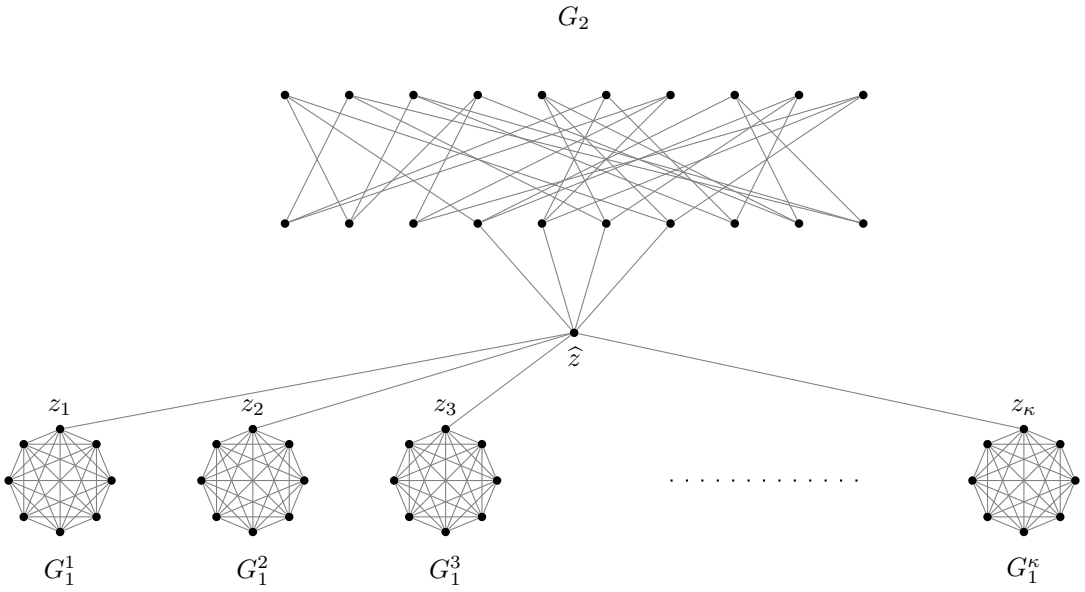


Figure 8.1: The graph described in Section 8.3.5 with $t_{\text{coal}} = \Omega(t_{\text{meet}} + \sqrt{\alpha} \cdot \log n \cdot t_{\text{mix}})$.

The graph is designed such that the vast majority of meetings (between any two random walks) occur in a relatively small part of the graph (G_2 in Figure 8.1). On average, it takes a considerable number of time steps before random walks actually get to this part of the graph. What this implies is that for relatively short trajectories (of length significantly smaller than t_{meet}), it is quite likely that other random walks will not meet them (cf. Lemma 8.5). There is a bit of a dichotomy here, once a walk reaches G_2 it is likely that many random walks will meet it; however, a random walk not reaching G_2 is unlikely to be met by any other random walk.

Equipped with Theorem 8.1, we can bound $t_{\text{coal}} = \Theta(t_{\text{meet}})$ for all graphs satisfying $t_{\text{meet}}/t_{\text{mix}} \geq \log^2 n$. Therefore, the problem of bounding t_{coal} reduces to bounding t_{meet}.

For some of the other results including Theorem 8.2, we will need a more fine-grained approach to derive lower (or upper bounds) on the probability that two walks meet during a certain number of steps, which may or may not be smaller than the mixing time or meeting time. The starting point is the following simple observation. If we have two random walks $(X_t)_{t\geq 0}$ and $(Y_t)_{t\geq 0}$, and count the number of collisions $Z := \sum_{t=0}^{\tau-1} \mathbf{1}_{X_t = Y_t}$ before time step τ,then

$$\mathbb{P}[\, Z \geq 1\,] = \frac{\mathbb{E}[\, Z\,]}{\mathbb{E}[\, Z \mid Z \geq 1\,]}. \tag{8.1}$$

If we further assume that both walks start from the stationary distribution, then we have

$$\mathbb{P}[\, Z \geq 1\,] = \frac{\tau \cdot \|\pi\|_2^2}{\mathbb{E}[\, Z \mid Z \geq 1\,]}.$$

We should mention that this generic approach is of course not new, an analogous variant of counting visits to a vertex has been used by Cooper and Frieze in several works (*e. g.,* [CF05]) to derive very accurate bounds on the hitting (and cover time) on various classes of random graphs, or in Barlow et al. [BPS12] to bound the collisions of random walks on infinite graphs. However, using (8.1), we are able to obtain several improvements to existing bounds on the meeting time, and as a consequence for coalescing time. We believe that our work further highlights the power of this identity.

The crux of (8.1) is that in order to lower (or upper) bound the probability that the two walks meet, we need to derive a corresponding bound on $\mathbb{E}[\, Z \mid Z \geq 1\,]$, *i. e.,* the number of collisions conditioning on the occurrence of at least one collision.

Our results employ various tools to get a handle on this quantity, but here we mention one that is relatively simple:

$$\mathbb{E}[\, Z \mid Z \geq 1\,] \leq \max_{u \in V} \sum_{t=0}^{\tau-1} \sum_{v \in V} \left(p_{u,v}^t\right)^2. \tag{8.2}$$

The inner summand $\sum_{v \in V} (p_{u,v}^t)^2$ is the probability that two walks starting from the same vertex u will meet after further t steps. Thus, summing over t and conditioning on the first meeting happening (*i. e.,* the condition $Z \geq 1$) at some vertex u before time step τ yields the bound in (8.2). Despite the seemingly crude nature of this bound, it can be used to derive new results for $t_{\text{hit}}, t_{\text{meet}}$ and t_{coal} that significantly improve over the state-of-the-art for regular graphs.

8.3 Bounding the coalescence time

In this section we prove Theorem 8.1, one of our main results. We refer the reader to Section 8.2 for a high-level description of the proof ideas.

8.3.1 The coalescence process

We define the coalescence process as a stochastic process as follows: Let $S_0 \subseteq V$ be the set of nodes for which there is initially one random walk on it, and for all $v \in S_t$ let

$$Y_v(t) = \begin{cases} u \in N(v) & \text{w.p. } \frac{1}{2|N(v)|} \\ v & \text{w.p. } \frac{1}{2} \end{cases}$$

The set of *active* nodes in step $t+1$ is given by $S_{t+1} = \{Y_v(t) \mid v \in S_t\}$. The process satisfies the Markov property, *i. e.,*

$$\mathbb{P}[\, S_{t+1} \mid \mathcal{F}_t\,] = \mathbb{P}[\, S_{t+1} \mid S_t\,], \tag{8.3}$$

where $\mathcal{F}_t$ is the filtration up to time t. Finally, we define the *time of coalescence* as $t_{\text{coal}}(S_0) = \min\{t \geq 0 \mid |S_t| = 1\}$. Let $t_{\text{coal}}(S_0)$ be defined as expected consensus time when initially only the nodes of S_0 are occupied by random walks; we use the notation t_{coal} when $S_0 = V$. Throughout this paper, the expression w.h.p. (*with high probability*) means with probability at least $1 - n^{-\Omega(1)}$ and the expression w.c.p. (*with constant probability*) means with probability > 0. We use $\log n$ for the natural logarithm. Appendice A contains some known results about Markov Chains that we frequently use in our proofs.

8.3.2 A more amenable process

In order to prove our first main result, it is helpful to consider a more general stochastic process, P_{imm}, called the immortal process, involving multiple independent random walks. In the immortal process, whenever several random walks arrive at the same node at the same time a subset of them (rather than just one) may survive, while the remaining are merged with one of the surviving walks. To identify the random walks, we assume that each walk has a natural number (in $\mathbb{N}$) as an identifier. In order to define this process formally, we introduce some additional notation and definitions; then we state and prove some auxiliary lemmas. A related concept was introduced in [Oli12, Section 3.4] under the name of "allowed killings".

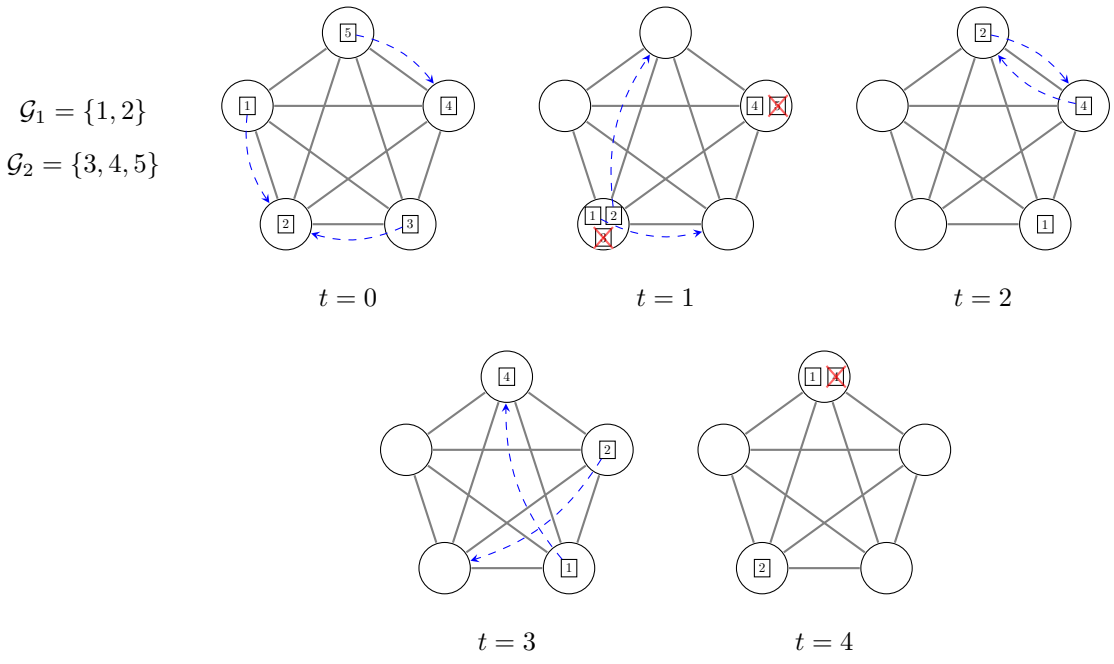


Figure 8.2: Illustration of the process P_{imm}.

As mentioned before, we assume that every random walk r has a unique identifier $\text{id}(r) \in \mathbb{N}$. We divide the ids into two groups $\mathcal{G}_1$, the group of immortal walks and $\mathcal{G}_2$ the group of the remaining (mortal) walks. Whenever two or more walks collide at a node, then

all walks with ids in $\mathcal{G}_1$ survive, while all walks with ids in $\mathcal{G}_2$ are killed (merged with some walk with id in $\mathcal{G}_1$). Furthermore, if all walks have ids in $\mathcal{G}_2$, *i. e.,* there are no walks with id in $\mathcal{G}_1$, then the walk with the minimum id among these walks survives. The ids along with the assignment of ids to groups determine which of the random walks that arrive at a given node at the same time survive.

Formally, let P_{imm} denote the following process:

1. At time 0, $S_0 = \{(u_r, \text{id}(r))\}$, where u_r is the starting node of random walk r and $\text{id}(r)$ is its identifier.

2. At time t, several random walks may arrive at the same node. The process P_{imm} allows some subset of them to survive, while the rest 'coalesce' with one of the surviving walks. Formally, S_{t+1} is defined using S_t as follows. Define the (random) next-step position of the random walk with id $i \in \mathbb{N}$ which is on node $v \in V$ to be

$$Y_{v,i}(t) := \begin{cases} u & \text{where } u \in N(v) \quad \text{w.p. } \frac{1}{2|N(v)|} \\ v & \text{w.p. } \frac{1}{2}, \end{cases}$$

   Let $R_v(t) := \{(Y_{v,i}(t), i) \mid (v,i) \in S_t\}, v \in V$ be the set of next-step positions (before merging happens) for random walks that were at node v at time t. Let

$$\hat{R}_v(t) := \{(v,i) \mid \exists u \in V, (v,i) \in R_u(t)\}$$

   be the random walks that have arrived at node v at time step $t+1$, just before merging happens. Then, merging happens w.r.t. the ids as follows:

   (a) If there exists $i \in \mathcal{G}_1$ such that $(v,i) \in \hat{R}_v(t)$ (at least one walk with id in $\mathcal{G}_1$ arrives at v), then

$$S_v(t+1) := \{(v,j) \mid (v,j) \in \hat{R}_v(t), j \in \mathcal{G}_1\}$$

   (b) If there is no $i \in \mathcal{G}_1$, such that $(v,i) \in \hat{R}_v(t)$ and $\hat{R}_v(t) \neq \emptyset$ (no walk with id in $\mathcal{G}_1$ arrives at v, but at least one walk arrives at v), then

$$S_v(t+1) := \{(v,j)\},$$

   where $j = \min\{i \mid (v,i) \in \hat{R}_v(t)\}$.

   (c) Otherwise, $S_v(t+1) := \emptyset$, *i. e.,* no walk arrived at v.

   Finally, let

$$S_{t+1} := \bigcup_{v \in V} S_v(t+1).$$

We now relate this more general process, P_{imm}, to the coalescing process defined in Section 8.3.1. Let P be regarded as a special instance of P_{imm} with $\mathcal{G}_1 = \{1\}$. In process P, only one of several walks arriving at the same node survives and by convention the one having the smallest id is chosen. Let $(S_t)_{t=0}^{\infty}$ denote the stochastic process P. If we define $\bar{S}_t := \{v \mid (v,i) \in S_t\}$, then $(\bar{S}_t)_{t=0}^{\infty}$ is a coalescence process as defined in Section 8.3.1. Moreover, P represented by $(S_t)_{t=0}^{\infty}$ is the coalescence process which additionally keeps track of the ids. Throughout this chapter, we assume that every random walk of S_0 is on a distinct node.

In the following we show that the time it takes to reduce to k random walks in the original process P is majorized by the time it takes in P_{imm} to reduce to k random walks. While this might be intuitive, one needs to be very careful about the dependencies between the meetings of different random walks: For instance a random walk which is immortal in P_{imm} might eliminate many other random walks whereas the corresponding coupled random walk in P might be eliminated early and therefore cannot eliminate said random walks.

Proposition 8.3. *Consider the following two processes:*

1. *Process P is the standard process of coalescing random walks, viewed as a special case of P_{imm} with $\mathcal{G}_1 = \{1\}$ as described above.*

2. *Process P_{imm} is the process defined above using groups $\mathcal{G}_1$ and $\mathcal{G}_2$, where $1 \in \mathcal{G}_1$.*

Let T^k, T^k_{imm} be the stopping times given by the condition that fewer than k random walks remain for the two processes respectively. Assume both processes start with the same initial configuration, i. e., the vertices occupied by walks in both processes are identical and there is only one walk per vertex in either process. Then, there exists a coupling such that

$$T^k \leq T^k_{\text{imm}}.$$

Proof. We will give a coupling between the moves of walks in P_{imm} and P_{int}, a new process that is essentially intermediate between P and P_{imm}; furthermore, we will show that the original process P is essentially a restricted view of the process P_{int}. The process P_{int} will label the walks *dead*, *alive*, and *phantom*. We emphasize that a phantom walk is not considered alive. Note that the processes P and P_{imm} can be viewed as processes which assign labels to each random walk of the type alive and dead.

Let S_t^{Q} denote the set of tuples of alive walks in process $Q \in \{P, P_{\text{int}}, P_{\text{imm}}\}$ at time t. Let $\bar{S}_t^{\text{Q}} = \{v \mid (v,i) \in S_t^{\text{Q}}\}$ for $Q \in \{P, P_{\text{int}}, P_{\text{imm}}\}$ be the set of nodes which are occupied by at least one alive walk (there might be several in P_{imm} at $t \geq 1$). In order to prove the proposition, we show that there exists a coupling, such that for any $t \in \mathbb{N}$

$$\bar{S}_t^{\text{P}} \subseteq \bar{S}_t^{P_{\text{int}}} \tag{8.4}$$

$$\bar{S}_t^{P_{\text{int}}} \subseteq \bar{S}_t^{P_{\text{imm}}} \tag{8.5}$$

implying that $|\bar{S}_t^{\text{P}}| \leq |\bar{S}_t^{P_{\text{imm}}}|$ which yields the claim since

$$T^k = \min\{t \geq 0 \colon |\bar{S}_t^P| \leq k\} \leq \min\{t \geq 0 \colon |\bar{S}_t^{P_{\text{imm}}}| \leq k\} = T_{\text{imm}}^k.$$

We now define P_{int}. As mentioned above, the walks in P_{int} will be given three kinds of labels alive, dead, or phantom; the dead walks do not continue ahead in time; alive and phantom walks do.

Formally, P_{int} using the groups $\mathcal{G}_1$ and $\mathcal{G}_2$ is defined as follows. We say that walk r is of type $\mathcal{G}_i$, if $\text{id}(r) \in \mathcal{G}_i$ for $i \in \{1, 2\}$. Whenever at least one walk arrives[3] on a node, then the following happens.

1. At least one of the walks is of type $\mathcal{G}_1$
   (a) At least one walk of type $\mathcal{G}_1$ is alive
      i. the walk of $\mathcal{G}_1$ with the smallest id is labeled as alive (even if it was labeled phantom before)
      ii. all other walks of type $\mathcal{G}_1$ (if there are any) are labeled as phantom
      iii. alive walks of type $\mathcal{G}_2$ are labeled dead (if present).
   (b) All walks of type $\mathcal{G}_1$ are phantom walks
      i. There is no walk of type $\mathcal{G}_2$
         A. No label is changed
      ii. There is at least one walk of type $\mathcal{G}_2$
         A. the walk of type $\mathcal{G}_1$ with the smallest id is labeled as alive
         B. all other walks of type $\mathcal{G}_1$ (if there are any) are labeled as phantom
         C. alive walks of type $\mathcal{G}_2$ are labeled dead.
2. All walks are of type $\mathcal{G}_2$
   (a) the walk of $\mathcal{G}_2$ with the smallest id is labeled as alive
   (b) all other walks are labeled as dead.

Note, that walks of $\mathcal{G}_1$ are either alive or phantom and walks of $\mathcal{G}_2$ are either alive or dead. Also, note that in the process P_{int}, there is at most one *alive* walk at any given node. Throughout the proof we regard the processes in two stages: First, each random walk selects a destination (possibly the same node it was on) and moves there. In the second phase the walks are merged according to the process. See Figure 8.3 for an illustration.

[3]Throughout, by arrive we take into account that walks may arrive at a node from the same node through laziness.

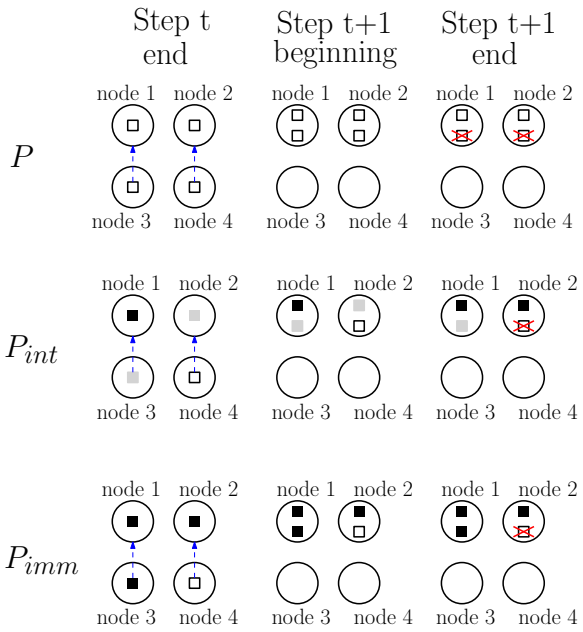


Figure 8.3: An illustration of couplings between the processes. The squares depict the random walks. Walks of $\mathcal{G}_1$ are colored black and gray (phantom) and the nodes of $\mathcal{G}_2$ are white. The blue arrows denote the moving decisions. Observe that in P_{int} a phantom becomes alive (and a walk of $\mathcal{G}_2$ is labeled dead).

We prove (8.4) by induction on t starting from the same initial configuration: if $v \in \bar{S}_t^{\text{P}}$, then $v \in \bar{S}_t^{P_{\text{int}}}$. Consider the inductive step from t to $t+1$ and assume that the claim holds at the end of round t (after merging happened). For the (unique) random walk at $v \in \bar{S}_t^{\text{P}}$ under process P, we couple its transition to node $Y_v(t+1)$ (where we possibly have $Y_v(t+1) = v$) with the corresponding alive walk of $\bar{S}_t^{P_{\text{int}}}$ (there might be several walks of P_{int}, however only one is alive and we couple with this alive walk). Let S be the set of nodes to which a random walk in P moved, *i. e.,* $S = \{Y_v(t+1) \colon v \in \bar{S}_t^{\text{P}}\}$. Observe, that before the merging takes place in round $t+1$ (but moves have been made), there is, by induction hypothesis and the coupling, at least one alive walk of P_{int} on each node of S. Furthermore, the definition of P_{int} ensures that whenever an alive random walk moves to a node, then after merging takes place, at least[4] one alive walk remains. Thus, our coupling ensures that if $v \in \bar{S}_{t+1}^{\text{P}}$, then $v \in \bar{S}_{t+1}^{P_{\text{int}}}$. In words, if one looks at the subsets where there is an alive walk of P_{int}, this is essentially the standard coalescence process. This finishes the proof of (8.4) and we turn to proving (8.5).

When starting from the same initial configuration, we will provide a coupling that satisfies the following invariants.

1. There is a bijective map from the alive and phantom walks of P_{int} to the alive walks of P_{imm}, such that the following holds. All walks of P_{int} of type $\mathcal{G}_i$ are mapped to walks of P_{imm} of type $\mathcal{G}_i$, for $i \in \{1, 2\}$.

[4]By definition, there is actually exactly one alive walk.

2. Whenever a walk of type $\mathcal{G}_2$ is labeled dead in P_{imm}, then it is also labeled dead in P_{int} and vice versa.

At the beginning there are no dead or phantom walks in P_{int}, there are no dead walks in P_{imm}, all walks are alive and as the starting positions in P_{imm} and P_{int} are the same, an arbitrary bijective mapping may be chosen, so long as it respects node positions and walk types.

Assume the invariant holds at time t. We take one random walk step for each alive or phantom random walk in P_{int}. These are coupled with the corresponding walks in P_{imm}, under the chosen map. Walks that are already dead are neither simulated in P_{int} nor in P_{imm}. Hence, we can ensure the bijection between the walks of $\mathcal{G}_1$ in both processes holds at time $t+1$.

We now prove the second invariant. Note that whenever a walk r of type $\mathcal{G}_2$ in $P_{\text{imm}}(P_{\text{int}})$ is labeled dead, this implies there must have been another walk r' on the same node at the same time. Since there is a bijective map, r' must be on the same node in $P_{\text{int}}(P_{\text{imm}})$. We have that either r' is of type $\mathcal{G}_1$ or r' is of type $\mathcal{G}_2$ and that $\text{id}(r') < \text{id}(r)$. In either case, r is also killed (labeled dead) in $P_{\text{int}}(P_{\text{imm}})$. Hence, we can ensure the bijection between the walks of $\mathcal{G}_2$ in both processes holds at time $t+1$. Thus, the invariant holds at time $t+1$. By induction, and since the alive walks of P_{int} are a subset of the alive walks of P_{imm} the invariant holds throughout the process and yielding (8.5). This finishes the proof. □

8.3.3 Meeting Time Distribution Prior to t_{meet}

Let $(X_t)_{t\geq 0}$ and $(Y_t)_{t\geq 0}$ be independent random walks starting at arbitrary positions. For τ a multiple of t_{mix}, the following lemma gives a lower bound on the probability of intersection of the two random walks in τ steps.

Lemma 8.4. *Let $(X_t)_{t\geq 0}$ and $(Y_t)_{t\geq 0}$ be two independent random walks starting at arbitrary positions. Let* $\text{intersect}(X_t, Y_t, \tau)$ *be the event that there exists $0 \leq s \leq \tau$, such that $X_s = Y_s$. Then*

$$\mathbb{P}[\,\text{intersect}(X_t, Y_t, 5t_{\text{mix}})\,] \geq \frac{1}{32\alpha},$$

where $\alpha = t_{\text{meet}}/t_{\text{mix}}$. Furthermore, for any $1 \leq b \leq \frac{e-1}{e} \cdot \alpha$, there exists a constant $c > 0$, such that

$$\mathbb{P}[\,\text{intersect}(X_t, Y_t, c\,b\,t_{\text{mix}})\,] \geq \frac{b}{\alpha},$$

Proof. First, let $(\tilde{X}_t)_{t\geq 0}$ and $(\tilde{Y}_t)_{t\geq 0}$ be two random walks that start from two independent samples drawn from the stationary distribution and are run for $\ell := 2\lceil\alpha\rceil\lceil t_{\text{mix}}\rceil$ steps. Notice that $\ell \geq 2t_{\text{meet}}$, and hence, by Markov's inequality,

$$\mathbb{P}\Big[\,\text{intersect}(\tilde{X}_t, \tilde{Y}_t, \ell)\,\Big] \geq \frac{1}{2}. \tag{8.6}$$

Furthermore, if we divide the interval $[1, \ell]$ into $2\lceil\alpha\rceil$ consecutive sections of length $\lceil t_{\text{mix}}\rceil$ each, the probability for a collision in each of these section is identical and therefore the union bound implies

$$\mathbb{P}\Big[\,\text{intersect}(\tilde{X}_t, \tilde{Y}_t, \ell)\,\Big] \leq 2\lceil\alpha\rceil \cdot \mathbb{P}\Big[\,\text{intersect}(\tilde{X}_t, \tilde{Y}_t, t_{\text{mix}})\,\Big], \tag{8.7}$$

and hence combining equation (8.6) and (8.7) yields

$$\mathbb{P}\Big[\,\text{intersect}(\tilde{X}_t, \tilde{Y}_t, t_{\text{mix}})\,\Big] \geq \frac{1}{4\lceil\alpha\rceil}.$$

Consider now two independent random walks $(X_t)_{t\geq 0}$ and $(Y_t)_{t\geq 0}$ starting at arbitrary positions. By applying Lemma A.27 to both walks, with probability at least $(1-e^{-1})^2$ both $X_{4t_{\text{mix}}}$ and $Y_{4t_{\text{mix}}}$ are drawn independently from the stationary distribution since $4t_{\text{mix}} \geq t_{\text{sep}}$. Therefore,

$$\mathbb{P}[\,\text{intersect}(X_t, Y_t, 5t_{\text{mix}})\,] \geq (1-e^{-1})^2 \cdot \mathbb{P}\Big[\,\text{intersect}(\tilde{X}_t, \tilde{Y}_t, t_{\text{mix}})\,\Big] \geq (1-e^{-1})^2 \cdot \frac{1}{4\lceil\alpha\rceil}.$$

Observing that for any $\alpha \geq 1$, the RHS above expression is greater than $1/(32\alpha)$ completes the proof of the first part. For the second part, we consider k blocks of length $5t_{\text{mix}}$. Due to independence of different blocks, the probability of that the two walks meet in at least one of the k blocks is at least $1-(1-\frac{1}{32\alpha})^k$. We set $k := \lceil * \rceil 32b/(1-e^{-1})$, $x := 1/(32\alpha)$. We distinguish between two cases.

Case $k \cdot x < 1$: We use the fact that $(1-x)^k \leq e^{-xk} \leq 1-(1-e^{-1})xk$ for $0 \leq x < 1$, $k \geq 0$ and $xk \leq 1$. We derive that the probability of intersecting after k blocks is at least $1-(1-\frac{1}{32\alpha})^k \geq (1-e^{-1})k/(32\alpha) = b/\alpha$.

Case $k \cdot x \geq 1$: We have $1-(1-\frac{1}{32\alpha})^k \geq 1-(1-\frac{1}{32\alpha})^{32\alpha} \geq 1-1/e \geq b/\alpha$. In both cases the second part follows. □

At the heart of the proof of Theorem 8.1 lies the following lemma that analyses the marginal distribution of the meeting time distribution. That is, we only expose the first random walk $(X_t)_{t=0}^{\tau}$, and look at how this affects the probability of meeting. In essence, we show that at least one of the two "orthogonal" cases hold. In Case 1 (corresponding to set C_1), there is at least a modest probability that after exposing (X_t), (Y_t) will intersect with significant probability. Otherwise, in Case 2 (corresponding to set C_2), there is a significant probability that after exposing (X_t), (Y_t) will intersect with at least a modest probability.

Lemma 8.5. *Fix $\tau \in \mathbb{N}$ and a graph G. Let $(X_t)_{t=0}^{\tau}$ and $(Y_t)_{t=0}^{\tau}$ be independent random walks, where the starting nodes X_0 and Y_0 are drawn independently from the stationary distribution $\boldsymbol{\pi}$ (w.r.t. to G), and the walks are run for τ steps. Let $p = \mathbb{P}[\,\text{intersect}(X_t, Y_t, \tau)\,]$ and let $\mathcal{T}_\tau$ denote the set all possible trajectories of a walk of length τ in G (including possible*

self-loops). We define the following two categories C_1 and C_2 with $C_1 \subseteq C_2$

$$C_1 := \{(z_0, \dots, z_\tau) \in \mathcal{T}_\tau \colon \mathbb{P}[\,\exists 0 \le s \le \tau, Y_s = z_s\,] \ge \sqrt{p}\}$$
$$C_2 := \{(z_0, \dots, z_\tau) \in \mathcal{T}_\tau \colon \mathbb{P}[\,\exists 0 \le s \le \tau, Y_s = z_s\,] \ge p/3\}.$$

Then, $\mathbb{P}[\,(X_t)_{t=0}^{\tau} \in C_1\,] \ge \frac{p}{3}$ or $\mathbb{P}[\,(X_t)_{t=0}^{\tau} \in C_2\,] \ge \frac{\sqrt{p}}{3}$.

While the actual lower bounds on the probabilities appear rather crude, it turns out that the "significant probability" $\sqrt{p}/3$ is best possible, as we demonstrate in our lower bound construction later. Remarkably, the fact that the "modest probability" is only $p/3$ and much smaller than $\sqrt{p}/3$ does not affect the tightness of our bound, since in Claim 8.7, we can make up for this gap in both cases through a simple amplification argument over the unexposed random walks.

Proof. Let us suppose that $\mathbb{P}[\,(X_t)_{t=0}^{\tau} \in C_1\,] < \frac{p}{3}$. We show that this implies $\mathbb{P}[\,(X_t)_{t=0}^{\tau} \in C_2\,] \ge \frac{\sqrt{p}}{3}$. Assume for the sake of contradiction $\mathbb{P}[\,(X_t)_{t=0}^{\tau} \in C_2\,] < \frac{\sqrt{p}}{3}$. We have

$$\begin{aligned} p &= \mathbb{P}[\,\text{intersect}(X_t, Y_t, \tau)\,] \\ &\le \mathbb{P}[\,(X_t)_{t=0}^{\tau} \in C_1\,] \cdot 1 + \mathbb{P}[\,(X_t)_{t=0}^{\tau} \in (C_2 \setminus C_1)\,] \cdot \sqrt{p} + \mathbb{P}[\,(X_t)_{t=0}^{\tau} \notin C_2\,] \cdot \frac{p}{3} \\ &< p/3 + \sqrt{p}/3 \cdot \sqrt{p} + p/3 \le p, \end{aligned}$$

a contradiction. This completes the proof. □

It is well-known that starting with k random walks, the coalescence time is bounded by $O(t_{\text{meet}} \log k)$, this can be deduced from the proof presented in [HP01]. For the sake of completeness, we give a self-contained proof[5].

Proposition 8.6. *We have $t_{\text{coal}}(S_0) = O(t_{\text{meet}} \log |S_0|)$.*

Proof. Let P be the coalescing process (with ids) defined in Section 8.3.2. Recall that $\mathcal{G}_1 = \{1\}$. Let S_t be set of coalescing random walks at an arbitrary time step t. In the following we show the slightly stronger claim that the expected time to reduce the number of random walks by a constant factor is $O(t_{\text{meet}})$.

Formally, we fix an arbitrary time step t_0. With $T := \min\{t \ge t_0 \colon |S_t| \le 99/100 \cdot |S_{t_0}|, |S_{t_0}| \ge 100\}$ denoting the first time step the number of coalescing random walks reduces by a factor of 99/100, we will prove that $\mathbb{E}[\,T\,] = O(t_{\text{meet}})$. Iterating the argument $O(\log |S_0|)$ times implies that the expected time it takes to reduce to 100 random walks

[5] One might be tempted to pair random walks in groups of two and run them for $2t_{\text{meet}}$ time steps so that, by Markov inequality, they meet with probability at least 1/2. Repeating this iteratively would yield the claim. To formalize such an argument one would need to disallow coalescence between different pairs of random walk which differs from the stochastic process we reduce to in Section 8.3.2.

is $O(t_{\text{meet}} \log |S_0|)$. Note that the expected time to reduce from 100 random walks to 1 is bounded by $O(t_{\text{meet}})$. Hence, the claim $t_{\text{coal}}(S_0) = O(t_{\text{meet}} \log |S_0|)$ follows.

It remains to show that the expected number of time steps it takes to reduce the number of random walks by a factor of 99/100 is indeed $O(t_{\text{meet}})$.

We divide time into blocks of length $\tau := c\frac{e-1}{e} t_{\text{meet}} + 4t_{\text{mix}}$, where c is the constant of Lemma 8.4, *i. e.*, $\mathbb{P}\Big[\,\text{intersect}(X_t, Y_t, c\frac{e-1}{e} t_{\text{meet}})\,\Big] \geq \frac{e-1}{e}$. We are primarily interested in what happens at the end of the blocks, *i. e.*, at time steps $t_0, t_0 + \tau, t_0 + 2\tau, \ldots$. For simplicity, we will start counting time from 0 at the beginning of each block. Let $(X_t)_{t \geq 0}$ be the random walk with id 1. After $4t_{\text{mix}}$ steps, we can couple the state of the random walk $(X_t)_{t \geq 4t_{\text{mix}}}$ with a node drawn from $\boldsymbol{\pi}$ with probability at least $(1 - e^{-1})$, since $4t_{\text{mix}} \geq t_{\text{sep}}$ (see Lemma A.27). Further, note that conditioned on this coupling, the statement of Lemma 8.5 implies that $(X_t)_{t \geq 4t_{\text{mix}}} \in C_2$ w.p. at least $p/3$, where we used $C_2 \subseteq C_1$, and where $p := \mathbb{P}\Big[\,\text{intersect}(\tilde{X}_t, \tilde{Y}_t, c \cdot \frac{e-1}{e} \cdot t_{\text{meet}})\,\Big] \geq \frac{e-1}{e}$ for $\tilde{X}_0, \tilde{Y}_0 \sim \boldsymbol{\pi}$.

We condition on the successful coupling of $X_{4t_{\text{mix}}}$ with a node drawn from $\boldsymbol{\pi}$ and that $(X_t)_{t \geq 4t_{\text{mix}}} \in C_2$, which happens with probability at least $(1 - e^{-1})p/3 = \frac{(e-1)^2}{3e^2}$ (called event $\mathcal{E}$). Finally, consider any random walk $(Y_t)_{t \geq 0} b$ with id other than 1. Again with probability at least $1 - e^{-1}$ we can couple $Y_{4t_{\text{mix}}}$ with a node drawn from $\boldsymbol{\pi}$ and conditioned on successful coupling, $(Y_t)_{t \geq 4t_{\text{mix}}}$ meets $(X_t)_{t \geq 4t_{\text{mix}}}$ between time steps $[4t_{\text{mix}}, \tau]$ with probability at least $p/3$, by definition of C_2. Thus, conditioned on event $\mathcal{E}$, each walk of $\mathcal{G}_2$ vanishes w.p. $(1 - e^{-1})p/3 = \frac{(e-1)^2}{3e^2}$ and thus the expected fraction of walks killed in the τ time steps is at least $\frac{(e-1)^2}{3e}$.

Let $Z_\ell = |S_{t_0 + \ell \cdot \tau}|$ denote the number of random walks alive at the beginning of block ℓ.

$$\mathbb{E}\Big[\, Z_\ell \mid \mathcal{F}_{t_0 + (\ell-1)\cdot\tau} \,\Big] \leq Z_{\ell-1} - (Z_{\ell-1} - 1) \cdot \frac{(e-1)^4}{9e^4} \leq Z_{\ell-1} - \frac{Z_{\ell-1}}{100}.$$

The above holds as long as $Z_{\ell-1} \geq 100$. We can therefore apply Lemma A.14 with parameters $g = 99/100 \cdot S_0$ and $\beta = 99/100$ to obtain that $\mathbb{E}[\,T\,] = O(\tau) = O(t_{\text{meet}})$, which completes the proof. □

8.3.4 Upper Bound - Proof of Theorem 8.1

We commence by considering the process P_{imm} defined in Section 8.3.2. This allows us to establish Claim 8.7 providing us with the following trade-off. For a given period τ of length at least t_{mix} we obtain a bound on the required number of periods to reduce the number of random walks by an arbitrary factor. The proof relies heavily on Lemma 8.5 which divides the walks of $\mathcal{G}_1$ into two groups allowing us to expose the walks of $\mathcal{G}_1$ first and then to calculate the probability of the walks of $\mathcal{G}_2$ to intersect with them. In fact, we will also use the symmetric case where the roles of $\mathcal{G}_1$ and $\mathcal{G}_2$ are switched. These probabilities are derived from the time-probability trade-off presented in Lemma 8.4. We then use Claim 8.7 to derive a bound on the number of time steps it takes to reduce the number of walks to

$\lceil 2\alpha \rceil$, where $\alpha = t_{\text{meet}}/t_{\text{mix}}$ (Corollary 8.8). From there on we employ Claim 8.7 to reduce the number of walks to 1 in $O(t_{\text{meet}})$ time steps. Melding both phases together yields the bound of Theorem 8.1.

We now define a process $P_{\text{imm}}(S_0, k)$ with $k < |S_0|$, which is a parameterized version of the process P_{imm} defined in Section 8.3.2:

- Let $|S_0| = k'$; there are k' random walks with ids $1, \ldots, k'$ and starting nodes $v_1, \ldots, v_{k'}$. Thus, $S_0 = \{(v_i, i) \mid 1 \leq i \leq k'\}$.
- Let $\mathcal{G}_1 = \{1, \ldots, k\}$ and $\mathcal{G}_2 = \{k+1, \ldots, k'\}$. Recall that, by definition of P_{imm}, we have that if some random walks with ids in $\mathcal{G}_1$ and some with ids in $\mathcal{G}_2$ are present on the same node at the same time, only the ones with ids in $\mathcal{G}_1$ survive. If all the random walks have ids in only in $\mathcal{G}_1$, then all of them survive. If all random walks have ids only in $\mathcal{G}_2$, then only the one with the smallest id survives.

We define

$$\text{IDs}(S_t) := \{\text{id}(r) \mid (u_r, \text{id}(r)) \in S_t\}, t \in \mathbb{N}.$$

The following lemma gives the expected time it takes to reduce the number of random walks in $\mathcal{G}_2$ from $k' - k$ to some arbitrary integer $g \geq k$: given a period of length τ and integer g, assuming that $k = |\mathcal{G}_1|$ is large enough, we derive a bound on the number of periods of length τ until the walks in $\mathcal{G}_2$ are reduced to g. The required size of k is a function of the probability for two random walks drawn from $\boldsymbol{\pi}$ intersecting after τ time steps.

Claim 8.7. *Let $\tau \in \mathbb{N}$, let $(X_t)_{t=0}^{\tau}$ and $(Y_t)_{t=0}^{\tau}$ be independent random walks run for τ steps, with X_0 and Y_0 drawn independently from $\boldsymbol{\pi}$. Let $p_\tau \leq \mathbb{P}[\,\text{intersect}(X_t, Y_t, \tau)\,]$ be a lower bound on the probability of the intersection of the two walks during the τ steps. Consider an instantiation of $P_{\text{imm}}(S_0, k)$. Suppose that $k \geq \frac{3}{(1-e^{-1})\cdot p_\tau}$. For some $1 \leq g \leq |S_0| - k$, define the stopping condition $T_g = \min\{t \geq 0 \mid |\,\text{IDs}(S_t) \cap \mathcal{G}_2| \leq g\}$. Then the expected stopping time satisfies*

$$\mathbb{E}[\,T_g\,] = O\left((4t_{\text{mix}} + \tau) \cdot \sqrt{\frac{1}{p_\tau}} \cdot (\log|\mathcal{G}_2| - \log g)\right).$$

We first describe the high-level proof idea, before delving into the formal proof. We divide time into blocks of size $4t_{\text{mix}} + \tau$. For any random walk $(Z_t)_{t=0}^{4t_{\text{mix}}+\tau}$ we can couple its position after $4t_{\text{mix}} \geq t_{\text{sep}}$ w.c.p. with a node drawn from $\boldsymbol{\pi}$. Thus, conditioning on the success of this coupling we have, by Lemma 8.5, $\mathbb{P}\Big[\,(Z_t)_{t=4t_{\text{mix}}}^{4t_{\text{mix}}+\tau} \in C_1\,\Big] \geq \frac{p_\tau}{3}$ or $\mathbb{P}\Big[\,(Z_t)_{t=4t_{\text{mix}}}^{4t_{\text{mix}}+\tau} \in C_2\,\Big] \geq \frac{\sqrt{p_\tau}}{3}$. In the former case we have that w.c.p. there is at least one random walk r in $\mathcal{G}_1$ which is, due to independence of the walks, in class C_1. The hypothetical extension of the trajectory of any random walk in $r' \in \mathcal{G}_2$ intersects with r w.p. $c\sqrt{p_\tau}/3$, where the constant arises due to the fact that we also need to couple the state

of r' at time $4t_{\text{mix}}$ to a node drawn according to $\boldsymbol{\pi}$. (We need to consider the hypothetical extension because the walk r' may get eliminated sooner–this only helps us.) Thus, r' gets eliminated w.p. at least $c\sqrt{p_\tau}$ for a suitable constant c.

In the latter case we have that w.p. at least $c\sqrt{p_\tau}/3$ a random walks of $\mathcal{G}_2$ is in class C_2. Every random walk in that class intersects w.c.p. with at least one of the walks of $\mathcal{G}_1$. Thus, in both cases, we have that in each block a random walk of $\mathcal{G}_2$ is eliminated w.p. a least $c\sqrt{p_\tau}$ for some constant c. Thus, the number of random walks in $\mathcal{G}_2$ decrease in expectation by a factor of $c\sqrt{p_\tau}$.

Proof. We will consider the process in *blocks* each consisting of $4t_{\text{mix}}+\tau$ time steps. For convenience in the proof, we'll restart counting time steps from 0 at the beginning of each block; we keep track of the total number of time steps by counting the number of blocks. Let C_1 and C_2 be as defined in Lemma 8.5. Then we perform a case analysis by considering the two possible outcomes described in Lemma 8.5 separately. We define $Z_j = |\operatorname{IDs}(S_{j\cdot(4t_{\text{mix}}+\tau)}) \cap \mathcal{G}_2|$, *i. e.,* the number of walks remaining in $\mathcal{G}_2$ after j blocks of time have passed. For any $j \geq 1$, we will show that there exists a constant $c > 0$ such that,

$$\mathbb{E}[\, Z_j \mid \mathcal{F}_{j-1}\,] \leq Z_{j-1} \cdot (1 - c\sqrt{p_\tau}).$$

By using Lemma A.14, we get $\mathbb{E}[\,T_g\,] = O\Big((4t_{\text{mix}}+\tau)\cdot\frac{1}{\sqrt{p_\tau}}\cdot(\log|\mathcal{G}_2| - \log g)\Big)$ (the factor $(4t_{\text{mix}}+\tau)$ appears as the size of the block). Recall that $\mathcal{F}_j$ is the filtration up to end of the jth block. In the remainder we show that we have indeed $\mathbb{E}[\, Z_j \mid \mathcal{F}_{j-1}\,] \leq Z_{j-1}\cdot(1 - c\sqrt{p_\tau})$.

Case 1. $\mathbb{P}[\,(X_t)_{t=0}^{\tau} \in C_1\,] \geq \frac{p_\tau}{3}$:

Consider any random walk r in $\mathcal{G}_1$ at the beginning of a *block*. Using Lemma A.27, after $4t_{\text{mix}}$ steps we can couple the state of the random walk with a node drawn from $\boldsymbol{\pi}$ with probability at least $(1 - e^{-1})$. Furthermore, conditioned on this coupling, the portion of the random walk between time steps $4t_{\text{mix}}$ and $4t_{\text{mix}} + \tau$ of the walk is in class C_1 with probability at least $\frac{p_\tau}{3}$. Since $k \geq \frac{3}{p_\tau\cdot(1-e^{-1})}$, w.p. $c_1 > 0$, in any block, there exists a walk in $\mathcal{G}_1$ that has the portion between time steps $4t_{\text{mix}}$ and $4t_{\text{mix}} + \tau$ in C_1.

Fix a block and condition on the event that there is a walk in $\mathcal{G}_1$, denoted by r_1, whose portion between time steps $4t_{\text{mix}}$ and $4t_{\text{mix}} + \tau$ is in C_1. Consider any walk in $\mathcal{G}_2$, denoted by r_2, at the beginning of the block. We want to argue that this walk r_2 has a reasonable probability of intersecting some walk in $\mathcal{G}_1$ in this block of time steps. First, consider (the possibly hypothetical continuation of r_2) walk r_2' for the entire length of the block. The reason for this is that if r_2 and some walk from $\mathcal{G}_1$ are at the same node at the same time sometime in the block, r_2 will be eliminated in the process $P_{\text{imm}}(S_0, k)$; however, we can consider its hypothetical extension to the entire length of the block. Using Lemma A.27 the state of the walk r_2' at time step $4t_{\text{mix}}$ can be coupled with a node drawn from $\boldsymbol{\pi}$ with

probability at least $c_2 := 1 - e^{-1}$. Then conditioned on successful coupling, the probability that r_2' and r_1 collide during time steps $4t_{\text{mix}}$ and $4t_{\text{mix}} + \tau$ is at least $\sqrt{p_\tau}$ (by definition of C_1 in Lemma 8.5). Thus, the probability that r_2 hits at least one walk in $\mathcal{G}_1$ is at least $c_1 \cdot c_2 \cdot \sqrt{p_\tau}$. Note that it is also possible for r_2' to be eliminated by another walk from $\mathcal{G}_2$. In any case, we have that r_2 is eliminated w.p. at least $c\sqrt{p_\tau}$ and we get

$$\mathbb{E}[\, Z_j \mid \mathcal{F}_{j-1}\,] \leq Z_{j-1} \cdot (1 - c_1 \cdot c_2\sqrt{p_\tau}).$$

Case 2. $\mathbb{P}[\,(X_t)_{t=0}^{\tau} \in C_2\,] \geq \frac{\sqrt{p_\tau}}{3}$:
Consider a walk in $\mathcal{G}_2$, denoted by r_2, at the beginning of a block; as in the previous case, we will consider a possibly hypothetical continuation r_2' of r_2. Using Lemma A.27 we can couple the state of r_2' at time step $4t_{\text{mix}}$ with a node drawn from $\boldsymbol{\pi}$ with probability at least $1 - e^{-1}$. Furthermore, conditioned on the successful coupling, with probability at least $\frac{\sqrt{p_\tau}}{3}$ the trajectory of r_2' between the time steps $4t_{\text{mix}}$ to $4t_{\text{mix}} + \tau$ is in C_2. Thus, with probability at least $p := (1 - e^{-1})\frac{\sqrt{p_\tau}}{3}$, r_2' has a trajectory between time steps $4t_{\text{mix}}$ and $4t_{\text{mix}} + \tau$ that lies in C_2. Now consider any random walk $r_1 \in \mathcal{G}_1$ at the beginning of the block. Again, using Lemma A.27 with probability at least $1 - e^{-1}$, we can couple the state of the random walk at time $4t_{\text{mix}}$ with a node drawn from $\boldsymbol{\pi}$. Conditioned on this between time steps $4t_{\text{mix}}$ to $4t_{\text{mix}} + \tau$, this random walk hits any trajectory whose portion between time steps $4t_{\text{mix}}$ to $4t_{\text{mix}} + \tau$ lies in C_2 with probability at least $p_\tau/3$ (by definition of C_2 in Lemma 8.5). Since $k = |\mathcal{G}_1| \geq \frac{3}{(1-e^{-1})\cdot p_\tau}$, with at least constant probability $c_1 > 0$ there is some walk in $\mathcal{G}_1$ that intersects any fixed trajectory whose portion between time steps $4t_{\text{mix}}$ to $4t_{\text{mix}} + \tau$ lies in C_2. Since the random walks in $\mathcal{G}_1$ are independent, by the definition of the immortal process, we have that any walk in $\mathcal{G}_2$ is eliminated by the end of the block with probability at least $c_1 \cdot p = c\sqrt{p_\tau}$ for some constant $c > 0$. Similarly as before, it is possible that r_2 is eliminated by at least one of the walks of $\mathcal{G}_2$, which only increases the probability for r_2 of being eliminated. We get

$$\mathbb{E}[\, Z_j \mid \mathcal{F}_{j-1}\,] \leq Z_{j-1} \cdot (1 - c\sqrt{p_\tau}).$$

□

In the following we bound the time T required to reduce to $2\lceil\alpha\rceil$ random walks. The claim follows by applying Claim 8.7 to derive a bound on T_{imm} for processP_{imm}, and using the majorization of T by T_{imm} (Proposition 8.3).

Corollary 8.8. *Consider the coalescence process starting with set S_0 and let $\alpha = t_{\text{meet}}/t_{\text{mix}}$. Let $T_1 = \min\{t \geq 0 \mid |S_t| \leq 2\lceil\alpha\rceil\}$. Then $\mathbb{E}[\,T_1\,] = O(t_{\text{mix}} \cdot \sqrt{\alpha} \cdot \log|S_0|)$.*

Proof. We consider the process P (defined in Section 8.3.2), which is identical to the coalescence process, but in addition also keeps track of ids of random walks and that al-

lows only the walk with the smallest id to survive. We assume that the ids are from the set $\{1, 2, \ldots, |S_0|\}$. Let $S_0 = \{(v_1, 1), \ldots, (v_{|S_0|}, |S_0|)\}$ and $\bar{S}_0 = \{i \colon (v, i) \in S_0\}$. We consider the process $P_{\text{imm}}(S_0, k)$ and $k = \lceil \alpha \rceil$. Let T_1^* be the stopping time defined by $|\operatorname{IDs}(\bar{S}_t) \cap \mathcal{G}_2| \leq \alpha$ for the process $P_{\text{imm}}(S_0, k)$. By definition of P_{imm} and Proposition 8.3, it follows that T_{imm} stochastically dominates T. Thus, it suffices to bound $\mathbb{E}[\, T_{\text{imm}}\,]$. W.l.o.g. we assume that $\alpha \geq 6\frac{e-1}{e}$, otherwise the claim follows directly from Proposition 8.6. We apply Lemma 8.4 with $b = 6$ and derive that for some suitable constant c,

$$p = \mathbb{P}[\,\text{intersect}(X_{t\geq 0}, Y_{t\geq 0}, 6ct_{\text{mix}})\,] \geq \frac{6}{\alpha},$$

Thus, we have

$$\frac{3}{(1-e^{-1})\cdot p} \leq \frac{3}{\frac{1}{2}\cdot p} \leq \alpha \leq k$$

Applying Claim 8.7 with $g = \alpha$, $\tau = 6ct_{\text{mix}}$ (where c is a constant as given by Lemma 8.4), $p_\tau = 6/\alpha$, and observing that $k \geq \frac{3}{(1-e^{-1})\cdot p_\tau}$, we get the required result. □

In the following we bound the time T required to reduce from $2\lceil\alpha\rceil$ random walks to a single random walk. The proof uses the same ideas as before (Corollary 8.8) however, this time we consider several phases and in each we reduce the number of random walks by a constant factor. The expected time per phase is geometrically increasing as the number of walks decreases and the overall time is essentially dominated by the time for a constant number of random walks to meet, which is $O(t_{\text{meet}})$.

Lemma 8.9. *Consider the coalescence process starting with set S_0, satisfying $|S_0| \leq 4\alpha \log \alpha$, where $\alpha = t_{\text{meet}}/t_{\text{mix}}$. Let $T_2 := \min\{t \geq 0 \mid |S_t| \leq 1\}$. Then $\mathbb{E}[\, T_2\,] = O(t_{\text{meet}})$.*

Proof. We will consider the coalescence process in phases. Let ℓ be the largest integer such that $|S_0| \geq \left(\frac{4}{3}\right)^\ell$. For $j \geq 1$, the jþ phase ends when $|S_t| < \left(\frac{4}{3}\right)^{\ell-j+1}$. The $(j+1)$þ phase begins as soon as the jþ phase ends. Note that it may be the case that some phases are empty. Let $T_2(j)$ denote the time for phase j to last. We will only consider phases up to which $\ell - j + 1 \geq 32$.

Now we focus on a particular phase j. Let t_j be the time when the jþ phase begins and let S_{t_j} denote the corresponding set at that time. Thus, we have

$$\left(\frac{4}{3}\right)^{\ell-j+1} \leq |S_{t_j}| < \left(\frac{4}{3}\right)^{\ell-j+2} \tag{8.8}$$

We consider the process P_{imm} defined in Section 8.3.4 as follows. Define $n_j = |S_{t_j}|$. Fix a phase j and define $S'_0 = \{(v_1, 1), \ldots, (v_{n_j}, n_j)\}$ and $\bar{S}'_0 = \{v_1, \ldots, v_{n_j}\}$. Then, consider again the set of occupied vertices (ignoring the labels) $\bar{S}_{t_j+t} = \{v \mid \exists i \in \mathbb{N}, (v, i) \in S'_t\}$ with

$t \in \mathbb{N}$. Thus, phase j ends when $|S'_t| = |\bar{S}_{t_j+t}| < \left(\frac{4}{3}\right)^{\ell-j+1}$. Let

$$k_j := \left\lceil \frac{|S'_0|}{2} \right\rceil$$

be the size of $\mathcal{G}_1$ and consider the process $P_{\text{imm}}(S'_0, k_j)$ as defined in Section 8.3.4. Let

$$g_j := \left\lfloor \frac{|S'_0| - k_j}{3} \right\rfloor$$

and

$$T_2^*(j) := \min\{t \mid |\text{IDs}(S'_t) \cap \mathcal{G}_2| \leq g_j\}.$$

We note that as long as $\ell - j + 1 \geq 32$, $g_j \geq 1$ and at time $T_2^*(j)$,

$$|S'_t| \leq g_j + k_j \leq \frac{|S'_0| - k_j}{3} + k_j = \frac{|S'_0|}{3} + \frac{2k_j}{3} \leq \frac{|S'_0|}{3} + \frac{|S'_0|}{3} + \frac{2}{3} < \frac{3}{4} \cdot |S'_0|.$$

By Proposition 8.3, $T_2^*(j)$ stochastically dominates $T_2(j)$ and hence it suffices to bound $\mathbb{E}[\,T_2^*(j)\,]$. In order to bound $\mathbb{E}[\,T_2^*(j)\,]$, we define

$$b_j := 32\alpha \log(4/3)(\ell - j + 1)(3/4)^{\ell-j+1}.$$

Since we only consider phases with j respecting $\ell - j + 1 \geq 32$ we have $b_j \leq b_{\ell-31} \leq ((e-1)/e)\alpha$. Furthermore, we have $b_j \geq b_0 \geq 4\alpha \log \alpha (3/4)^\ell \geq 1$, where the last inequality follows from $(4/3)^\ell \leq |S_0| \leq 4\alpha \log \alpha$, which in turn follows from definition of ℓ and the assumed bound on $|S_0|$. Applying Lemma 8.4 with this value of b_j, we get that for

$$\tau_j := c b_j t_{\text{mix}},$$

for independent random walks $(X_t)_{t=0}^{\tau_j}$, $(Y_t)_{t=0}^{\tau_j}$, $\mathbb{P}[\,\text{intersect}(X_t, Y_t, \tau_j)\,] \geq p_j$, where

$$p_j := 32 \log(4/3)(\ell - j + 1)(3/4)^{\ell-j+1}.$$

We seek to apply Claim 8.7 to bound $\mathbb{E}[\,T_2^*(j)\,]$. We first verify that the conditions of Claim 8.7 are fulfilled. In particular, we verify that $k_j \geq \frac{8}{p_j}$; to see this consider the following:

$$\frac{8}{p_j} = \frac{8}{32 \log(4/3)(\ell - j + 1)} (4/3)^{\ell-j+1} \leq \frac{1}{4} \cdot \left(\frac{4}{3}\right)^{\ell-j+1} \leq \frac{1}{2} \cdot |S'_0| \leq k_j,$$

where we used (8.8) and $|S'_0| = |S_{t_j}|$ in the second-last inequality. Thus we can apply Claim 8.7 and derive

$$\mathbb{E}[\,T_2^*(j)\,] \le (\tau_j + 4t_{\text{mix}}) \cdot \frac{1}{\sqrt{p_j}} \cdot (\log|\,\text{IDs}(S'_0) \cap \mathcal{G}_2| - \log g_j)$$

and we continue by dissecting that bound. Since $b_j \ge 1$, there exists a suitably large constant c_1, so that $\tau_j + 4t_{\text{mix}} \le c_1 b_j t_{\text{mix}}$. Furthermore,

$$\frac{b_j}{\sqrt{p_j}} = \frac{32\alpha\log(4/3)(\ell-j+1)(3/4)^{\ell-j+1}}{\sqrt{32\log(4/3)(\ell-j+1)(3/4)^{\ell-j+1}}} = O\left(\alpha\sqrt{\ell-j+1}\cdot\left(\frac{3}{4}\right)^{(\ell-j+1)/2}\right).$$

Observe that, by definition, $|\,\text{IDs}(S'_0)\cap\mathcal{G}_2|/g_j \le 3$, hence $\log|\,\text{IDs}(S'_0)\cap\mathcal{G}_2| - \log g_j \le \log(3)$. Putting everything together, we get that there is a constant c_2 such that,

$$\mathbb{E}[\,T_2^*(j)\,] \le c_2 \cdot t_{\text{mix}} \cdot \alpha \cdot \sqrt{\ell-j+1}\left(\frac{3}{4}\right)^{(\ell-j+1)/2} \tag{8.9}$$

Note that since we stop when $\ell - j + 1 < 32$, there are at most $\ell - 30$ phases considered. Let $\tilde{T}$ be the random variable denoting the time step when the last phase ends; at this point $|S_{\tilde{T}}| = O(1)$. Therefore, using Proposition 8.6, $\mathbb{E}\Big[\,T_2 - \tilde{T}\,|\,\tilde{T}\,\Big] = O(t_{\text{meet}})$. But, clearly $\tilde{T}$ is stochastically dominated by $\sum_{j=0}^{\ell-30} T_2^*(j)$. Thus, we have

$$\begin{aligned}
\mathbb{E}[\,T_2\,] &= \mathbb{E}\Big[\,\tilde{T}\,\Big] + \mathbb{E}\Big[\,\mathbb{E}\Big[\,T_2 - \tilde{T}\;\big|\;\tilde{T}\,\Big]\,\Big] \\
&\le c_2 \cdot t_{\text{mix}} \cdot \alpha \sum_{j=0}^{\ell-30} \sqrt{\ell-j+1}\left(\frac{3}{4}\right)^{(\ell-j+1)/2} + c_3 t_{\text{meet}} &(8.10)\\
&\le c_2 \cdot t_{\text{mix}} \cdot \alpha + c_3 t_{\text{meet}} = O(t_{\text{meet}}) &(8.11)
\end{aligned}$$

Above, in (8.10) we used (8.9) and the fact that $\mathbb{E}\Big[\,T_2 - \tilde{T}\;|\;\tilde{T}\,\Big] \le c_3 t_{\text{meet}}$ for some constant $c_3 > 0$ and in step (8.11), we used the fact that $\sum_{j=32}^{\infty} j c^j < 1$ for $c \le \sqrt{3/4}$. □

Thus, the first phase (Corollary 8.8) and the second phase (Lemma 8.9) take together $O(\sqrt{\alpha}\cdot\log n \cdot t_{\text{mix}} + t_{\text{meet}})$ time steps, which yields Theorem 8.1.

8.3.5 Lower Bound - Proof of Theorem 8.2

In this section we give a construction of a graph family in order to establish lower bounds on $t_{\text{coal}}(G)$ in terms of $t_{\text{meet}}(G)$ and $t_{\text{mix}}(G)$ demonstrating that Theorem 8.1 is asymptotically tight. Additionally, our construction generalizes a claim of Aldous and Fill [AF02, Chapter 14]: They mention that it is possible construct regular graphs that mimic the n-star in the sense that the $t_{\text{meet}} = o(t_{\text{avg-hit}})$, without giving further details of the construction. Our construction shows that even the coalescence time can be significantly smaller than the

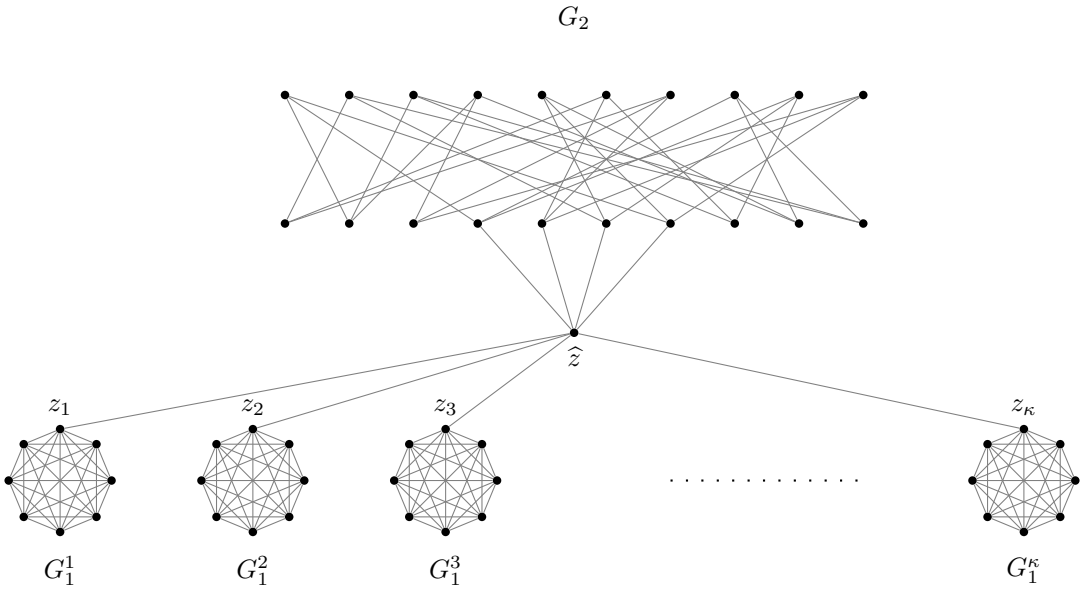


Figure 8.4: The graph described in Section 8.3.5 with $t_{\text{coal}} = \Omega(t_{\text{meet}} + \sqrt{\alpha} \cdot \log n \cdot t_{\text{mix}})$.

average hitting time for almost-regular graphs. For our family of almost-regular graphs, there is a polynomial gap between t_{meet} and $t_{\text{avg-hit}}$. More importantly, we show that these almost-regular graphs have a gap of $\sqrt{t_{\text{mix}}/t_{\text{meet}}} \cdot \log n$ between coalescing and meeting time. This shows that the bound in Theorem 8.1 is best possible, even if we constrain G to be almost-regular. We refer the reader to Section 8.2 for a high-level description of the proof ideas.

More precisely, in the proof of Theorem 8.2 we shall give an explicit construction of a graph family $G = G_n$ with $t_{\text{coal}} = \Omega(\sqrt{\alpha_n} \cdot \log n \cdot t_{\text{mix}})$, where $\alpha_n = t_{\text{meet}}/t_{\text{mix}}$. For the remainder of this section, we will drop the dependence on n and will simply use G instead of G_n and α instead α_n.

The construction of G (see Figure 8.4 for an illustration) is based on two building blocks, G_1 and G_2. First, let $G_1 = (V_1, E_1)$ be a clique of size $\sqrt{n}$. Let $G_2 = (V_2, E_2)$ be a $\sqrt{n}$-regular bipartite Ramanujan Graph on $n/\sqrt{\alpha'}$ nodes [MSS15], where $\alpha' = \max\{\alpha, 2^{20} \cdot C^2\}$, where $C > 1$ is the universal constant of Corollary A.30. The graph G is made of one copy of G_2, $\kappa = \sqrt{n}$ copies of G_1 (denoted by $G_1^1, G_1^2, \ldots, G_1^\kappa$), and a node $\widehat{z}$, which has an edge to $\sqrt{n/\alpha'}$ distinct nodes of G_2 and to each of the designated nodes $z^i \in V_1^i$ in G_1^i for $i \in [1, \kappa]$. It is not difficult to see that this graph is almost-regular, *i. e.,* maximum and minimum degree differ by at most a constant.

In Lemma 8.13, Lemma 8.14, Lemma 8.15 and Lemma 8.16 respectively we show that $t_{\text{mix}} = \Theta(n)$, $t_{\text{meet}} = \Theta(\alpha' n)$, $t_{\text{coal}} = \Omega(\sqrt{\alpha'} \cdot n \log n)$, and $t_{\text{avg-hit}} = \Omega(n^{3/2})$. We start with the following auxiliary lemma which shows that the walk restricted to V_2 behaves similarly to the walk restricted to $V_2 \cup \{\widehat{z}\}$, meaning that the walks have very similar t-step probabilities.

Lemma 8.10. *Let P denote the transition matrix of the random walk on G, Q the transition matrix of the random walk on G_2 and $\widehat{Q}$ be the transition matrix of the random walk on the subgraph of G induced by $V_2 \cup \{\widehat{z}\}$. Let $S^* = \{u \in V_2 \cap N(\widehat{z})\}$. Then the following statements hold:*

(i) For any $u, v \in V_2$ we have $\|\mathbf{p}^t_{u,\cdot} - \mathbf{q}^t_{u,\cdot}\|_{TV} \leq \sum_{i=1}^{t-1} p^i_{u,S^*}/(2\sqrt{n}) \leq t/(2\sqrt{n})$.

(ii) For any $u, v \in V_2$ we have $\|\widehat{\mathbf{q}}^t_{u,\cdot} - \mathbf{q}^t_{u,\cdot}\|_{TV} \leq \sum_{i=1}^{t-1} p^i_{u,S^*}/(2\sqrt{n}) \leq t/(2\sqrt{n})$.

(iii) For any $u, v \in V_2$ we have that after $t = t_{\mathrm{mix}}(G_2)$ time steps $\|\mathbf{p}^t_{u,\cdot} - \mathbf{p}^t_{v,\cdot}\|_{TV} \leq o(1) + 2/e$.

Proof. Let $(X_t)_{t\geq 0}$ be the Markov chain with transition matrix P and let $(Y_t)_{t\geq 0}$ be the Markov chain with transition matrix Q. We will inductively couple these two random walks starting from $X_0 = Y_0 = u$. Given that we coupled both chains up to time $t-1$, we can couple (X_t, Y_t) such that $X_t = Y_t$ with an error probability

$$\begin{aligned}\mathbb{P}[\, X_t \neq Y_t \mid X_{t-1} = Y_{t-1}\,] &= \mathbb{P}[\, X_t \neq Y_t \mid X_{t-1} = Y_{t-1}, X_{t-1} \in S^*\,] \cdot \mathbb{P}[\, X_{t-1} \in S^*\,] \\ &\quad + \mathbb{P}[\, X_t \neq Y_t \mid X_{t-1} = Y_{t-1}, X_{t-1} \in V_2 \setminus S^*\,] \cdot \mathbb{P}[\, X_{t-1} \in V_2 \setminus S^*\,] \\ &\leq p^{t-1}_{u,S^*}/(2\sqrt{n}) + 0.\end{aligned}$$

We have, by [LPW06, Proposition 4.7],

$$\|\mathbf{p}^t_{u,\cdot} - \mathbf{p}^t_{v,\cdot}\|_{\mathrm{TV}} = \inf\{\mathbb{P}[\, X \neq Y\,] \mid (X, Y) \text{ is a coupling of } \mathbf{p}^t_{u,\cdot} \text{ and } \mathbf{p}^t_{v,\cdot}\}.$$

Hence, by a union bound over t steps,

$$\begin{aligned}\|\mathbf{p}^t_{u,\cdot} - \mathbf{p}^t_{v,\cdot}\|_{\mathrm{TV}} &= \inf\{\mathbb{P}[\, X \neq Y\,] \mid (X, Y) \text{ is a coupling of } \mathbf{p}^t_{u,\cdot} \text{ and } \mathbf{p}^t_{v,\cdot}\} \leq \mathbb{P}[\, X_t \neq Y_t\,] \\ &\leq \sum_{i=1}^{t-1} p^i_{u,S^*}/(2\sqrt{n}) \leq \frac{t}{2\sqrt{n}}.\end{aligned}$$

To prove the second part we redefine $(X_t)_{t\geq 0}$ to be the Markov chain with transition matrix $\widehat{Q}$ and the proof is identical.

We proceed with the last part. For $u, v \in V_2$ we have that after $t = t_{\mathrm{mix}}(G_2)$ time steps, by the triangle inequality and using that $t_{\mathrm{mix}}(G_2) = O(1)$, by Proposition A.29, we get

$$\begin{aligned}\|\mathbf{p}^t_{u,\cdot} - \boldsymbol{\pi}^{G_2}(\cdot)\|_{\mathrm{TV}} &\leq \|\mathbf{p}^t_{u,\cdot} - \mathbf{q}^t_{u,\cdot}\|_{\mathrm{TV}} + \|\mathbf{q}^t_{u,\cdot} - \boldsymbol{\pi}^{G_2}(\cdot)\|_{\mathrm{TV}} \\ &\leq \frac{t_{\mathrm{mix}}(G_2)}{2\sqrt{n}} + \|\mathbf{q}^t_{u,\cdot} - \boldsymbol{\pi}^{G_2}(\cdot)\|_{\mathrm{TV}} \\ &\leq o(1) + \|\mathbf{q}^t_{u,\cdot} - \boldsymbol{\pi}^{G_2}(\cdot)\|_{\mathrm{TV}} \leq o(1) + 1/e,\end{aligned}$$

where the last inequality follows form the definition of mixing time. Again, by the triangle inequality, $\|\mathbf{p}^t_{u,\cdot} - \mathbf{p}^t_{v,\cdot}\|_{\mathrm{TV}} \leq o(1) + 2/e$. □

Based on Lemma 8.10, we can now bound the hitting time to reach $\widehat{z}$, which will later be used to establish the bounds on the mixing and meeting time of the whole graph G. But first, we prove that the mixing time of the graph $\widehat{G}$ induced by $V_2 \cup \{\widehat{z}\}$ is constant and that after mixing on $\widehat{G}$, the random walk has a probability of $\Omega(1/n)$ to hit $\widehat{z}$ in a constant number of time steps.

Lemma 8.11. *The following three statements hold.*

(i) *Let $\widehat{G}$ be the induced graph by the vertices $V_2 \cup \{\widehat{z}\}$. Then $t_{\text{mix}}(\widehat{G}) = O(1)$.*

(ii) *Let $u \in V \setminus \{\widehat{z}\}$. Then there exists a constant $c \geq 1$ such that $\mathbb{P}[\, T_{\text{hit}}(u, \widehat{z}) \geq n/c\,] \geq 1/2$.*

(iii) *Let $u \in V \setminus \{\widehat{z}\}$. Then $t_{\text{hit}}(u, \widehat{z}) = O(n)$.*

Proof. We prove the statements one by one.

(i) Let Q be the transition matrix of a random walk restricted to G_2. Let $d^Q(t)$ be the total variation distance w.r.t. the transition matrix Q. Further, let $\widehat{Q}$ be the transition matrix of a random walk restricted to $\widehat{G}$. Recall that $t_{\text{mix}}(G_2) = O(1)$, by Proposition A.29.

Fix an arbitrary $t \in [2t_{\text{mix}}(G_2), 2t_{\text{mix}}(G_2) + 7]$. In the following we show $\|\widehat{\mathbf{q}}^t_{u,\cdot} - \boldsymbol{\pi}^{\widehat{G}}(\cdot)\|_{\text{TV}} \leq 1/e$. We first consider any start vertex $u \in V_2 \setminus \{\widehat{z}\}$ and afterwards the vertex $u = \widehat{z}$. Let $\mathcal{D}$ be the set of distributions over $V(\widehat{G}) = V_2 \cup \{\widehat{z}\}$ assigning no probability mass to $\widehat{z}$, *i. e.*,

$$\mathcal{D} = \{D' \colon \text{ for } u \sim D' \text{ we have } \mathbb{P}[\, u = \widehat{z}\,] = 0\}. \tag{8.12}$$

For any such $D' \in \mathcal{D}$, we have, by definition of the total variation distance,

$$\|\widehat{\mathbf{q}}^t_{u\sim D',\cdot} - \boldsymbol{\pi}^{\widehat{G}}(\cdot)\|_{\text{TV}} = 0 + \frac{1}{2}\sum_{v\in V_2}\left|\widehat{q}^t_{u\sim D',v} - \pi^{\widehat{G}}(v)\right| + \frac{1}{2}\left|\widehat{q}^t_{u\sim D',\widehat{z}} - \pi^{\widehat{G}}(\widehat{z})\right|.$$

For $u \in V_2$ observe that $\pi^{\widehat{G}}(u) \in [\pi^{G_2}(u)(1-\zeta), \pi^{G_2}(u)(1+\zeta)]$ for some $\zeta = o(1)$. By [LPW06, Exercise 4.1] we have the following identity for $d^Q(t)$. Let $\mathcal{D}^*$ be the set of all distributions over $V(G_2)$, then

$$d^Q(t) = \max_{D\in\mathcal{D}^*} \|\mathbf{q}^t_{u\sim D,\cdot} - \boldsymbol{\pi}^{G_2}(\cdot)\|_{\text{TV}} \geq \max_{D'\in\mathcal{D}} \|\mathbf{q}^t_{u\sim D',\cdot} - \boldsymbol{\pi}^{G_2}(\cdot)\|_{\text{TV}}.$$

Thus, for $\delta_v := |\widehat{q}^t_{u,v} - q^t_{u,v}|$, we get by using triangle inequality,

$$\begin{aligned}
\frac{1}{2}\sum_{v\in V_2}\left|\widehat{q}^t_{u\sim D',v} - \pi^{\widehat{G}}(v)\right| &\le \frac{1}{2}\sum_{v\in V_2}\left|\widehat{q}^t_{u\sim D',v} - \pi^{G_2}(v)\right| + \frac{1}{2}\sum_{v\in V_2}|\pi^{G_2}(v) - \pi^{\widehat{G}}(v)| \\
&\le \frac{1}{2}\sum_{v\in V_2}\left|\widehat{q}^t_{u\sim D',v} - \pi^{G_2}(v)\right| + \frac{1}{2}\sum_{v\in V_2}\pi^{G_2}(v)\zeta \\
&\le \frac{1}{2}\sum_{v\in V_2}\left|q^t_{u\sim D',v} - \pi^{G_2}(v)\right| + \frac{1}{2}\sum_{v\in V_2}|\delta_v| + \frac{1}{2}\sum_{v\in V_2}\pi^{G_2}(v)|\zeta| \\
&\le d^Q(t) + 1/32 + \frac{\zeta}{2}, \\
&\le d^Q(t) + 1/32 + 1/32, \qquad (8.13)
\end{aligned}$$

where the second-last inequality is due to Lemma 8.10.(ii), $\frac{1}{2}\sum_{v\in V}|\delta_v| \le t/(2\sqrt{n}) \le \frac{1}{32}$. By definition of the $t_{\mathrm{mix}}(G_2)$ and by sub-multiplicativity we have $d^Q(t) \le d^Q(2t_{\mathrm{mix}}(G_2)) \le 1/e^2$.

The above equation (8.13) only consider the variation distance w.r.t. V_2. For $\widehat{z}$ we have $\frac{1}{2}|\widehat{q}^t_{u\sim D',\widehat{z}} - \pi^{\widehat{G}}(\widehat{z})| \le (2t_{\mathrm{mix}}(G_2)+7)/\sqrt{n} \le 1/32$.

Putting everything together we get we get

$$\begin{aligned}
\|\widehat{\mathbf{q}}^t_{u\sim D',\cdot} - \boldsymbol{\pi}^{\widehat{G}}(\cdot)\|_{\mathrm{TV}} &= \frac{1}{2}\sum_{v\in V_2}\left|\widehat{q}^t_{u\sim D',v} - \pi^{\widehat{G}}(v)\right| + \frac{1}{2}\left|\widehat{q}^t_{u\sim D',\widehat{z}} - \pi^{\widehat{G}}(\widehat{z})\right| \\
&\le d^Q(t) + 1/32 + 1/32 + 1/32 \le 1/e^2 + 3/32 \qquad (8.14) \\
&\le 1/e. \qquad (8.15)
\end{aligned}$$

Consider the random walk starting at $\widehat{z}$ and let $(X_0, X_1, \dots)$ denote its trajectory. Observe that at time 7 we have

$$\widehat{q}^7_{\widehat{z},\widehat{z}} \le \frac{1}{2^7} + \sum_{i\le 7}\sum_{v\in N(\widehat{z})}\widehat{q}^{i-1}_{\widehat{z},v}\cdot\frac{1}{2(\sqrt{n}+1)} \le \frac{1}{2^7} + \frac{7^2}{\sqrt{n}} \le 1/32.$$

The set of distribution for the position of the random walk at time 7 conditioning on $X_7 \neq \widehat{z}$ gives the same distribution $\mathcal{D}$ as defined in (8.12). Let $D_{\widehat{z}} \in \mathcal{D}$ be distribution of the random at time 7 starting at $\widehat{z}$. Hence, by(8.14), we get

$$\begin{aligned}
\|\widehat{\mathbf{q}}^{2t_{\mathrm{mix}}(G_2)+7}_{\widehat{z},\cdot} - \boldsymbol{\pi}^{\widehat{G}}(\cdot)\|_{\mathrm{TV}} &\le \widehat{q}_{\widehat{z},V(\widehat{G})\setminus\{\widehat{z}\}}\cdot\|\widehat{\mathbf{q}}^{2t_{\mathrm{mix}}(G_2)}_{u\sim D_{\widehat{z}},\cdot} - \boldsymbol{\pi}^{\widehat{G}}(\cdot)\|_{\mathrm{TV}} + \widehat{q}_{\widehat{z},\widehat{z}}\cdot 1 \qquad (8.16) \\
&\le 1\cdot(1/e^2 + 3/32) + 1/32 \le 1/e. \qquad (8.17)
\end{aligned}$$

Thus, for $t' = 2t_{\text{mix}}(G_2) + 7$ we have $\|\widehat{\mathbf{q}}^{t'}_{\widehat{z},\cdot} - \boldsymbol{\pi}^{\widehat{G}}(\cdot)\|_{\text{TV}} \leq 1/e$. Together with (8.14), we conclude that for all $u \in V'$, $\|\widehat{\mathbf{q}}^{t'}_{u,\cdot} - \boldsymbol{\pi}^{\widehat{G}}(\cdot)\|_{\text{TV}} \leq 1/e$ and by definition of t_{mix} and we get $t_{\text{mix}}(\widehat{G}) \leq 2t_{\text{mix}} + 7 = O(1)$.

(ii) To prove $\mathbb{P}[\, T_{\text{hit}}(u, \widehat{z}) \geq n/c \,] \geq 1/2$ for $u \in V_2$ we show that the random walk restricted to $\widehat{G}$ does not hit $\widehat{z}$ after n/c_1 steps w.c.p. for some large enough constant c_1. By the Union bound, for some large constants c_1, c_2 that

$$\mathbb{P}\Big[\, T^G_{\text{hit}}(u, \widehat{z}) \leq n/c_1 \,\Big] = \mathbb{P}\Big[\, T^{\widehat{G}}_{\text{hit}}(u, \widehat{z}) \leq n/c_1 \,\Big] \leq \sum_{t=1}^{n/c_1} \widehat{q}^{\,t}_{u,\widehat{z}} \leq \sum_{t=1}^{c_2 \log n} 1/\sqrt{n} + \sum_{t=c_2 \log n}^{n/c_1} \widehat{q}^{\,t}_{u,\widehat{z}}$$
$$\leq o(1) + n/c_1 \cdot (\pi^{\widehat{G}}(\widehat{z}) + 1/n^2) \leq 1/2,$$

where we used $\widehat{q}^{\,t}_{u,\widehat{z}} \leq \pi^{\widehat{G}}(\widehat{z}) + \sqrt{\frac{\pi^{\widehat{G}}(\widehat{z})}{\pi^{\widehat{G}}(u)}}\lambda_2(\widehat{G})^t$ (Proposition A.26).

We proceed by bounding that $\mathbb{P}[\, T_{\text{hit}}(u, \widehat{z}) \geq n/c_1 \,] \geq 1/2$ for $u \in V_1$. Consider first a random walk $(\tilde{X}_t)_{t \geq 0}$ restricted to $G^1_1 = G_1$ that starts at vertex z^1 and let $\tilde{P}$ denote the transition matrix. Furthermore, in order to couple the random walk $\tilde{X}_t$ restricted to G_1 with a random walk in G, we will consider the random variable $\tilde{Z} := \sum_{t=0}^{t^{G_1}_{\text{sep}}} \mathbf{1}_{\tilde{X}_t = z_1}$. Since G_1 is a clique, $t^{G_1}_{\text{sep}} = O(1)$, and $\tilde{p}^t_{z_1,z_1} \leq \frac{1}{\sqrt{n}} + \lambda_2(G_1)^t$ by Proposition A.26, where $\lambda_2(G_1)$ is some constant bounded away form 1. Therefore, $\mathbb{E}\Big[\, \tilde{Z} \,\Big] = \sum_{t=0}^{n/c_1} \tilde{p}^t_{z_1,z_1} \leq 2\sqrt{n}/c_1$. Let $\gamma := 4 \cdot \mathbb{E}\Big[\, \tilde{Z} \,\Big]$. Then, by Markov's inequality

$$\mathbb{P}\Big[\, \tilde{Z} \geq \gamma \,\Big] \leq 1/4.$$

Consider now the straightforward coupling between a random walk $(X_t)_{t \geq 1}$ in G that starts at vertex z^1 and the random walk $(\tilde{X}_t)_{t \geq 1}$ restricted to G^i_1 that starts at the same vertex. Whenever the random walk $\tilde{X}_t$ is at a vertex different from z^1, then the random walk X_t makes the same transition. If the random walk $\tilde{X}_t$ is at vertex z^1, then there is a coupling so that the random walk X_t makes the same transition as $\tilde{X}_t$ with probability $\frac{2\sqrt{n}-1}{2\sqrt{n}}$. Conditional on the event $\tilde{Z} \leq \gamma$ occurring, the random walk $\tilde{X}_t$ follows the random walk X_t up until step n/c_1 with probability at least

$$p := \left(\frac{2\sqrt{n}-1}{2\sqrt{n}}\right)^{\gamma} \geq 3/4,$$

since the random walk $\tilde{X}_t$ has at most γ visits to z^1. Therefore, by the Union bound,

$$\mathbb{P}\Big[\, T^G_{\text{hit}}(u, \widehat{z}) \geq n/c_1 \,\Big] \geq \mathbb{P}\Big[\, \cup_{t=0}^{n/c_1} X_t = \tilde{X}_t \,\Big] \geq 1 - \mathbb{P}\Big[\, \tilde{Z} \geq \gamma \,\Big] - (1-p) \geq 1/2$$

and the proof is complete.

(iii) We proceed by showing $t_{\text{hit}}(u,\widehat{z}) = O(n)$ for $u \in V_2$.

Let Q be the transition matrix of the random walk restricted to G_2. Let $u \in V_2$ and $S^* = N(\widehat{z})$ be the neighbors of $\widehat{z}$ in G_2. For every $v \in S^*$ we have $\pi^{G_2}(v) = \frac{\sqrt{n}+1}{\frac{n}{\sqrt{\alpha'}}\sqrt{n}+\frac{\sqrt{n}}{\sqrt{\alpha'}}} \geq \frac{\sqrt{\alpha'}}{1.2n}$. Hence, after $t = t_{\text{sep}}(G_2)$ we have that

$$q^t_{u,S^*} := \sum_{v\in S^*} q^t_{u,v} \geq \sum_{v\in S^*} \pi^{G_2}(v)(1-e^{-1}) \geq \frac{\sqrt{n}}{\sqrt{\alpha'}} \cdot \frac{\sqrt{\alpha'}}{1.2n}(1-e^{-1}) = \frac{1-e^{-1}}{1.2\sqrt{n}}.$$

By Lemma 8.10, we have for any $u \in V_2$ that $\|\mathbf{p}^t_{u,\cdot} - \mathbf{q}^t_{u,\cdot}\|_{\text{TV}} \leq t_{\text{sep}}(G_2)/(2\sqrt{n})$. To bound $T^G_{\text{hit}}(u,\widehat{z})$ we show that after $t_{\text{sep}}+1 = O(1)$ steps the random walk hits $\widehat{z}$ w.p. $\Omega(1/n)$.

We distinguish between two cases.

(a) For all $i \leq t$ we have $p^t_{u,S^*} \leq 1/t_{\text{sep}}(G_2)$. Thus, by Lemma 8.10.(i)

$$\begin{aligned} p^t_{u,S^*} = \sum_{v\in S^*} p^t_{u,v} &\geq q^t_{u,S^*} - \|\mathbf{p}^t_{u,\cdot} - \mathbf{q}^t_{u,\cdot}\|_{\text{TV}} \\ &\geq \frac{1-e^{-1}}{1.2\sqrt{n}} - \sum_{i=1}^{t-1} p^i_{u,S^*}/(2\sqrt{n}) \\ &\geq \frac{1-e^{-1}}{1.2\sqrt{n}} - \frac{t_{\text{sep}}(G_2)}{t_{\text{sep}}(G_2)2\sqrt{n}} = \Omega(1/\sqrt{n}). \end{aligned}$$

Hence, the random walk hits $\widehat{z}$ after $t_{\text{sep}}(G_2)+1$ w.p. at least $p^t_{u,S^*} \cdot \min_{v\in S^*}\{p_{v,\widehat{z}}\} = \Omega(1/n)$.

(b) Otherwise there exists a t^* such that $p^{t^*}_{u,S^*} > 1/t_{\text{sep}}(G_2)$. Thus the random walk hits $\widehat{z}$ after $t_{\text{sep}}(G_2)+1$ w.p. at least $p^{t^*}_{u,S^*} \cdot \min_{v\in S^*}\{p_{v,\widehat{z}}\} = \Omega(1/n)$.

Thus after $O(1)$ steps the random walk hits $\widehat{z}$ w.p. $\Omega(1/n)$.

We now show a similar statement if $u \in V_1$. Let $(X_t)_{t\geq 0}$ be a random walk on G starting on u. Observe that X_t (the walk on G) hits $\widehat{z}$ with probability $p^1_{u,z^1} \cdot p^1_{z^1,\widehat{z}} = \Omega(1/n)$ in 2 time steps. Hence, for any $u \in V$ we $\mathbb{P}[\, T_{\text{hit}}(u,\widehat{z}) = O(1)\,] = \Omega(1/n)$. Thus, repeating this iteratively and using independence yields $t_{\text{hit}}(u,\widehat{z}) = O(n)$ for $u \in V$.

□

To establish a bound on the mixing time of G, we will make use of a following result of Peres and Sousi [PS15] (Theorem A.31 in Appendice A) to relate the mixing time of a graph to the hitting time of large sets. Peres and Sousi [PS15] show the following. For any $\beta < 1/2$, let $t_H(\beta) = \max_{u,A:\pi(A)\geq\beta} t_{\text{hit}}(u,A)$. Then there exist positive constants c_β and

c'_β such that

$$c'_\beta \cdot t_H(\beta) \leq t_{\text{mix}}(1/4) \leq c_\beta \cdot t_H(\beta).$$

In the following we show for any β close enough to $1/2$, that any $A \subseteq V$ satisfying $\pi(A) \geq \beta$ must include at least a constant fraction of nodes from a constant fraction of copies of G_1.

Claim 8.12. *Let $\beta = 1/2 - 10^{-3}$. For any $A \subseteq V$ with $\pi(A) \geq \beta$, define $H(A) = \{i \mid |G_1^i \cap A| \geq |V_1|/(2e)\}$. Then, $|H(A)| \geq \kappa/(2e)$.*

Proof. This follows from a simple pigeon-hole argument: Suppose $|H(A)| < \kappa/(2e)$ was true. Then,

$$\begin{aligned}\pi(A) &\leq |H(A)| \cdot \pi(V_1) + (\kappa - |H(A)|) \cdot \left(\frac{\pi(V_1)}{2e} + \pi(z^i)\right) + \pi(V_2) + \pi(\widehat{z}) \\ &< \frac{\kappa}{2e} \cdot \pi(V_1) + \kappa \cdot \left(\frac{\pi(V_1)}{2e} + \pi(z^i)\right) + 1/20 < \beta \leq \pi(A),\end{aligned}$$

which is a contradiction and hence choice of A must fulfill $|H(A)| \geq \kappa/(2e)$. □

We are now ready to determine the mixing time of G. The lower bound is a simple application of Cheeger's inequality, while the upper bound combines the previous lemmas with Theorem A.31.

Lemma 8.13. *Let G be the graph described at the beginning of Section 8.3.5. We have $t_{\text{mix}}(G) = \Theta(n)$.*

Proof. First we show $t_{\text{mix}} = \Omega(n)$. The *conductance* of $G = (V, E)$ is defined by $\Phi(G) = \min_{\substack{U \subseteq V, \\ 0 < \text{vol}(U) \leq \text{vol}(V)/2}} \frac{|E(U, V \setminus U)|}{\text{vol}(U)}$. In particular, for $U = V_1$ we get that $\Phi(G) \leq \frac{4}{n}$. Hence, by Cheeger's inequality and $\left(\frac{1}{1-\lambda_2(G)} - 1\right) \cdot \log(\frac{e}{2}) \leq t_{\text{mix}}(1/e)$ (see, *e. g.*, [LPW06, Chapter 12]),

$$\frac{n}{4} \leq \frac{1}{\Phi(G)} \leq \frac{2}{1 - \lambda_2(G)} = \frac{2}{1 - \lambda_2(G)} - 2 + 2 \leq \frac{2t_{\text{mix}}}{\log(\frac{e}{2})} + 2.$$

Rearranging the terms yields $t_{\text{mix}} = \Omega(n)$.

We proceed with the upper bound on the mixing time. Let $\beta = 1/2 - 10^{-3}$ and let $A \subseteq V$ be an arbitrary set satisfying $\pi(A) \geq \beta$. First, we apply Claim 8.12 to conclude that $|H(A)| \geq \kappa/(2e)$. This immediately implies that with $Z := \{z^i \colon i \in H(A)\}$, $|Z| \geq \kappa/(2e)$. The remainder of the proof is divided into the following three parts:

(i) Starting from any vertex $u \in V$, with probability at least $1/2$, the random walk hits z^* after $2 \max_{u \in V} t_{\text{hit}}(u, \widehat{z}) = O(n)$ steps.

(ii) With constant probability $p_1 > 0$, the random walk moves from z^* to a vertex in Z.

(iii) With constant probability $p_2 > 0$ a random walk starting from a vertex in Z will hit A after one step.

It is clear that combining these three results shows that with constant probability $\frac{1}{2}p_1p_2 > 0$, a random walk starting from an arbitrary vertex $u \in V$ hits a vertex in A after $O(n)+1+1$ time steps. Iterating this and using independence shows that $t_{\text{hit}}(u, A) = O(n)$, and hence by Theorem A.31, $t_{\text{mix}} = O(n)$ as needed.

Part (i). Consider $\max_{u\in V} t_{\text{hit}}(u, \widehat{z})$. For $u \in V$, Lemma 8.11.(iii) implies $t_{\text{hit}}(u, \widehat{z}) = O(n)$.

Part (ii). If the random walk is on z^*, then since $\deg(z^*) = \kappa + \sqrt{n/\alpha'}$, $|Z| \geq \kappa/(2e)$, it follows that the random walk hits a vertex in Z after one step with constant probability $p_1 := \frac{|Z|}{2(\kappa+\sqrt{n/\alpha'})} > 0$.

Part (iii). Finally, for any $z \in Z$ we have that $p_2 = p_{z,A} = \frac{|V_1|/(2e)}{2\sqrt{n}} > 0$ and the proof is complete. □

In the following we establish the bound on the meeting time. As it turns out, any meeting is very likely to happen on V_2 and it takes about $\Theta(\alpha' n)$ time steps until both walks reach V_2 simultaneously. The lower bound then follows from our common analysis method (8.1). The upper bound combines the mixing time bound of $O(n)$ (Lemma 8.13), and that once a random walk reaches a copy of G_1, it says there for $\Theta(n)$ steps with constant probability Lemma 8.11.(ii).

Lemma 8.14. *Let G be the graph described at the beginning of Section 8.3.5. We have $t_{\text{meet}}(G) = \Theta(\alpha' n)$.*

Proof. We start by proving $t_{\text{meet}} = \Omega(\alpha' n)$: Consider two non-interactingDEGreplace these by something, random walks with starting positions drawn from the stationary distribution $\boldsymbol{\pi}$. Let $\ell = c'\alpha' n$, for some small enough constant $c' > 0$. Let Z_1 be the number of collisions of the two random walks on the nodes in $V_1^1 \cup V_1^2 \cup \cdots \cup V_1^\kappa$. Let Z_2 be the number of collisions of the two random walks on the nodes in V_2. Let Z_* be the number of collisions of the two random walks on the node $\widehat{z}$.

Let Z be the number of collisions of the two walks during the first ℓ time steps, *i. e.*, $Z = Z_1 + Z_2 + Z_*$. Using the Union bound we derive

$$\begin{aligned}\mathbb{P}[\, Z \geq 1\,] &\leq \mathbb{P}[\, Z_1 \geq 1\,] + \mathbb{P}[\, Z_2 \geq 1\,] + \mathbb{P}[\, Z_* \geq 1\,] \\ &\leq \frac{\mathbb{E}[\, Z_1\,]}{\mathbb{E}[\, Z_1 \,|\, Z_1 \geq 1\,]} + \frac{\mathbb{E}[\, Z_2\,]}{\mathbb{E}[\, Z_2 \,|\, Z_2 \geq 1\,]} + \frac{\mathbb{E}[\, Z_*\,]}{\mathbb{E}[\, Z_* \,|\, Z_* \geq 1\,]}. \end{aligned} \tag{8.18}$$

We have $\mathbb{E}[\, Z_1\,] \leq \ell n\left(\frac{2}{n}\right)^2$, $\mathbb{E}[\, Z_2\,] \leq \ell\frac{n}{\sqrt{\alpha'}}\left(\frac{2}{n}\right)^2$, and $\mathbb{E}[\, Z_*\,] \leq \ell\left(\frac{2}{n}\right)^2$, since $\max_u \pi(u) \leq 2/n$. Conditioning on $Z_1 \geq 1$ and since both random walks start from the stationary

distribution, we have, by Observation A.25, that the first meeting happens in the first $\ell/2$ time steps w.p. at least $1/2$.

Consider $\mathbb{E}[\, Z_1 \,|\, Z_1 \geq 1\,]$. Suppose the meeting occurred at node $u \in V_1$. Let $\mathcal{E}_1$ be the event that for $u \in V_1$ we have $T_{\text{hit}}(u, \widehat{z}) \geq n/c$ for both walks, where $c > 0$ is a large enough constant. By Lemma 8.11.(ii), we have that $\mathbb{P}[\,\mathcal{E}_1\,] \geq (1/2)^2 = 1/4$ due to independence of the walks. For any $t < n/c$ let $\widehat{\mathbf{p}}^t_{u,\cdot}$ be the distribution of the random walk on G_1 starting on u after t time steps under the conditioning $\mathcal{E}_1$. Observe that $\sum_{v \in V_1} \widehat{p}^t_{u,v} = 1$ implying that $\sum_{v \in V_1} (\widehat{p}^t_{u,v})^2 \geq \sum_{v \in V_1} \left(\frac{1}{|V_1|}\right)^2 = 1/|V_1|$. Hence, we get

$$\mathbb{E}[\, Z_1 \,|\, Z_1 \geq 1\,] \geq \mathbb{E}[\, Z_1 \,|\, Z_1 \geq 1, \mathcal{E}_1\,] \cdot \mathbb{P}[\,\mathcal{E}_1\,] \geq \frac{1}{2} \min_{u \in V_1} \sum_{t=0}^{n/c-1} \sum_{v \in V_1} (\widehat{p}^t_{u,v})^2 \geq \frac{1}{4} \sum_{t=0}^{n/c-1} 1/|V_1| = \frac{\sqrt{n}}{4c}.$$

Using an exactly analogous analysis for Z_2 we can upper bound $\mathbb{E}[\, Z_2 \,|\, Z_2 \geq 1\,]$ as follows:

$$\mathbb{E}[\, Z_2 \,|\, Z_2 \geq 1\,] \geq \mathbb{E}[\, Z_2 \,|\, Z_2 \geq 1, \mathcal{E}_2\,] \cdot \mathbb{P}[\,\mathcal{E}_2\,] \geq \frac{1}{4} \min_{u \in V_2} \sum_{t=0}^{n/c-1} \sum_{v \in V_2} (\widehat{p}^t_{u,v})^2 \geq \frac{1}{4} \sum_{t=0}^{n/c-1} 1/|V_2| = \frac{\sqrt{\alpha'}}{4c},$$

where $\mathcal{E}_2$ is the event that for $u \in V_2$ we have $T_{\text{hit}}(u, \widehat{z}) \geq n/c$ for some large enough constant c. Plugging everything into (8.18) and using $\ell = c'\alpha' n$ yields

$$\begin{aligned}
\mathbb{P}[\, Z \geq 1\,] &\leq \frac{\mathbb{E}[\, Z_1\,]}{\mathbb{E}[\, Z_1 \,|\, Z_1 \geq 1\,]} + \frac{\mathbb{E}[\, Z_2\,]}{\mathbb{E}[\, Z_2 \,|\, Z_2 \geq 1\,]} + \frac{\mathbb{E}[\, Z_*\,]}{\mathbb{E}[\, Z_* \,|\, Z_* \geq 1\,]} \\
&\leq \frac{\ell n \left(\frac{2}{n}\right)^2}{\frac{\sqrt{n}}{4c}} + \frac{\ell \frac{n}{\sqrt{\alpha'}} \left(\frac{2}{n}\right)^2}{\frac{\sqrt{\alpha'}}{4c}} + \frac{\ell \left(\frac{2}{n}\right)^2}{1} \\
&\leq o(1) + 16c \cdot c' + o(1) \leq 1/2,
\end{aligned}$$

for any constant $c' \in (0, \frac{1}{33c}]$. This finishes the proof of $t_{\text{meet}} = \Omega(\alpha' n)$. In the remainder we prove $t_{\text{meet}} = O(\alpha' n)$. Consider two independent walks $(X_t)_{t \geq 0}$ and $(Y_t)_{t \geq 0}$ on G, both starting from arbitrary nodes. Note $t_{\text{sep}} = t_{\text{sep}}(G) \leq 4 t_{\text{mix}} = O(n)$ by Lemma 8.13, and

$$p_0 := \mathbb{P}[\, \{X_{t_{\text{sep}}} \in V_2\} \cap \{Y_{t_{\text{sep}}} \in V_2\}\,] \geq \left(\sum_{u \in V_2} (1 - e)\pi(u) \right)^2 = \Omega\left(\left(1/\sqrt{\alpha'}\right)^2 \right) = \Omega(1/\alpha').$$

We assume in the following that $\{X_{t_{\text{sep}}} \in V_2\} \cap \{Y_{t_{\text{sep}}} \in V_2\}$. We have $t_{\text{mix}}(G_2) = O(1)$, by Proposition A.29. Consider a random walk $(\tilde{X}_t)_{t \geq t_{\text{sep}}}$ restricted to G_2 that starts at vertex $X_{t_{\text{sep}}} \in V_2$ and let $\tilde{P}$ denote the transition matrix. Furthermore, in order to couple the random walk $\tilde{X}_t$ restricted to G_2 with a random walk in G, we will consider the random variable

$$\tilde{Z} := \sum_{t=t_{\text{sep}}}^{t_{\text{sep}}+n/c-1} \sum_{z \in N(\widehat{z})} \mathbf{1}_{\tilde{X}_t = z},$$

for $c = 32$. Thus, for any $z \in N(\widehat{z})$,

$$\begin{aligned}\mathbb{E}\Big[\,\tilde{Z}\,\Big] &\leq t_{\mathrm{mix}}(G_2) + \sum_{t=t_{\mathrm{sep}}+t_{\mathrm{mix}}(G_2)+1}^{t_{\mathrm{sep}}+n/c-1} |N(\widehat{z})|(\pi^{G_2}(z) + d^{\tilde{P}}(t)) \\ &\leq t_{\mathrm{mix}}(G_2) + |N(\widehat{z})|(n/c) + O(1) \leq (1+1/e)\sqrt{n}/c.\end{aligned}$$

Let $\gamma := 8(1+1/e)\sqrt{n}/c$. Then, by Markov's inequality

$$\mathbb{P}\Big[\,\tilde{Z} \geq \gamma\,\Big] \leq 1/8.$$

Consider now the straightforward coupling between a random walk $(X_t)_{t\geq t_{\mathrm{sep}}}$ in G that starts at vertex $\tilde{X}_{t_{\mathrm{sep}}} \in V_2$ and the random walk $(\tilde{X}_t)_{t\geq t_{\mathrm{sep}}}$ restricted to G_2 that starts at the same vertex. Whenever the random walk $\tilde{X}_t$ is at a vertex in $V_2 \setminus \{N(\widehat{z})\}$, then the random walk X_t makes the same transition. If the random walk $\tilde{X}_t$ is at vertex $z' \in N(\widehat{z})$, then there is a coupling so that the random walk X_t makes the same transition as $\tilde{X}_t$ with probability $\frac{2\sqrt{n}}{2\sqrt{n}+2}$. Conditional on the event $\{\tilde{Z} \leq \gamma\}$ occurring, the random walk $\tilde{X}_t$ follows the random walk X_t up until step n/c with probability at least

$$p_1 := \left(\frac{2\sqrt{n}}{2\sqrt{n}+2}\right)^{\gamma} = \left(1 - \frac{1}{\sqrt{n}+1}\right)^{\gamma} \geq \frac{3}{4},$$

since the random walk $\tilde{X}_t$ has at most γ visits to $N(\widehat{z})$. Consider now the random walk $(\tilde{Y}_t)_{t\geq t_{\mathrm{sep}}}$ using $\tilde{P}$ (*i. e.,* restricted to V_2) starting at $Y_{t_{\mathrm{sep}}}$, i.e., $\tilde{Y}_{t_{\mathrm{sep}}} = Y_{t_{\mathrm{sep}}}$. By an analogous argument as before we can couple $(Y_t)_{t\geq t_{\mathrm{sep}}}$ and $(\tilde{Y}_t)_{t\geq t_{\mathrm{sep}}}$ for n/c time steps w.p. at least p_1.

Furthermore, after $t_{\mathrm{sep}}(G_2) = O(1)$ steps we can couple $\tilde{X}_t$ and $\tilde{Y}_t$ with nodes drawn independently from $\boldsymbol{\pi}^{G_2}$. Hence,

$$p_2 := \mathbb{P}\Big[\,\tilde{X}_{t+t_{\mathrm{sep}}(G_2)} = \tilde{Y}_{t+t_{\mathrm{sep}}(G_2)} \mid \mathcal{F}_t\,\Big] \geq (1-1/e)^2 \|\boldsymbol{\pi}^{G_2}\|_2^2 \geq \frac{\sqrt{\alpha'}}{8n}.$$

Recall that $\alpha' \geq 2^{20} t_{\mathrm{sep}}(G_2)^2$ by definition. Therefore, the probability that $\tilde{X}_t$ and $\tilde{Y}_t$ do not meet in the time-interval $[t_{\mathrm{sep}}(G_1), t_{\mathrm{sep}}(G_1) + n/c - 1]$ is at most

$$p_3 := (1-p_2)^{\lfloor n/(t_{\mathrm{sep}}(G_2)c)\rfloor} \leq (1-p_2)^{\left\lfloor 2^{10} n/(\sqrt{\alpha'}c)\right\rfloor} \leq 1/4.$$

Therefore, by the Union bound,

$$\mathbb{P}\Big[\,\cup_{t=0}^{t_{\mathrm{sep}}(G_1)+n/c-1} X_t = Y_t\,\Big] \geq p_0 \cdot \Big(1 - \mathbb{P}\Big[\,\tilde{Z} \geq \gamma\,\Big] - 2\cdot(1-p_1) - p_3\Big) = \Omega(\alpha').$$

Repeating this $O(1/p_3)$ times and using the independence yields that the expected meeting time is $O((t_{\mathrm{sep}}(G_1) + n/c - 1)/p_3) = O(\alpha' n)$ and the proof is complete.

□

Finally, we analyze the coalescing time of G. The proof idea is to consider $\sqrt[5]{n}$ random walks starting from $\boldsymbol{\pi}$ and show that meetings only occur on V_2 and that at least one random walk requires $\Omega(\sqrt{\alpha'} \cdot n \log n)$ time steps to reach V_2.

Lemma 8.15. *Let G be the graph described at the beginning of Section 8.3.5. We have $t_{\text{coal}}(G) = \Omega(\sqrt{\alpha'} \cdot n \log n)$.*

Proof. Let $\varepsilon = 1/5$. We show that even the coalescing time of n^ε random walks requires $\Omega(\sqrt{\alpha'} \cdot n \log n)$ time steps w.c.p.. Let R be a collection of n^ε independent, *i. e.,* non-interacting, random walks with starting positions drawn from the stationary distribution $\boldsymbol{\pi}$. We define the following three bad events:

(i) Let $\mathcal{E}_1$ be the event that any of the n^ε random walks meet on a node $V \setminus V_2$ in $\sqrt{\alpha'} \cdot n \log^2 n$ steps.

(ii) Let $\mathcal{E}_2$ be the event that fewer than $n^\varepsilon/4$ random walks start on copies of G_1, *i. e.,* on nodes $V \setminus (V_2 \cup \widehat{z})$.

(iii) Let $\mathcal{E}_3$ be the event that all random walks starting from a copy of G_1 require fewer than $c \cdot \sqrt{\alpha'} \cdot n \log n$ time steps for leaving $V \setminus (V_2 \cup z^*)$ for some constant $c > 0$ to be determined later.

In the following we show that $\mathbb{P}[\,\mathcal{E}_1\,] = o(1)$, $\mathbb{P}[\,\mathcal{E}_2\,] = o(1)$, and $\mathbb{P}\Big[\,\mathcal{E}_3 \mid \overline{\mathcal{E}_2}\,\Big] < 1/e$, which implies, by union bound,

$$\begin{aligned}\mathbb{P}\Big[\,\overline{\mathcal{E}_1} \cap \overline{\mathcal{E}_2} \cap \overline{\mathcal{E}_3}\,\Big] &\geq \mathbb{P}\Big[\,\overline{\mathcal{E}_1}\,\Big] - (1 - \mathbb{P}\Big[\,\overline{\mathcal{E}_2} \cap \overline{\mathcal{E}_3}\,\Big]) \\ \geq 1 - o(1) - \left(1 - (1 - o(1)) \cdot \left(1 - \frac{1}{e}\right)\right) &\geq 1 - \frac{1}{2e}.\end{aligned}$$

Conditioning on $\overline{\mathcal{E}_1} \cap \overline{\mathcal{E}_2} \cap \overline{\mathcal{E}_3}$, none of the independent random walks meet on any node $V \setminus V_2$ and hence they are indistinguishable from coalescing random walks until they reach V_2. Therefore, it is necessary for all random walks to reach G_2 in order to coalesce. Hence, we conclude that $t_{\text{coal}}(G) = \Omega(\sqrt{\alpha'} \cdot n \log n)$ yielding the lemma.

(i) We now prove $\mathbb{P}[\,\mathcal{E}_1\,] = o(1)$. Consider any pair of the random walks R. Since both random walks start from the stationary distribution, the probability for them to meet on a node on $\widehat{z}$ in a fixed step $t \geq 0$ is at most $O(1/n^2)$.

Hence, by the Union bound over $\binom{n^\epsilon}{2}$ pairs of random walks and $\sqrt{\alpha'} \cdot n \log^2 n \leq n \log^3 n$ steps, the probability of any two random walks meeting on $\widehat{z}$ is at most

$$p_1 := \binom{n^\epsilon}{2} \cdot n \log^3 n \cdot O(1/n^2) = o(1),$$

since $\epsilon = \frac{1}{5}$. Furthermore, the probability that no two walks start on the same copy of G_1 is at most $p_2 := n^{\varepsilon} \cdot \frac{n^{\varepsilon}}{\sqrt{n}} = o(1)$ by the Union bound.

Moreover, using a Chernoff bound together with Lemma 8.11.(ii), it follows that a random walks visits the vertex z^* at most $10 \log^3 n$ times during $n \log^3 n$ steps with probability at least $1 - n^{-2}$. By the Union bound over all random walks, it follows that w.p. at least $1 - n^{-1}$, each random walk visits at most $10 \log^3 n$ different copies of G_1, and by construction of G each such copy is chosen uniformly and independently at random among $G_1^1, G_1^2, \ldots, G_1^{\kappa}$. Therefore, the probability that there exists a copy of G_1 which is visited by at least two random walks in $n \log^3 n$ steps is at most

$$p_3 := n^{-1} + n^{\varepsilon}(10 \log^3 n + 1) \cdot \frac{n^{\varepsilon}(10 \log^3 n + 1)}{\sqrt{n}} = o(1). \tag{8.19}$$

Putting everything together, using union bound, yields $\mathbb{P}[\,\mathcal{E}_1\,] \leq p_1 + p_2 + p_3 = o(1)$.

(ii) We now prove $\mathbb{P}[\,\mathcal{E}_2\,] = o(1)$. The probability p for each random walk to start on a node of $V \setminus (V_2 \cup \widehat{z})$ is $\pi(V \setminus (V_2 \cup \widehat{z})) \geq 1/2$. For each of the random walks with label $1 \leq i \leq n^{\varepsilon}$ we define the indicator variable X_i to be one, if that random walk starts on $V \setminus (V_2 \cup \widehat{z})$. Let $X = \sum_{i=1}^{n^{\varepsilon}} X_i$. We have $\mathbb{E}[\,X\,] = n^{\varepsilon} \cdot \mathbb{E}[\,X_i\,] \geq n^{\varepsilon}/2$. Since the starting positions of the n^{ε} random walks are drawn independently, by a Chernoff bound

$$\mathbb{P}[\,\mathcal{E}_2\,] = \mathbb{P}\Big[\,X \leq \frac{1}{4} n^{\varepsilon}\,\Big] \leq \mathbb{P}[\,X \leq \mathbb{E}[\,X\,]/2\,] \leq e^{-n^{\varepsilon}/16} = o(1).$$

(iii) We now prove $\mathbb{P}\Big[\,\mathcal{E}_3 \mid \overline{\mathcal{E}_2}\,\Big] < 1/4$. From Lemma 8.11.$(ii)$ we get that w.p. at least $1/2$ a random walk starting at any node $u \in V_1$ does not leave G_1, *i. e.,* does not reach z^*, after $c_1 n$ time steps for some constant $c_1 > 0$. It is easy to see that the number of visits to $\widehat{z}$ required before the random walk hits G_2 instead of returning to G_1 is w.c.p. at least $\sqrt{\alpha'}/2$; this is because the fraction of edges from $\widehat{z}$ to G_2 is $\sqrt{n/\alpha'}/(\sqrt{n/\alpha'} + \sqrt{n})$. Using a Chernoff bound, we conclude that any random walk starting at G_1 doesn't hit G_2 during the first $T = c_1 \cdot \sqrt{\alpha' n}/2$ time steps with constant probability $p > 0$. Thus the probability that a random walk does not reach G_2 after $\lambda \cdot T$ time steps is at least p^{λ}, for any integer $\lambda \geq 1$. Setting $\lambda = \epsilon \cdot \log(1/p) \cdot \log(n/4)$, the probability that all of the at least $\frac{1}{4} n^{\varepsilon}$ random walks starting from G_1 reach G_2 within $\lambda \cdot T = \Omega(\sqrt{\alpha'} \cdot n \log n)$ steps is

$$\mathbb{P}\Big[\,\mathcal{E}_3 \mid \overline{\mathcal{E}_2}\,\Big] \leq (1 - p^{\lambda})^{\frac{1}{4} n^{\varepsilon}} \leq 1/e,$$

completing the proof.

□

The following lemma establishes a bound on the average hitting time.

Lemma 8.16. *Let G be the graph described at the beginning of Section 8.3.5. We have* $t_{\text{avg-hit}} = \Omega(n^{3/2})$

Proof. Consider a random walk that starts from an arbitrary vertex $u \in V$. By Lemma 8.11.(ii), every time a vertex z^i is visited, with probability at least $c > 0$ it takes $\Omega(n)$ time steps to visit another vertex z^j, $j \neq i$. Using a Chernoff bound, it follows that with probability larger than $1/2$ it takes at least $\Omega(n^{3/2})$ time steps to visit at least half of the nodes in $\{z^1, z^2, \ldots, z^\kappa\}$. By symmetry, it follows that for every vertex in a copy of G_1 there are $\Omega(n)$ vertices to which the hitting time is $\Omega(n^{3/2})$. Thus, by symmetry, $t_{\text{avg-hit}} = \sum_{u,v \in V} \pi(u) \cdot \pi(v) \cdot t_{\text{hit}}(u, v) = \Omega(n^2 \frac{1}{n^2} n^{3/2}) = \Omega(n^{3/2})$. □

Chapter 9

3-Majority [BCE+17]

In the 3-*Majority*, each node of a graph has an opinion, and in every round each node chooses independently three random neighbours and adopts the opinion of the majority where ties are broken arbitrarily. We are interested in the *consensus time*, which is the first point in time where all nodes have the same opinion. The system consists of n anonymous nodes connected by a complete graph. Initially, each node supports one opinion from the set $[k] \coloneqq \{1, \ldots, k\}$. We refer to these opinions as *colors*. The system state is modeled as a *configuration* vector $\mathbf{c}$, whose i-th component c_i denotes the number (support) of nodes with color i.

A consensus process is specified by an *update rule* that is executed by each node. The so-called Voter process (also known as Polling), uses the most naïve update rule: In every round, each node samples one neighbor independently and uniformly at random and adopts that node's color. Two further natural and prominent consensus processes are the 2-Choices and the 3-*Majority* process. Their corresponding update rules, as executed synchronously by each node, are as follows:

- 2-Choices: Sample two nodes independently and uniformly at random. If the samples have the same color, adopt it. Otherwise, ignore them and keep your current color.
- 3-Majority: Sample three nodes independently and uniformly at random. If a color is supported by at least two samples, adopt it. Otherwise, adopt the color of one of them at random[1].

One reason for the interest in these processes is that they represent simple and efficient self-stabilizing solutions for *Byzantine agreement* [PSL80, Rab83]: achieving consensus in the presence of an adversary that can disrupt a bounded set of nodes each round [BCN+14b, BCN+16, CER14, EFK+16]. Further interest stems from the fact that they capture aspects

[1] Equivalently, the node may adopt the color of a fixed sample (the first, or second, or third).

of how agreement is reached in social networks and biological systems [BDDS10, CER14, FPM+02].

At first glance, the above processes look quite different. But a slight reformulation of 3-Majority's update rule reveals an intriguing connection:

- 3-Majority (alt.): Sample two nodes independently and uniformly at random. If the samples have the same color, adopt it. Otherwise, sample a new neighbor and adopt its color.

This highlights the fact that 3-Majority is a combination of 2-Choices and Voter: Each node u performs the update rule of 2-Choices. If the sampled colors do not match, instead of keeping its color, u executes the update rule of Voter. Interestingly enough, both 3-Majority and 2-Choices behave identical in expectation[2]. In comparison to Voter, both 2-Choices and 3-Majority exhibit a drift: they favor colors with a large support, for which it is more likely that the first two samples match. In particular, if there is a certain initial bias[3] towards one color, Voter still needs linear time (in n) to reach consensus, while both 2-Choices and 3-Majority can exploit the bias to achieve sublinear time. On the other hand, it is unknown how 2-Choices and 3-Majority behave when they start from configurations having a large number of colors and no (or small) bias since for neither of the models reasonable bounds are unknown in the general setting with up to n colors.

9.1 Results

In this chapter, we give the first unconditional sublinear bound on any of these processes – an open issue from, e.g., [BCN+16]).

The following theorem states slightly simplified version of our upper bound (see Theorem 9.8.

Theorem 9.1 (Simplified)**.** *Starting from an arbitrary configuration,* 3*-Majority reaches consensus with high probability in* $O(n^{3/4}\log^{7/8} n)$ *rounds.*

The proof is more based on a combination of various techniques and results from different contexts. This approach not only results in a concise proof of the upper bound, but yields some additional, interesting results along the way. We give a brief overview of our approach in the next paragraph.

Should the bound of Ghaffari and Lengler [GL17] carry over from 2-Choices to 3-Majority, then together with our results (see Theorem 9.8 and Section 9.4), we would get that the expected consensus time is $O(k \log n)$ for any k.

[2]Simple calculations [BCN+14b, EFK+16] show that, for both processes, if x_i is the current fraction of nodes with color i then the expected fraction of nodes with color i after one round is $x_i^2 + (1 - \sum x_j^2) \cdot x_i$.

[3]The *bias* is the difference between the number of nodes supporting the most and second most common color.

9.2 Approach and Technical Contributions

To derive our upper bound on the time to consensus required by 3-Majority, we split the analysis in two phases: (a) the time needed to go from n to $\approx n^{1/4}$ colors and (b) the time needed to go from $\approx n^{1/4}$ to one color. The runtime of the second phase follows by a simple application of [BCN+16] and is $\tilde{O}(n^{3/4})$. Bounding the runtime of the first phase is more challenging: we cannot rely on the drift from a bias or similar effects, and it is not clear how to perform a direct analysis in this setting (3-Majority is geared towards biased configurations). To overcome this issue, we resort to a coupling between Voter and 3-Majority. Since the construction of such a coupling seems elusive, we use some machinery from majorization theory [MOA11] to merely prove the *existence* of the coupling (see next paragraph). As a consequence of (the existence of) this coupling, we get that the time needed by 3-Majority to reduce the number of colors to a fixed value is stochastically dominated by the time Voter needs for this (Proposition 9.10). This, finally, allows us to upper bound the time needed by 3-Majority[4] to go from $\approx n$ to $\approx n^{1/4}$ colors by the time Voter needs for this (which, in turn, we bound in Lemma 9.12 by $\tilde{O}(n/k)$).

The technically most interesting part of our analysis is the proof of the stochastic dominance between 3-Majority and Voter. It works for a wide class of processes (including Voter and 3-Majority), which we call *anonymous consensus* (AC-) processes (see Definition 9.2). These are defined by an update rule that causes each node to adopt any color i with the same probability α_i that depends only on the current frequency of colors.

In the following, we provide a natural way to compare two processes. First, we define a way to compare two configurations $\mathbf{c}$ and $\mathbf{c}'$. We use *vector majorization* for this purpose: $\mathbf{c}$ majorizes $\mathbf{c}'$ ($\mathbf{c} \succeq \mathbf{c}'$) if the total support of the j largest colors in $\mathbf{c}$ is not smaller than that in $\mathbf{c}'$ for all $j \in [k]$. In particular, note that a configuration where all nodes have the same color majorizes any other configuration. Intuitively, this can be thought of as a potential ϕ with $\phi(\mathbf{c}) \geq \phi(\mathbf{c}')$ if and only if $\mathbf{c} \succeq \mathbf{c}'$. Let us write $P(\mathbf{c})$ for the (random) configuration obtained by performing one step of a process P on configuration $\mathbf{c}$. Consider two processes P, P' and two configurations $\mathbf{c}, \mathbf{c}'$ with $\mathbf{c} \succeq \mathbf{c}'$. We say P *dominates* P' if, for all $j \in [k]$, the following holds:

> For every pair of color distributions $\mathbf{c}, \mathbf{c}'$ such that $\phi(\mathbf{c}') \leq \phi(\mathbf{c})$ after one-step P remains closer towards consensus than P', *i. e.,* $\mathbb{E}[\,\phi(P'(\mathbf{c}'))\,] \leq \mathbb{E}[\,\phi(P(\mathbf{c}))\,]$.

Note that this definition is not restricted to AC-processes.

Our main technical result (Theorem 9.4) proves that, for two AC-processes, P dominating P' implies that the time needed by P' to reduce the number of colors to a fixed value stochastically dominates the time P needs for this. Note that while this statement

[4] Note that for a large number of colors, a node executing 3-Majority behaves with high probability like a node performing Voter. Thus, it is relatively tight to bound 3-Majority by Voter in this parameter regime.

might sound obvious, it is not true in general (if one of the processes is not an AC-process): 2-Choices dominates Voter, but it is much slower in reducing the number of colors when there are many colors.

9.3 Consensus Model & Technical Framework

This section introduces our technical framework using concepts from majorization theory, which is used in Section 9.4 to derive the sublinear upper bound on 3-Majority. We provide a few definitions and state the main result of this section (Theorem 9.4).

9.3.1 Comparing Anonymous Consensus Processes

We first define a class of processes defined by update rules that depend only on the current configuration. The update rule states that each nodes adopts a color i with the same probability $\alpha_i(\mathbf{c})$, where $\mathbf{c} \in \mathcal{C}$ is the current configuration. In particular, node IDs (including the sampling node's ID) do not influence the outcome. In this sense, such update rules are *anonymous*.

Definition 9.2 (Anonymous Consensus Processes)**.** *Given a distributed system of n nodes, an* anonymous consensus process P_α *is characterized by a* process function $\alpha\colon \mathcal{C} \to [0,1]^n$ *with $\sum_{i\in[n]} \alpha_i(\mathbf{c}) = 1$ for all $c \in \mathcal{C}$. When in configuration $c \in \mathcal{C}$, each node independently adopts opinion $i \in [k]$ with probability $\alpha_i(\mathbf{c})$. We use the shorthand* AC-processes *to refer to this class.*

Given an AC-process P_α and a fixed initial configuration, let[5] $P_\alpha(t)$ denote the configuration of P_α at time t. By Definition 9.2, $(P_\alpha(t))_{t\geq 0}$ is a Markov chain, since $P_\alpha(t)$ depends only on $P_\alpha(t-1)$. Another immediate consequence of Definition 9.2 is that $P_\alpha(t)$ conditional on $P_\alpha(t-1) = \mathbf{c}$ is distributed according to $\mathsf{Mult}(n, \alpha(\mathbf{c}))$; in other words, the 1-step distribution of an AC-process is a multinomial distribution. Two important examples of AC-processes include Voter and 3-Majority:

- In the Voter process $P_{\alpha^{(\mathcal{V})}}$, each node samples one node (according to the pull mechanism) and (always) adopts that node's opinion. Thus

$$\alpha_i^{(\mathcal{V})}(\mathbf{c}) = \frac{c_i}{n}. \tag{9.1}$$

- In the 3-Majority process $P_{\alpha^{(3M)}}$, each node samples independently and uniformly at random three nodes. If a color is supported by at least two of the samples,

[5] Recall that, with a slight abuse of notation we also write $P(\mathbf{c})$ for the (random) configuration obtained by performing one step of a process P on configuration $\mathbf{c}$.

adopt it. Otherwise, adopt a random one of the sampled colors. Simple calculations (see [BCN+14b]) show

$$\alpha_i^{(3M)}(\mathbf{c}) = \frac{c_i}{n} \cdot \left(1 + \frac{c_i}{n} - \left\|\frac{c}{n}\right\|_2^2\right). \tag{9.2}$$

For any protocol P starting with configuration $\mathbf{c} \in \mathcal{C}$ let $T_P^\kappa(\mathbf{c})$ denote the first time step where the number of remaining colors reduces to κ where $\kappa \in \mathbb{N}$. The next definition introduces dominance between protocol. Intuitively, a protocol P dominates another protocol P' if their expected behavior preserves majorization.

Definition 9.3 (Protocol Dominance)**.** *Consider two (not necessarily* AC*-) processes $P, \tilde{P}$. We say P dominates $\tilde{P}$ if for all $\mathbf{c}, \tilde{\mathbf{c}} \in \mathcal{C}$ with $\mathbf{c} \succeq \tilde{\mathbf{c}}$ we have that $\mathbb{E}[\,P(\mathbf{c})\,] \succeq \mathbb{E}\Big[\,\tilde{P}(\tilde{\mathbf{c}})\,\Big]$ holds.*

Note that, in the case of AC-protocols, Definition 9.3 can be stated as follows: P_α dominates $P_{\tilde{\alpha}}$ if and only if $\mathbf{c} \succeq \tilde{\mathbf{c}} \Rightarrow \alpha(\mathbf{c}) \succeq \tilde{\alpha}(\tilde{\mathbf{c}})$ for all $\mathbf{c}, \tilde{\mathbf{c}} \in \mathcal{C}$ with $\mathbf{c} \succeq \tilde{\mathbf{c}}$. With this, the main result of our framework can be stated as follows.

Theorem 9.4. *Consider two* AC*-Processes P and P' where P dominates P'. Assume P and P' are started from the same configuration $\mathbf{c} \in \mathcal{C}$. Then, for any $\kappa \in \mathbb{N}$, the time needed by P' to reduce the number of remaining colors to κ dominates the time P needs for this, i.e.,*

$$T_{P'}^\kappa(\mathbf{c}) \geq^{st} T_P^\kappa(\mathbf{c}).$$

One should note that the statement of Theorem 9.4 is not true in general (i.e., for non-AC-processes). In particular, 2-Choices dominates Voter, but our upper bound on Voter (Lemma 9.12) and our lower bound on 2-Choices (Theorem 10.5 in Chapter 10) contradict the statement of Theorem 9.4: The 2-Choices process takes $\Omega(n/\log n)$ time steps to reduce to $k = n/\log n$ opinions whereas Voter reduces to k w.h.p. in only $O(n/k \log n) = O(\log^2(n))$ time steps.

9.3.2 Coupling two AC-Processes - Proof of Theorem 9.4

In order to prove Theorem 9.4, we formulate a strong 1-step coupling property for AC-processes:

Lemma 9.5 (1-Step Coupling)**.** *Let P_α and $P_{\tilde{\alpha}}$ be two* AC*-processes. Consider any two configurations $\mathbf{c}, \tilde{\mathbf{c}} \in \mathcal{C}$ with $\alpha(\mathbf{c}) \succeq \tilde{\alpha}(\tilde{\mathbf{c}})$. Let $\mathbf{c}'$ and $\tilde{\mathbf{c}}'$ be the configurations of P_α and $P_{\tilde{\alpha}}$ after one round, respectively. Then, there exists a coupling such that $\mathbf{c}' \succeq \tilde{\mathbf{c}}'$.*

Proof. Consider the processes P_α and $P_{\tilde{\alpha}}$ with the configurations $\mathbf{c}$ and $\tilde{\mathbf{c}}$ from the theorem's statement. Let $\mathbf{Y} = \mathbf{c}'$ and $\mathbf{X} = \tilde{\mathbf{c}}'$ denote the configurations resulting after one round of

P_α on c and $P_{\tilde{\alpha}}$ on $\tilde{\mathbf{c}}$, respectively. Let $\mathbf{\Theta} \coloneqq \alpha(\mathbf{c})$ and $\tilde{\Theta} \coloneqq \tilde{\alpha}(\tilde{\mathbf{c}})$. As observed earlier in Section 9.3.1, we have $\mathbf{Y} \sim \mathsf{Mult}(n, \Theta_1)$ and $\mathbf{X} \sim \mathsf{Mult}(n, \Theta_2)$. By the theorem's assumption, we have $\mathbf{\Theta} \succeq \tilde{\mathbf{\Theta}}$. Since, by Proposition A.20, the function $\mathbf{\Theta} \to \mathbb{E}[\, f(\mathsf{Mult}(n, \Theta))\,]$ is Schur-convex for any Schur-convex function $f(\cdot)$ for which the expectation exists, we get $\mathbf{X} \preceq^{\text{st}} \mathbf{Y}$.

Since the configuration space $\mathcal{C}$ is a finite subset of $\mathbb{R}^n$, it is closed and so is $\{\, (x, y) \mid x \preceq y \,\}$. We now apply Theorem 9.7 (Strassen's Theorem) to get that there exists a coupling between $\mathbf{X}$ and $\mathbf{Y}$ such that[6] $\mathbf{X} \preceq \mathbf{Y}$. This finishes the proof. □

Note that Theorem 9.4 is an immediate consequence of Lemma 9.5: Since P dominates P' (which is, for AC-processes, equivalent to $\alpha(\mathbf{c}) \succeq \tilde{\alpha}(\tilde{\mathbf{c}})$ for all $\mathbf{c}, \tilde{\mathbf{c}}$ with $\mathbf{c} \succeq \tilde{\mathbf{c}}$) we can apply Lemma 9.5 iteratively to get Theorem 9.4. The fine-grained comparison enabled by Lemma 9.5 is based on three observations:

1. The (pre-) order "$\preceq$" on the set of configurations naturally measures the closeness to consensus. Indeed, a configuration with only one remaining color is maximal with respect to "$\preceq$". Similarly, the n-color configuration is minimal.

2. We can define a vector variant "$\preceq^{\text{st}}$" of stochastic domination (see Definition 9.6) such that $\mathbf{\Theta}_1 \preceq \mathbf{\Theta}_2 \Rightarrow \mathsf{Mult}(m, \mathbf{\Theta}_1) \preceq^{\text{st}} \mathsf{Mult}(m, \mathbf{\Theta}_2)$ ([MOA11, Proposition 11.E.11] or Proposition A.20).

3. Consider two configurations $\mathbf{c}, \tilde{\mathbf{c}} \in \mathcal{C}$ with $\alpha(\mathbf{c}) \succeq \tilde{\alpha}(\tilde{\mathbf{c}})$. Since $P_\alpha(\mathbf{c}) \sim \mathsf{Mult}(n, \alpha(\mathbf{c}))$ and $P_{\tilde{\alpha}} \sim \mathsf{Mult}(n, \tilde{\alpha}(\tilde{\mathbf{c}}))$, the previous observations imply that one step of P_α on c is stochastically "better" than one step of $P_{\tilde{\alpha}}$ on $\tilde{\mathbf{c}}$. Our goal is to apply Lemma 9.5 iteratively to get Theorem 9.4. For this, we prove a coupling showing majorization between the resulting configurations. We achieve this via a variant of Strassen's Theorem (see Theorem 9.7 below), which translates *stochastic* domination among random vectors to the *existence* of such a coupling.

We now give a definition of stochastic majorization that is compatible with the preorder "$\preceq$" on the configuration space $\mathcal{C}$ (cf. [MOA11, Chapter 11]).

Definition 9.6 (Stochastic Majorization)**.** *For two random vectors* $\mathbf{X}$ *and* $\mathbf{Y}$ *in* $\mathbb{R}^d$*, we write* $\mathbf{X} \preceq^{st} \mathbf{Y}$ *and say that* $\mathbf{Y}$ stochastically majorizes $\mathbf{X}$ *if* $\mathbb{E}[\, f(\mathbf{X})\,] \leq \mathbb{E}[\, f(\mathbf{Y})\,]$ *for all Schur-convex functions* f *on* $\mathbb{R}^d$ *such that the expectations are defined.*

We proceed by stating the aforementioned variant (Theorem 9.7) of Strassen's Theorem (Theorem A.21).

Theorem 9.7 (Strassen's Theorem (variant))**.** *Consider a closed subset* $\mathcal{A} \subseteq \mathbb{R}^n$ *such that the set* $\{\, (x, y) \mid x \preceq y \,\}$ *is closed. For two random vectors* X *and* Y *over* $\mathcal{A}$*, the following conditions are equivalent:*

[6] Observe that Strassen's Theorem gives us that $\mathbb{P}[\, \mathbf{X} \preceq \mathbf{Y}\,] = 1$. That is, $\mathbf{X} \preceq \mathbf{Y}$ holds *almost surely*. However, since $\mathcal{C}$ is finite, this actually means that $\tilde{\mathbf{c}}' = \mathbf{X} \preceq \mathbf{Y} = \mathbf{c}'$ holds (surely).

(i) (Stochastic Majorization) $X \leq^{st} Y$ *and*

(ii) (Coupling) there is a coupling between X *and* Y *such that* $\mathbb{P}[\, X \preceq Y \,] = 1$.

Proof. Consider the cone

$$\mathcal{C} := \{\, f \colon \mathcal{A} \to \mathbb{R} \mid f \text{ is Schur-convex} \,\}$$

of real-valued Schur-convex functions on $\mathcal{A}$. This cone implies a preorder "$\leq_{\mathcal{C}}$" on $\mathcal{A}$ by the definition $x \leq_{\mathcal{C}} y :\Leftrightarrow f(x) \leq f(y)$ for all $f \in \mathcal{C}$. One can show that this preorder is the vector majorization "$\preceq$" (cf. [MOA11, Example 14.E.5])[7]. Now, "$\leq_{\mathcal{C}}$" being equal to "$\preceq$" has two implications:

(a) The stochastic majorization "$\leq^{\text{st}}_{\mathcal{C}}$" implied by the preorder "$\leq_{\mathcal{C}}$" is the stochastic majorization "$\leq^{\text{st}}$" from Definition 9.6 (cf. [MOA11, Definition 17.B.1]).

(b) Since a cone $\mathcal{C}$ is complete if it is maximal with respect to functions preserving the preorder "$\leq_{\mathcal{C}}$" (cf. [MOA11, Definition 14.E.2]), $\mathcal{C}$ is complete (Schur-convex functions are by definition the set of all functions preserving the majorization preorder).

From (a) we get that Condition (i) is actually Condition (i) of Theorem A.21. The same holds for Condition (ii). From (b) we get that $\mathcal{C} = \mathcal{C}^* = \mathcal{C}^+$ (cf. [MOA11, Proposition 17.B.3]), such that Conditions (i) and (ii) are equivalent by Theorem A.21. This finishes the proof. □

With this, Lemma 9.5 follows by a straightforward combination of the aforementioned machinery.

9.4 Upper Bound for 3-Majority - Proof of Theorem 9.8

In this section, we provide a sublinear upper bound on the time needed by 3-Majority to reach consensus with high probability. This is one of our main results and is formulated in the following theorem.

Theorem 9.8. *Starting from* any *configuration* $c \in \mathcal{C}$*, 3-Majority reaches consensus w.h.p. in* $O(n^{3/4} \log^{7/8} n)$ *rounds.*

The analysis is split into two phases, each consisting of $O(n^{3/4} \log^{7/8} n)$ rounds.

Phase 1: From up to n to $n^{1/4} \log^{1/8}$ colors. This is the crucial part of the analysis. Instead of analyzing 3-Majority directly, we use our machinery from Section 9.3.1 to show that 3-Majority is not slower than Voter (Proposition 9.10). Then, we prove that Voter reaches $O(n^{1/4})$ colors in $O(n^{3/4} \log^{7/8} n)$ rounds (Lemma 9.12).

[7] Alternatively, one checks this manually: The direction $x \preceq y \Rightarrow x \leq_{\mathcal{C}} y$ is trivial by the definition of Schur-convexity. For $x \leq_{\mathcal{C}} y \Rightarrow x \preceq y$ consider the $n+1$ Schur-convex functions $z \mapsto \sum_{j \in [i]} \mathbf{z}^{\downarrow}$ for $i \in [n]$ and $z \mapsto -\|z\|_1$.

Phase 2: From up to $n^{1/4}\log^{1/8} n$ to 1 color (consensus). Once we reached a configuration with $n^{1/4}\log^{1/8} n$ colors, we can apply [BCN+16, Theorem 3.1] (see Theorem 9.9), a previous analysis of 3-Majority. It works only for initial configurations with at most $k \leq n^{1/3-\epsilon}$ colors ($\epsilon > 0$ arbitrarily small). In that case, [BCN+16, Theorem 3.1] yields a runtime of $O((k^2 \log^{1/2} n + k \log n) \cdot (k + \log n))$. Since the first phase leaves us with $O(n^{1/4})$ colors, this immediately implies that the second phase takes $O(n^{3/4} \log^{7/8} n)$ rounds.

This section proceeds by proving the runtime of Phase 1 in two steps: dominating the runtime of 3-Majority by that of Voter (Section 9.4.1) and proving the corresponding runtime for Voter (Section 9.4.2). In the end, we can combine these results together with [BCN+16, Theorem 3.1] to prove Theorem 9.8.

Proof of Theorem 9.8. Consider any initial configuration $\mathbf{c} \in \mathcal{C}$. By applying Lemma 9.12 for $k = n^{1/4}$, we get that Voter reduces the number of remaining colors w.h.p. from initially at most n to $n^{1/4}$ in $O(n^{3/4} \log^{7/8} n)$ rounds. By Proposition 9.10, the time it takes 3-Majority to reach some fixed number of remaining colors is dominated by the time it takes Voter to reach the same number of remaining colors. In particular, we get that 3-Majority also reduces the number of remaining colors w.h.p. to $n^{1/4}$ in $O(n^{3/4} \log^{7/8} n)$ rounds. That is, the first phase takes $O(n^{3/4} \log^{7/8} n)$ rounds.

For the second phase, we apply [BCN+16, Theorem 3.1] (see Theorem 9.9) for $k = n^{1/4} = o\left(n^{1/3}\right)$. This immediately yields that the second phase takes $O(n^{3/4} \log^{7/8} n)$ rounds, finishing the proof. □

Theorem 9.9 ([BCN+16, Theorem 3.1])**.** *Let $\varepsilon > 0$ be an arbitrarily small constant. Starting from any initial configuration with $k \leq n^{1/3-\varepsilon}$ colors, 3-Majority reaches consensus w.h.p. in*

$$O\left((k^2 \log^{1/2} n + k \log n) \cdot (k + \log n)\right)$$

rounds.

9.4.1 Analysis of Phase 1: 3-Majority vs. Voter

We prove the following proposition.

Proposition 9.10. *Consider the clique. We have tht Voter ($\mathcal{V}$) and 3-Majority ($3M$) started from the same initial configuration $\mathbf{c} \in \mathcal{C}$. There is a coupling such that after any round, the number of remaining colors in Voter is not smaller than those in 3-Majority. In particular, the time Voter needs to reach consensus stochastically dominates the time needed by 3-Majority to reach consensus, i.e.,*

$$T^{\kappa}_{3M}(\mathbf{c}) \leq^{st} T^{\kappa}_{\mathcal{V}}(\mathbf{c}).$$

Proof. By Theorem 9.4, all we have to prove is $\mathbf{c} \succeq \tilde{\mathbf{c}} \Rightarrow \alpha^{(3M)}\mathbf{c} \succeq \alpha^{(\mathcal{V})}(\tilde{\mathbf{c}})$ (see Section 9.3.1). To this end, consider two configurations $\mathbf{c}, \tilde{\mathbf{c}} \in \mathcal{C}$ with $\mathbf{c} \succeq \tilde{\mathbf{c}}$. Let $\mathbf{p} \coloneqq \alpha^{(3M)}\mathbf{c}$ and $\tilde{\mathbf{p}} \coloneqq \alpha^{(\mathcal{V})}(\tilde{\mathbf{c}})$. We have to show $\mathbf{p} \succeq \tilde{\mathbf{p}}$. Since these are probability vectors, we have $\|\mathbf{p}\|_1 = 1 = \|\tilde{\mathbf{p}}\|_1$. It remains to consider the partial sums for $k \in [n]$. For this, let $\mathbf{x} \coloneqq \mathbf{c}/n$ and $\tilde{\mathbf{x}} \coloneqq \tilde{\mathbf{c}}/n$. Remember that $p_i = x_i^2 + (1 - \|x\|_2^2) \cdot x_i$ ((9.2)) and $\tilde{p}_i = \tilde{x}_i$ ((9.1)). In the following, we assume (w.l.o.g.) $\mathbf{p} = \mathbf{p}^{\downarrow}$ and $\tilde{\mathbf{p}} = \tilde{\mathbf{p}}^{\downarrow}$ (this implies $\mathbf{x} = \mathbf{x}^{\downarrow}$ and $\tilde{\mathbf{x}} = \tilde{\mathbf{x}}^{\downarrow}$). We compute

$$\begin{aligned} \sum_{i=1}^{k} p_i - \sum_{i=1}^{k} \tilde{p}_i &= \sum_{i=1}^{k} x_i^2 + \sum_{i=1}^{k} x_i - \|\mathbf{x}\|_2^2 \sum_{i=1}^{k} x_i - \sum_{i=1}^{k} \tilde{x}_i \\ &\geq \sum_{i=1}^{k} x_i^2 - \|\mathbf{x}\|_2^2 \sum_{i=1}^{k} x_i. \end{aligned} \tag{9.3}$$

We have to show that this last expression is non-negative, which is equivalent to

$$\|\mathbf{x}\|_2^2 \leq \left(\sum_{i=1}^{k} x_i^2\right) / \left(\sum_{i=1}^{k} x_i\right). \tag{9.4}$$

This holds trivially for $k = n$ (where we have equality). Thus, it is sufficient to show that $(\sum_{i=1}^{k} x_i^2)/(\sum_{i=1}^{k} x_i)$ is non-increasing in k. That is, for any $k \in [n-1]$ we seek to show the inequality

$$\frac{\sum_{i=1}^{k+1} x_i^2}{\sum_{i=1}^{k+1} x_i} = \frac{\sum_{i=1}^{k} x_i^2 + x_{k+1}^2}{\sum_{i=1}^{k} x_i + x_{k+1}} \leq \frac{\sum_{i=1}^{k} x_i^2}{\sum_{i=1}^{k} x_i}. \tag{9.5}$$

This inequality is of the form $\frac{A+x}{B+x} \leq \frac{A}{B}$, where $A, B, x > 0$. Rearranging shows that this is equivalent to $x \leq A/B$. Thus, (9.5) holds if and only if $x_{k+1} \leq (\sum_{i=1}^{k} x_i^2)/(\sum_{i=1}^{k} x_i)$. This last inequality holds via $x_{k+1} \cdot \sum_{i=1}^{k} x_i = \sum_{i=1}^{k} x_i \cdot x_{k+1} \leq \sum_{i=1}^{k} x_i \cdot x_i = \sum_{i=1}^{k} x_i^2$, where we used $\mathbf{x} = \mathbf{x}^{\downarrow}$. This finishes the proof. □

9.4.2 Analysis of Phase 1: A Bound for Voter

We analyze the time the Voter process takes to reduce the number of remaining colors from n to k. One should note that [BGKM16] studies a similar process. However, their analysis relies critically on the fact that their process is lazy (i.e., nodes do not sample another node with probability 1/2), while our proof does not require any laziness.

We make use of the well-known duality (via time reversal) between the Voter process and *coalescing random walks*. In the coalescing random walks process there are initially n independent random walks, one placed at each of the n nodes. While performing synchronous steps, whenever two or more random walks meet, they coalesce into a single random walk. Let T_C^k denote the number of steps it takes to reduce the number of random walks from n to k in the coalescing random walks process (the *coalescence time*). Similarly, let $T_{\mathcal{V}}^k$ denote the number of rounds it takes Voter to reduce the number of remaining colors from n to k.

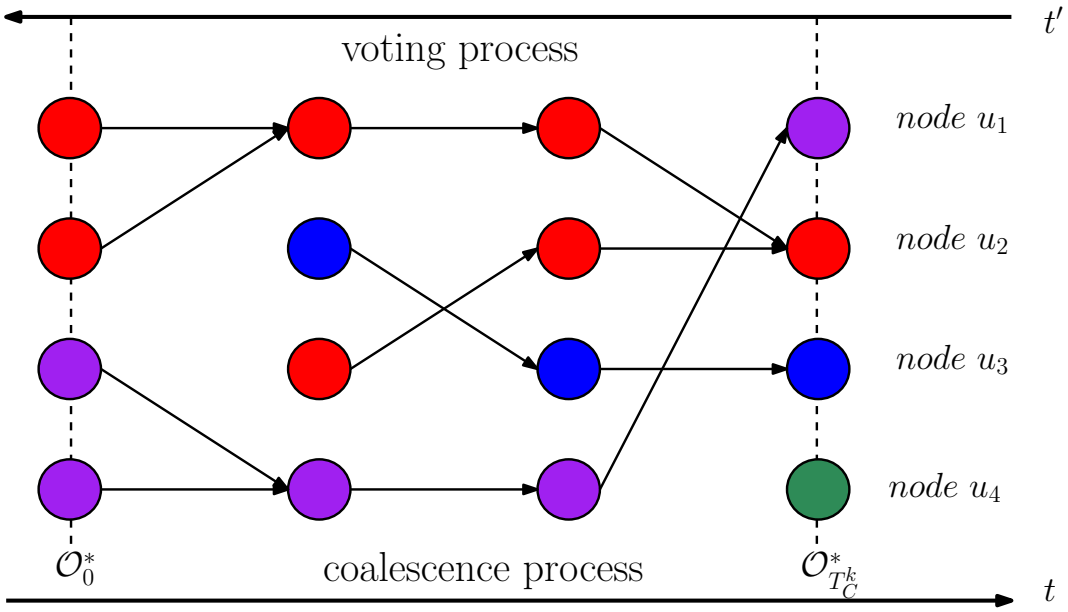


Figure 9.1: Running the coalescence process from right to left (an edge from u to v means that the token on u -if any- moves to v) yields that after $T = 4$ rounds the number of random walks reduces to $k = 2$. Using the same random choices (black arrows) for the voter process and running the process from left to right (an edge from u to v means that u pulls v's opinion) we derive that the number of opinions after $T = 4$ rounds is also 2. This is no coincidence as we show in Proposition 9.11.

The following lemma uses the high-level idea of the proof presented in [AF02, Chapter 14] which only considers the case $k = 1$. For the purposes of our proof we would only require a coupling with $T_{\mathcal{V}}^k \leq T_C^k$, but for the sake of completeness we show the stronger claim $T_{\mathcal{V}}^k = T_C^k$.

Proposition 9.11. *For any graph $G = (V, E)$, there exists a coupling such that $T_C^k = T_{\mathcal{V}}^k$.*

Proof. For $t \in \mathbb{N}$ and for $u \in V$ define the random variables $Y_t(u)$ with $Y_t(u) \sim \text{uniform}(N(u))$, where uniform$(\cdot)$ denotes the uniform distribution and $N(u)$ denotes the neighborhood of u. Hence, $Y_t(u) = v$ means that u pulls information from node v in step t. In the Coalescence process, the random variable $Y_t(u) \in N(u)$, $t \in [0, T_C^k)$ captures the transition performed by the random walk which is at u at time t (if any). In other words, these random variables define the arrows in Figure 9.1. For the voter process $Y_t(u) = v$ means that in step t node u adopts the opinion of v.

Let $X(u) = (X_0(u) = u, X_1(u), \ldots, X_{T_C^k}(u))$ be the trajectory of the random walk starting at u. We can thus express

$$X_t(u) = \begin{cases} u & \text{if } t = 0 \\ Y_{t-1}(X_{t-1}(u)) & \text{otherwise.} \end{cases} \tag{9.6}$$

Thus, this trajectory $X(u)$ and the random variable T_C^k are completely determined by the random variables $\mathcal{Y} = \{Y_t(u) \colon t \in \mathbb{N}, u \in V\}$.

Let $\mathcal{V}_{T_C^k}$ be the Voter process whose starting time $t' = 0$ equals the time T_C^k of the coalescence process (see also Figure 9.1). Let $\mathcal{O}^*_{T_C^k - t'}(u)$ be the opinion of u at time t' of $\mathcal{V}_{T_C^k}$. For every node $u \in V$ and $t' \in [0, T_C^k]$ we can thus express

$$\mathcal{O}^*_{T_C^k - t'}(u) = \begin{cases} u & \text{if } t' = 0 \\ \mathcal{O}^*_{T_C^k - (t'-1)}(Y_{T_C^k - t'}(u)) & \text{otherwise.} \end{cases} \tag{9.7}$$

Note that (9.7) constructs a coupling between the Voter process and the coalescence process through the common usage of the random variables $\mathcal{Y}$ in (9.6) and (9.7). In particular, by unrolling (9.6) and (9.7) we get

$$\begin{aligned} X_{T_C^k}(u) &= Y_{T_C^k - 1}(Y_{T_C^k - 2}(\dots(Y_0(X_0(u)))\dots)) \\ &\overset{(a)}{=} Y_{T_C^k - 1}(Y_{T_C^k - 2}(\dots(Y_0(u))\dots)) \\ O^*_0(u) &= O^*_{T_C^k}(Y_{T_C^k - 1}(Y_{T_C^k - 2}(\dots(Y_0(u))\dots))) \\ &\overset{(b)}{=} Y_{T_C^k - 1}(Y_{T_C^k - 2}(\dots(Y_0(u)\dots)), \end{aligned}$$

where and (a) we used that $X_0(u) = u$ and in (b) we used that $O^*_{T_C^k}(v) = v$ for all v. The above equations imply

$$X_{T_C^k}(u) = \mathcal{O}^*_0(u). \tag{9.8}$$

Let $Z_t = \{X_t(u) \colon u \in V\}$ denote the positions of the remaining walks in the coalescence process at time t. Observe that $|Z_0| = n$, $|Z_{T_C^k}| \leq k$, by definition of T_C^k. We have, by (9.8), that

$$Z_{T_C^k} = \{X_{T_C^k}(u) \colon u \in V\} = \{\mathcal{O}^*_0(u) \colon u \in V\} =: \mathcal{O}^*_0. \tag{9.9}$$

From (9.9) we infer $|\mathcal{O}^*_0| = |Z_{T_C^k}| \leq k$, which implies that

$$T_{\mathcal{V}}^k \leq T_C^k.$$

In the reminder we generalize the previous coupling to show that

$$T_{\mathcal{V}}^k = T_C^k.$$

In particular, we consider the Voter process for all starting position $\tau < T_C^k$ (all nodes have different colors at round t) and show that the resulting number of opinions is strictly more than k.

Let $\mathcal{V}_\tau$ be the Voter process that starts at time $\tau \in [0, T_C^k)$, and let $\mathcal{O}^\tau_{T_C^k - t'}(u)$ be the opinion of u at time t' of $\mathcal{V}_\tau$. For every node $u \in V$ and $t' \in [0, \tau]$ we have

$$\mathcal{O}^\tau_{\tau - t'}(u) = \begin{cases} u & \text{if } t' = \tau \\ \mathcal{O}^\tau_{\tau-(t'-1)}(Y_{\tau - t'}(u)) & \text{otherwise.} \end{cases} \tag{9.10}$$

Similarly as before, by unrolling (9.6) and (9.10) we get

$$\begin{aligned} X_\tau(u) &= Y_{\tau-1}(Y_{\tau-2}(\dots(Y_0(X_0(u)))\dots)) \\ &\overset{(a)}{=} Y_{\tau-1}(Y_{\tau-2}(\dots(Y_0(u))\dots)) \\ O^\tau_0(u) &= O^\tau_\tau(Y_{\tau-1}(Y_{\tau-2}(\dots(Y_0(u))\dots))) \\ &\overset{(b)}{=} Y_{\tau-1}(Y_{\tau-2}(\dots(Y_0(u)\dots)), \end{aligned}$$

where and (a) we used that $X_0(u) = u$ and in (b) we used that $O^\tau_\tau(v) = v$ for all v. By defining $\mathcal{O}^0_{T_C^k - t'} = \{\mathcal{O}^0_{T_C^k - t'}(u) \colon u \in V\}$, from the above equations we get that

$$X_\tau(u) = \mathcal{O}^\tau_0(u).$$

Hence,

$$Z_\tau = \{X_\tau(u) \colon u \in V\} = \{\mathcal{O}^\tau_0(u) \colon u \in V\} =: \mathcal{O}^\tau_0. \tag{9.11}$$

Since for all $\tau < T_C^k$ we have $|Z_\tau| > k$, from (9.11) it follows that $|\mathcal{O}^\tau_0| = |Z_\tau| > k$ which yields the claim. □

Given the above duality, we are ready to prove the lower bound on Voter.

Lemma 9.12. *Consider an arbitrary initial configuration* $\mathbf{c} \in \mathcal{C}$. *Voter reaches a configuration* $\mathbf{c}'$ *having at most* k *remaining colors w.h.p. in* $O(\frac{n}{k} \log n)$ *rounds, i.e.,* $\mathbb{P}\Big[\, T_{\mathcal{V}}^k = O(\frac{n}{k} \log n) \,\Big] \geq 1 - 1/n$.

Proof. We prove the lemma using the well-known duality (via time reversal) between the Voter process and *coalescing random walks*.

It is well-known (e.g., [AF02]), that $T_{\mathcal{V}}^1 = T_C^1$. This statement generalizes for all $k \in [n]$ (see Proposition 9.11 for a proof) to

$$T_{\mathcal{V}}^k = T_C^k. \tag{9.12}$$

Thanks to the previous identity, we can prove the lemma's statement by proving that w.h.p. $T_C^k = O(\frac{n}{k} \log n)$. To this end, we show that $\mathbb{E}\Big[\, T_C^k \,\Big] = O(n/k)$. In order to get the claimed bound in concentration, we can apply the following standard argument. Consider

the process as a sequence of phases, each one of length $2\mathbb{E}\left[T_C^k\right]$. We say that a phase is successful when the number of remaining random walks drops below n/k. Thanks to our bound in expectation above and the Markov inequality, we easily get that every phase has probability $\Omega(1)$ to be successful. So, with high probability, there will be at least one success within the first $O(\log n)$ phases.

Let X_t denote the number of coalescing random walks at time t. We have $X_0 = n$ and $T_C^k = \min\{\, t \geq 0 \mid X_t \leq k \,\}$. We seek to apply drift theory (Theorem A.13 to derive a bound on $\mathbb{E}\left[T_C^k\right]$. Next, we compute an upper bound on $\mathbb{E}[\, X_{t+1} - X_t \mid X_t = x \,]$.

Let us begin assuming that k is any constant. It holds in general that

$$\mathbb{E}[\, X_{t+1} - X_t \mid X_t \geq 2 \,] \leq -1/n,$$

since in expectation two random walks collide w.p. $1/n$ in a given time step. Hence we can directly apply[8] Theorem A.13 with parameters $h(x) = 1/n$ to reduce from k random walks to 1, yielding the bound $\mathbb{E}\left[T_C^k\right] = O(n/k) = O(n)$, where in the latter equality we used that k is constant.

We now consider the case where k is larger than a big constant, say $k > 100$. Assume that in every time step the random walks move in two phases. Let W_1 denote an arbitrary set of $\lfloor X_t/2 \rfloor$ random walks and let W_2 denote the remaining ones. We first look at how the random walks in W_1 coalesce, then we consider the movement of the remaining walks W_2. Let $\mathcal{E}$ be the event that the walks in W_1 move onto more than $\lfloor X_t/4 \rfloor$ distinct nodes. This would imply that each walk in W_2 coalesces with one in W_1 with probability at least $\lfloor X_t/4 \rfloor/n$. We thus have

$$\mathbb{E}[\, X_{t+1} \mid X_t = x, \mathcal{E} \,] \leq x - \lceil x/2 \rceil \cdot \frac{\lfloor x/4 \rfloor}{n} \leq x - \frac{x^2}{10n}.$$

Moreover, conditioning on $\overline{\mathcal{E}}$ implies that there were at least $\lfloor X_t/2 \rfloor - \lfloor X_t/4 \rfloor$ collisions during the first phase. Thus,

$$\mathbb{E}\left[\, X_{t+1} \mid X_t = x, \overline{\mathcal{E}} \,\right] \leq x - (\lfloor x/2 \rfloor - \lfloor x/4 \rfloor) \leq x - \frac{x^2}{10n}.$$

Hence, by law of total expectation,

$$\begin{aligned}\mathbb{E}[\, X_{t+1} \mid X_t = x \,] = \mathbb{E}[\, X_{t+1} \mid X_t = x, \mathcal{E} \,]\mathbb{P}[\, \mathcal{E} \,] \\ + \mathbb{E}\left[\, X_{t+1} \mid X_t = x, \overline{\mathcal{E}} \,\right]\mathbb{P}\left[\, \overline{\mathcal{E}} \,\right] \leq x - \frac{x^2}{10n}.\end{aligned}$$

[8] Technically, one would have to define a new random variable which is 0 whenever the number of random walks reduces to 1. We illustrate this technicality shortly, for case $k > 100$ below.

In order to apply Theorem A.13, we define the random variables $(Y_t)_{t\geq 0}$ as follows

$$Y_t = \begin{cases} X_t & \text{if } X_t > k, \\ 0 & \text{otherwise.} \end{cases}$$

Let $T^* = \{t \geq 0 \mid Y_t = 0\}$. Since by construction we have $Y_t = X_t$ for $t < T_C^k$ and $Y_{T_C^k} = 0$ otherwise, it follows that

$$T_C^k = T^*. \tag{9.13}$$

Therefore,

$$\mathbb{E}[\, Y_{t+1} \mid Y_t = y, Y_t > k \,] \leq y - \frac{y^2}{10n}.$$

We can thus apply Theorem A.13 for the random variables $(Y_t)_{t\geq 0}$ with $x_{\min} = k, x_{\max} = n$, and $h(x) = \frac{x^2}{10n}$, obtaining

$$\mathbb{E}[\, T^* \,] \leq \frac{k}{k^2/(10n)} + \int_k^n \frac{1}{h(u)} \leq \frac{10n}{k} + 10n\left(-\frac{1}{n} - \left(-\frac{1}{k}\right)\right) \leq 20\frac{n}{k}. \tag{9.14}$$

Finally, from (9.12), (9.13) and (9.14) we get

$$\mathbb{E}\left[\, T_{\mathcal{V}}^k \,\right] = \mathbb{E}\left[\, T_C^k \,\right] = \mathbb{E}[\, T^* \,] \leq 20\frac{n}{k}, \tag{9.15}$$

concluding the proof. □ □

9.5 Limitations of 1-Step Coupling

In this section we show that there are configurations $\mathbf{c} \preceq \tilde{\mathbf{c}}$ such that $\alpha^{(hM)}(\mathbf{c}) \not\preceq \alpha^{(h+1M)}(\tilde{\mathbf{c}})$. This means that, Lemma 9.5 is not strong enough to derive Conjecture 13.1. Consider the configurations $\mathbf{x} := (1/2, 1/6, 1/6, 1/6) \preceq (1/2, 1/2, 0, 0) =: \tilde{\mathbf{x}}$ (for simplicity, we use the fraction vectors $\mathbf{x} = \mathbf{c}/n$). For symmetry reasons, we immediately get that $\alpha^{(h+1M)}(\tilde{\mathbf{c}}) = (1/2, 1/2, 0, 0) = \tilde{\mathbf{c}}$. However, even for $h = 3$, for the second configuration we get that the expected fraction of the nodes which adopt the first opinion after one step is

$$1 \cdot \binom{3}{0} \cdot \left(\frac{1}{2}\right)^3 + 1 \cdot \binom{3}{1} \cdot \left(\frac{1}{2}\right)^2 \cdot \frac{3}{6} + \frac{1}{3} \cdot \binom{3}{2} \cdot \frac{1}{2} \cdot \frac{3}{6} \cdot \frac{2}{6} = \frac{7}{12}. \tag{9.16}$$

The three terms of the sum on the left hand side correspond to the cases and probabilities for which the first color is adopted:

- all samples choose color 1 (probability to win is 1, number of cases $\binom{3}{0}$),
- two samples choose color 1 (probability to win is 1, number of cases $\binom{3}{1}$), or

- 1 sample chooses color 1 and the other samples choose different colors (probability to win is $1/3$, number of cases $\binom{3}{2}$).

Thus, for n large enough, with high probability the configuration resulting from $(h+1)$-Majority will not majorize the one resulting from $\mathrm{h}{-}Majority$.

Chapter 10

2-Choices [EFK+16, BCE+17]

The *2-Choices* process works as follows. Each node of a graph has an opinion, and in every round each node chooses independently two random neighbours and adopts their opinion if they coincide; otherwise the node keeps its own opinion. We are interested in the *plurality consensus time*, which is the first point in time where all nodes have the opinion of the initially most dominant opinion. In this chapter we put emphasize on the difference to 3-Majority and we establish a lower bound on 2-Choices showing a polynomial difference in the consensus time of both protocols. Another focus of this chapter is *Byzantine agreement*, where the goal is to achieve consensus in spite of an adversary who is allowed to change the opinion of a bounded number of nodes per round [PSL80, Rab83, BCN+14b, BCN+16, CER14, EFK+16].

Stability

In our analysis, we will show that the 2-Choices process can tolerate the presence of an adversary which is allowed to arbitrarily change the opinion of up to $F = c_1(c_1 - c_2)/(8n)$ arbitrarily selected nodes after every round. We will show that under these assumptions our 2-Choices process still guarantees that with high probability a vast majority of nodes accept the plurality opinion, that is, the initially dominant opinion. Observe that, similarly, all our theorems also hold if the adversary is allowed to change opinions at the *beginning* of a round. We use a definition similar to the definition by Becchetti et al. [BCN+16], which in turn has its roots in [AAE08, AFJ06].

Definition 10.1. *A* stabilizing near-plurality protocol *ensures the following properties:*

1. Almost agreement. *Starting from any initial configuration, in a finite number of rounds, the system must reach a regime of configurations where all but a negligible* bad *subset of nodes of size at most* $\mathrm{O}(n^{\varepsilon})$ *for some constant* $\varepsilon < 1$ *support the same opinion.*

2. Almost validity. *Given a large enough initial bias, the system is required to converge to the plurality opinion $\mathcal{A}$, with high probability, where all but a negligible* bad *set of nodes have opinion $\mathcal{A}$.*

3. Non-termination. *In dynamic distributed systems, nodes represent simple and anonymous computing units which are not necessarily able to detect any global property.*

4. Stability. *The convergence to such a weaker form of agreement is only guaranteed to hold with high probability.*

10.1 Results

The 2-Choices protocol seems to be very efficient if the number of colors is two [CER14]. The following result can be seen as an extension of Cooper et al. [CER14] on the complete graph when initially the number of opinions is larger than two. That is, we assume that every node of the complete graph $G = (V, E)$ has one of k possible opinions at the beginning, where $k = \mathrm{O}(n^{\epsilon})$ for some small positive constant ϵ. Then, the following theorem holds.

Theorem 10.2. *Consider the synchronous model on the complete graph with n nodes. Let $k = \mathrm{O}(n^{\varepsilon})$ be the number of opinions for some small constant $\varepsilon > 0$. Assume the initial bias is at least $c_1 - c_2 \geq z \cdot \sqrt{n \log n}$ for some constant z. Then 2-Choices plurality consensus process converges with high probability to the initially most dominant color within*

$$t_{2\mathrm{C}} = \mathrm{O}(n/c_1 \cdot \log n)$$

rounds. Moreover, the process fulfills the stabilizing near-plurality conditions in presence of any $F = c_1(c_1 - c_2)/(8n)$-dynamic adversary.

The difficulty in the analysis lies in the possibly diminishingly small initial *mass* of $\mathcal{A}$ in comparison to the mass of all other colors. Interestingly, the required initial gap does not depend on the number of opinions present. Moreover, we also show that if $c_1 - c_2 = \mathrm{O}(\sqrt{n})$, then $\mathcal{B}$ wins with constant probability.

Slightly later, Cooper et al. [CRRS16] proved the same run time in a much more general form by considering the class of regular expander graphs, albeit assuming a slightly more restrictive initial bias.

Finally, in Theorem 10.5, we show that the it takes $\Omega(n/\log n)$ rounds to reach consensus if all nodes start with distinct colors, *i. e.,* in the leader election setting. This shows that 3-Majority is polynomial faster (Chapter 9). On the other side our upper bound on 2-Choices shows that it beats 3-Majority in terms of the required initial bias. We defer the reader to Section 10.5 for an in-depth discussion on the compression between both dynamics.

Algorithm 8: Distributed Voting Protocol with Two Choices

Algorithm `2-Choices` $(G = (V, E)$, `color` $: V \to C)$
  for *round* $t = 1$ **to** $|C| \cdot \log |V|$ **do**
    at each node v **do in parallel**
      let $u_1, u_2 \in N(v)$ uniformly at random;
      if `color`(u_1) $=$ `color`(u_2) **then**
        `color`$(v) \leftarrow$ `color`(u_1);

10.2 Approach and Technical Contributions

This chapter rests on the shoulders of careful applications of Chernoff bounds. The main idea of the proof of the upper bound is to show that, by carefully applying Chernoff bounds, the initial bias of color 1 to all other colors increases sufficiently fast. Intuitively, the difficulty lies in the sheer number of initial opinions we allow. In contrast to what is permitted in most previous work, their total mass may significantly exceed the initial mass of c_1. We denote the number of nodes changing their opinion from color i ($\mathcal{C}_i$) to color j ($\mathcal{C}_j$) by Δ_{ij}. As mentioned before, we use $c_1, c_2, \ldots, c_k$ to denote the support of the colors $\{1, 2, \ldots, k\}$ at the beginning of a round. We will use $c'_1, \ldots, c'_k$ to denote the number of nodes of corresponding colors after the switching has been performed before the adversary changes F arbitrary nodes. Note that, $c'_1, \ldots, c'_k$ are not necessary monotonically decreasing (as opposed to $c_1, c_2, \ldots, c_k$).

Whenever we fix a configuration, we assume, w.l.o.g. that colors are ordered in descending order such that $c_1 \geq c_2 \geq \cdots \geq c_k$.

Observe that in the complete graph the number Δ_{ij} of nodes switching from color 1 to color 2 has a binomial distribution with parameters $\Delta_{ij} \sim \mathsf{Bin}(c_i, c_j^2/n^2)$. Clearly, the expectation and variance of Δ_{ij} are

$$\mathbb{E}[\,\Delta_{ij}\,] = \frac{c_i \cdot c_j^2}{n^2}$$

As for the lower bound in Theorem 10.5 showing that $t_{2\mathrm{C}} = \Omega(n/\log n)$, we refrain from applying Chernoff bounds in every step the result would be too weak. Instead we consider larger periods together with stochastic domination to derive stronger bounds.

10.3 Plurality Consensus with Two Choices

In Section 10.3.1 we show the upper bound Theorem 10.2 on the 2-Choices process. We show that if the initial bias is $\Omega(n \log n)$, then the initially most dominant color wins with high probability in $\mathrm{O}(k \cdot \log n)$ rounds.

In Section 10.4.2 we show two lower bounds: We show that if the initial bias is of order $O(\sqrt{n})$, then with constant probability a color different than color 1 will win (Theorem 10.6). Furthermore, we show that there are configurations from which we require $\Omega(k + \log n)$ rounds until any opinion wins (Theorem 10.7).

10.3.1 Upper bound - Proof of Theorem 10.2

The algorithm discussed in this section is formally defined in Algorithm 8. The interesting regime is when the largest color c_1 has non yet reached an absolute majority. Otherwise, that is if $c_1 \geq (1/2 + \varepsilon_1)n$ for some constant $\varepsilon_1 > 0$, the process converges within $O(\log n)$ steps with high probability. This follows from [CER14] since in the case of $c_1 \geq (1/2 + \varepsilon_1)n$ the size of the largest color is stochastically dominated by the size of the largest color when all other colors are merged into one single color.

For the sake of readability we assume in the following that $a \leq n/2$. Furthermore, observe that $c_1 > n/k$, since color 1 is the largest of k color classes. We introduce the following notation.

Let $S \subseteq C$ be a set of colors. We will use the random variable Δ_{iS} to denote the sum of all flows from color i to any color in S and Δ_{Si} to denote the sum of all flows from any color in S to i. We have in expectation

$$\mathbb{E}[\,\Delta_{Si}\,] = \sum_{j \in S} \frac{c_j \cdot c_i^2}{n^2} \qquad \text{and} \qquad \mathbb{E}[\,\Delta_{iS}\,] = \sum_{j \in S} \frac{c_i \cdot c_j^2}{n^2} \ .$$

For color i define $\bar{i}$ be the set of all other colors, in symbols $\bar{i} = \{j \colon j \in C \text{ with } j \neq i\}$. We observe that after one round the new number of nodes supporting color i is a random variable

$$c_i' = c_i + \sum_{j \neq i} \Delta_{ji} - \sum_{j \neq i} \Delta_{ij} = c_i + \Delta_{\bar{i}i} - \Delta_{i\bar{i}} \ .$$

Since all nodes perform their choices independently, the first sum $\Delta_{\overline{\mathcal{C}_i}i}$ has a binomial distribution with parameters $\Delta_{\overline{\mathcal{C}_i}i} \sim B(n - c_i, c_i^2/n^2)$. Furthermore, every node of color i changes its color away from $\mathcal{C}_i$ to any other opinion with probability $\sum_{j \neq i} c_j^2/n^2$. Moreover, $\Delta_{i\bar{i}}$ also has a binomial distribution with parameters $\Delta_{i\bar{i}} \sim \mathsf{Bin}(c_i, \sum_{j \neq i} c_j^2/n^2)$. In expectation

$$\mathbb{E}[\,c_i'\,] = c_i + \frac{(n - c_i)c_i^2}{n^2} - \frac{c_i}{n^2} \sum_{j \neq i} c_j^2 \ . \tag{10.1}$$

Note that these expected values are monotone w.r.t. the current size.

Observation 10.3. *Let r and s be two colors. It holds that if $c_r \leq c_s$ then $\mathbb{E}[\,c_r'\,] \leq \mathbb{E}[\,c_s'\,]$.*

Proof. By (10.1),

$$\mathbb{E}[\,c_i'\,] = c_i + \frac{c_i^2}{n} - \frac{c_i}{n^2}\sum_{\mathcal{C}_j} c_j^2 = c_i\left(1 + \frac{c_i}{n} - \sum_{\mathcal{C}_j}\frac{c_j^2}{n^2}\right) .$$

Thus,

$$\mathbb{E}[\,c_r'\,] = c_r\left(1 + \frac{c_r}{n} - \sum_j \frac{c_j^2}{n^2}\right) \overset{(c_r\le c_s)}{\le} c_s\left(1 + \frac{c_s}{n} - \sum_j \frac{c_j^2}{n^2}\right) = \mathbb{E}[\,c_s'\,] . \qquad \square$$

We are ready to prove the following lemma.

Lemma 10.4. *Assume that $c_1 - c_2 > z\cdot\sqrt{n\log n}$. There exists a constant z such that with high probability*

$$c_1' - c_2' > (c_1 - c_2)(1 + c_1/4n)$$

In the following proof we utilize certain methods which have also been used in [CER14] for the two-opinion plurality consensus process with two choices in more general graphs.

Proof. First we observe that

$$\sum_i c_i^2 = c_1^2 + \sum_{i\neq 1} c_i^2 \le c_1^2 + \sum_{i\neq 1} c_i\cdot c_2 = c_1^2 + (n - c_1)\cdot c_2 \le c_1^2 + n\cdot c_2 \tag{10.2}$$

We derive,

$$\begin{aligned}
\mathbb{E}[\,c_1' - c_2'\,] &= c_1 + \mathbb{E}[\,\Delta_{\bar 1 1}\,] - \mathbb{E}[\,\Delta_{1\bar 1}\,] - c_2 - \mathbb{E}[\,\Delta_{\bar 2 2}\,] + \mathbb{E}[\,\Delta_{2\bar 2}\,] \\
&= a + (n - c_1)\cdot\frac{c_1^2}{n^2} - \frac{c_1}{n^2}\sum_{i\neq 1} c_i^2 - c_2 - (n - c_2)\cdot\frac{c_2^2}{n^2} + \frac{c_2}{n^2}\sum_{i\neq 2} c_i^2 \\
&= c_1 - c_2 + \frac{1}{n^2}\left(c_1^2 n - c_1^3 - c_2^2 n + c_2^3 - c_1\sum_{i\neq 1} c_i^2 + c_2\sum_{i\neq 2} c_i^2\right) \\
&= c_1 - c_2 + \frac{1}{n^2}\left(n\left(c_1^2 - c_2^2\right) - c_1\left(c_1^2 + \sum_{i\neq 1} c_i^2\right) + c_2\left(c_2^2 + \sum_{i\neq 2} c_i^2\right)\right) \\
&= c_1 - c_2 + \frac{1}{n}\left(c_1^2 - c_2^2\right) - \frac{1}{n^2}\left(c_1\sum_i c_i^2 - c_2\sum_i c_i^2\right) \\
&= c_1 - c_2 + \frac{(c_1 - c_2)(c_1 + c_2)}{n} - \frac{1}{n^2}\sum_i c_i^2(c_1 - c_2) \\
&= (c_1 - c_2)\cdot\left(1 + \frac{(c_1 + c_2)}{n} - \frac{1}{n^2}\sum_i c_i^2\right) . \\
&\overset{(10.2)}{\ge} (c_1 - c_2)\left(1 + \frac{(c_1 + c_2)}{n} - \frac{c_1^2 + n\cdot c_2}{n^2}\right)
\end{aligned}$$

$$\geq (c_1 - c_2)\left(1 + \frac{c_1}{n} \cdot \left(1 - \frac{c_1}{n}\right)\right)$$
$$\geq (c_1 - c_2)\left(1 + \frac{c_1}{2n}\right) \ ,$$

where the last inequality uses $c_1 \leq n/2$. Before we apply Chernoff bounds to $c'_1 - c'_2$ we introduce the following notation.

$$\mathbb{E}[\,\Delta_{\bar{1}1}\,] = (n - c_1)\frac{c_1^2}{n^2} \ , \qquad \mathbb{E}[\,\Delta_{1\bar{1}}\,] = \frac{c_1}{n^2}\sum_{i\neq 1} c_i^2 \ ,$$
$$\mathbb{E}[\,\Delta_{\bar{2}2}\,] = (n - c_2)\frac{c_2^2}{n^2} \ , \qquad \mathbb{E}[\,\Delta_{2\bar{2}}\,] = \frac{c_2}{n^2}\sum_{i\neq 2} c_i^2 \ .$$

Furthermore, let $\delta_{\bar{1}1}$, $\delta_{1\bar{1}}$, $\delta_{\bar{2}2}$, $\delta_{2\bar{2}}$ be defined as

$$\delta_{\bar{1}1} = \frac{2\sqrt{n\log n}}{c_1} \ , \qquad \delta_{1\bar{1}} = \frac{2n\sqrt{\log n}}{\sqrt{c_1\sum_{i\neq 1} c_i^2}} \ ,$$
$$\delta_{\bar{2}2} = \frac{2\sqrt{n\log n}}{c_2} \ , \qquad \delta_{2\bar{2}} = \frac{2n\sqrt{\log n}}{\sqrt{c_2\sum_{i\neq 2} c_i^2}}$$

Since $c_1 \leq n/2$ we know for the second largest color (color 2) that $c_2 \geq n/2k$. Together with $c_1 \geq n/k \geq n^{1-\varepsilon}$ we get $0 < \delta < 1$ and $\delta_{xy}^2 \cdot \Delta_{xy} = \Omega(\log n)$ for $(\delta_{xy}, \Delta_{xy})$ defined above. We now apply Chernoff bounds to $c'_1 - c'_2$ and obtain with high probability

$$c'_1 - c'_2 \geq (c_1 - c_2) \cdot \left(1 + \frac{c_1}{2n}\right) - \sigma$$

where the "Chernoff deviation" σ is bounded as follows.

$$\begin{aligned}
\sigma &:= \delta_{\bar{1}1} \cdot \mathbb{E}[\,\Delta_{\bar{1}1}\,] + \delta_{1\bar{1}} \cdot \mathbb{E}[\,\Delta_{1\bar{1}}\,] + \delta_{\bar{2}2} \cdot \mathbb{E}[\,\Delta_{\bar{2}2}\,] + \delta_{2\bar{2}} \cdot \mathbb{E}[\,\Delta_{2\bar{2}}\,] \\
&= \frac{2\sqrt{n\log n}}{n^2}\left(c_1 n - c_1^2 + \sqrt{c_1 n\sum_{i\neq 1} c_i^2} + c_2 n - c_2^2 + \sqrt{c_2 n\sum_{i\neq 2} c_i^2}\right) \\
&\leq \frac{2\sqrt{n\log n}}{n^2}\left(\sqrt{n\sum_i c_i^2}(\sqrt{c_1} + \sqrt{c_2}) + c_1 n + c_2 n\right) \\
&\leq \frac{2\sqrt{n\log n}}{n^2}(2c_1 n + c_1 n + c_2 n) \\
&\leq \frac{8a\sqrt{n\log n}}{n} \ ,
\end{aligned}$$

where we used that $\sum_i c_i^2 \leq \sum_i c_1 \cdot c_i \leq c_1 n$. From the conditions in the statement of the lemma we know that $(c_1 - c_2) \geq z \cdot \sqrt{n\log n}$ for some constant z. If we assume that z is

large enough, e.g., $z \geq 32$, then we get with high probability

$$c_1' - c_2' \geq (c_1 - c_2) \cdot \left(1 + \frac{c_1}{4n}\right) . \qquad \square$$

While Lemma 10.4 shows that in the absence of an adversary the difference between the most dominant color and the second most dominant color increases in every round with high probability. It is easy to see that the same holds for any third color, *i. e.,* the increase in the distance between the most dominant color and any other given color is lower bounded by Lemma 10.4. By taking Union bound over all other colors we derive that the gap increases every round. To obtain a strong upper bound on the runtime, we will analyze how in the following (Theorem 10.2). the increase

Proof. Assume $c_1 - c_2 \geq z \cdot \sqrt{n \log n}$ for a sufficiently large constant z. From Lemma 10.4 we know that $c_1' - c_2' \geq (c_1 - c_2) \cdot (1 + c_1/4n)$ with high probability. By using a standard coupling, we get that $c_1' - c_j' \geq c_1' - c_2' \geq (c_1 - c_2) \cdot (1 + c_1/4n)$. Note that it may very well happen, especially if all colors have the same size except for color 1, that another color j "overtakes" color 2. However, with high probability

$$c_1' - \max_{j \geq 2} c_j' \geq (c_1 - c_2) \cdot (1 + c_1/4n).$$

We now take care of the adversary who may change up to F arbitrary nodes. Let c_i'' denote the support of color i the next round after the adversary influenced nodes. Clearly, we have

$$|c_i'' - c_i| \leq F$$

for all $i \in C$. We conclude that We have

$$c_1'' - c_j'' \geq c_1' - c_j' - 2F \geq (c_1 - c_2) \cdot (1 + c_1/4n - 2F/c_1 - c_2) \geq (c_1 - c_2) \cdot (1 + c_1/8n),$$

since $F = a(c_1 - c_2)/8n$.

Taking the union bound over all colors, we conclude that the distance between the most dominant color and every other color grows in every round by a factor of at least $(1 + c_1/4n)$ with high probability. Therefore, after $\tau = 4n/c_1$ rounds, the relative distance between color 1 and any other color doubles with high probability. Hence, the required time for color 1 to reach a size of at least $(1/2 + \varepsilon_1) \cdot n$ for a constant $\varepsilon_1 > 0$ is bounded by $\mathrm{O}(n/c_1 \cdot \log n)$. This bias is large enough that we assume in the following that all nodes which are not of color 1 are of the same color, say 2.

In absence of an adversary, we can see that after additional $\mathrm{O}(\log n)$ rounds every node has the same color 1, with high probability; see [CER14]. In each individual round, the growth described in Lemma 10.4 takes place with high probability. A union bound over all $\mathrm{O}(n/c_1 \cdot \log n)$ rounds yields that the protocol indeed converges to color 1 within

$O(n/c_1 \cdot \log n)$ rounds with high probability. The same analysis of [CER14] can be used even in the presence of an adversary. However, in this case we can only reach *almost validity* according to Definition 10.1, since the adversary is allowed to change $F = o(n)$ nodes per round.

Finally, we argue that the 2-Choices process trivially fulfills the property *almost agreement* according to Definition 10.1. Starting from an arbitrary initial distribution of colors, there is in every round a positive (albeit super-exponentially small in n) probability that all nodes adopt the same color. □

10.4 Lower Bound for 2-Choices

This section give lower bounds for 2-Choices. We first show that the consensus time can be almost linear with high probability starting from n colors (Section 10.4.1). We then complement on upper bounds of Section 10.3: Even if the initial bias is of order $\sqrt{n}$ (as opposed to $\sqrt{n \log n}$), then the second largest color might win with constant probability. Additionally, we give almost matching lower bounds on the consensus time in the setting where the initial bias is of order $\sqrt{n \log n}$.

10.4.1 Worst-Case Lower bound - Theorem 10.5

It turns out that, when started from an almost balanced configuration, the consensus time is dictated by the time it takes for one of the colors to gain a support of $\Omega(\log n)$. To prove this result, we prove a slightly stronger statement, that captures the slow initial part of the process when started from configurations with a maximal load of ℓ.

Theorem 10.5. *Let γ be a sufficiently large constant. Consider the 2-Choices process starting from any initial configuration $c \in \mathcal{C}$. Let $\ell \coloneqq \max_i c_i(0)$ be the support of the largest color. Then, for $\ell' \coloneqq \max\{2\ell, \gamma \log n\}$, it holds with high probability that no color has a support larger than ℓ' for $n/(\gamma\ell')$ rounds. In symbols,*

$$\mathbb{P}\left[\max_i c_i(t) > \ell' \text{ for some } t < \frac{n}{\gamma \ell'} \right] \leq \frac{1}{n}. \tag{10.3}$$

In particular, starting from the n-color configuration, it holds with high probability that no color has a support larger than $\gamma \log n$ for $\frac{n}{\gamma^2 \log n}$ rounds.

Proof. Let $T_i = \min\{t \geq 0 \mid c_i(t) > \ell'\}$. For any fixed opinion $i \leq k$ we show that $\mathbb{P}[T_i < n/(\gamma\ell')] \leq 1/n^2$, so that, by a union bound over all opinions and using that $T = \min\{T_i \mid i \leq k\}$, we obtain $\mathbb{P}[T < n/(\gamma\ell')] \leq 1/n$. Intuitively, we would like to show that, conditioning on $c_i \leq \ell'$, the expected number of nodes joining opinion i is dominated by a binomial distribution with parameters n and $p = (\ell'/n)^2$. The main obstacle to this is that naïvely applying Chernoff bounds for every time step yields a weak bound, since with

constant probability at each round at least one color increases its support by a constant number of nodes. Instead, we consider a new process $\mathcal{P}$ in which the number $P(t)$ of nodes supporting color i at time t majorizes $c_i(t)$ as long as $P(t) \leq \ell'$; we will then show that, after a certain time w.h.p. $P(t)$ is still smaller than ℓ' implying that $\mathcal{P}$ indeed majorizes the original process. Using the fact that in $\mathcal{P}$ we can simply apply Chernoff bounds over several rounds, we can finally get $c_i \leq P(t) \leq \ell'$ w.h.p..

Formally, process $\mathcal{P}$ is defined as follows. $P(0) := \ell$ and $P(t+1) = P(t) + \sum_{j \leq n} X_j^{(t)}$, where $X_j^{(t)}$ is a Bernoulli random variable with $\mathbb{P}[\, X_i = 1 \,] = p$ and, by a standard coupling, it is 1 whenever node j sees two times color i at round t (note that the latter event happens with probability at most p for any $t < T_i$). By definition, if $t < T_i$ it holds $c_i(t) \leq \ell'$, which implies that the probability that any node in the original process gets opinion i is at most p. Thus, we can couple 2-Choices and P for $t \leq T_i$ so that $c_i(t) \leq P(t)$. This implies that

$$T' \coloneqq \min\{t \geq 0 \mid P(t) \geq \ell'\} \preceq T_i. \tag{10.4}$$

In the remainder we show that $\mathbb{P}[\, T' < n/(\gamma\ell') \,] < 1/n^2$. For any round $t+1$, we define $\Delta_{t+1} \coloneqq P(t+1) - P(t) = \sum_{i \leq n} X_i$. Observe that $\Delta_{t+1} \sim \mathrm{Bin}(n, p)$. Let $t_0 = n/(\gamma\ell')$. In the following we bound

$$B \coloneqq P(t_0) - P(0) = \sum_{i=1}^{t_0} \Delta_i.$$

Observe that $B \sim \mathrm{Bin}(t_0 \cdot n, p)$ and thus $\mathbb{E}[\, B \,] = t_0 \cdot n \cdot p$. Using Chernoff bounds, e.g., [MU05, Theorem 4.4] we derive for any $\gamma \geq 18$

$$\begin{aligned}
\mathbb{P}[\, P(t_0) \geq \ell' \,] = \mathbb{P}[\, B \geq \ell' - \ell \,] &\leq \mathbb{P}\left[\, B \geq \max\left\{ 2\mathbb{E}[\, B \,], \frac{\gamma}{2} \log n \right\} \right] \\
&= \mathbb{P}\left[\, B \geq \mathbb{E}[\, B \,] \cdot \max\left\{ 2, 1 + \frac{\frac{\gamma}{2}\log n}{\mathbb{E}[\, B \,]} \right\} \right] \\
&\leq \exp\left(-\frac{\frac{\gamma}{2}\log n}{3} \right) \leq 1/n^3,
\end{aligned} \tag{10.5}$$

where we used that

$$\begin{aligned}
\max\left\{ 2\mathbb{E}[\, B \,], \frac{\gamma}{2} \log n \right\} &= \max\left\{ 2t_0 \cdot n \cdot p, \frac{\gamma}{2} \log n \right\} \\
&\leq \max\left\{ \frac{(\ell')^2}{\gamma\ell'}, \frac{\gamma}{2} \log n \right\} \\
&\leq \max\left\{ \frac{\ell'}{2}, \frac{\gamma}{2} \log n \right\} \leq \frac{\ell'}{2} = \ell' - \ell.
\end{aligned}$$

Putting everything together yields

$$\begin{aligned}\mathbb{P}[\,T < n/(\gamma\ell')\,] &= \mathbb{P}[\,T < t_0\,] && (10.6)\\ &\overset{(a)}{\leq} n\mathbb{P}[\,T_i < t_0\,]\\ &\overset{(b)}{\leq} n\mathbb{P}[\,T' < t_0\,]\\ &\overset{(c)}{\leq} n\mathbb{P}[\,P(t_0) \geq \ell'\,] \overset{(d)}{\leq} n^{-2}, && (10.7)\end{aligned}$$

where in (a) we used union bound over all colors, in (b) we used (10.4), in (c) we used that "$T' < t_0$" $\implies$ "$P(t_0) \geq \ell'$" and in (d) we used (10.5). This completes the proof. □

10.4.2 Complementing our Upper bounds Theorem 10.6 and Theorem 10.7

In this section we give lower bounds complementing our positive results: In the previous section, we proved that the process converges to color 1 with high probability if the initial imbalance $c_1 - c_2$ is not too small. Precisely, Theorem 10.2 states that if $c_1 - c_2 \geq z \cdot \sqrt{n \log n}$ for some constant z, color 1 wins with high probability. Conversely, in the following section we examine a lower bound on the initial bias. We will show, as stated in Theorem 10.6, that for an initial bias $c_1 - c_2 \leq z \cdot \sqrt{n}$ for some constant z we have a constant probability that color 2 "overtakes" color 1 in the first round, that is, $\mathbb{P}[\,c'_1 < c'_2\,] = \Omega(1)$ implying that color 2 wins w.c.p..

Our lower bound is based on the approximation of the binomial distribution with the normal distribution, which allows us to obtain a lower bound. In order to so, we apply the DeMoivre-Laplace limit theorem (Theorem A.6)

We now prove Theorem 10.6 which states that there exists an initial color assignment for which $c_1 = c_2 + z' \cdot \sqrt{n}$ but color 2 wins with constant probability even in absence of an adversary.

Theorem 10.6 (Lower Bound on the Initial Bias)**.** *For any $k \leq \sqrt{n}$ and constant z' there exists an initial assignment of colors to nodes for which $c_1 = c_2 + z' \cdot \sqrt{n}$ but $\mathbb{P}[\,c'_1 < c'_2\,] = \Omega(1)$ even in absence of an adversary.*

Proof. Let $z = z'/2$ and $n' = \frac{n-k+2}{2}$. Assume that we have the following initial color distribution among the nodes.

$$(c_1, c_2, c_3, \ldots, c_k) = (\lfloor n' \rfloor + \lfloor z \cdot \sqrt{n} \rfloor, \lceil n' \rceil - \lfloor z \cdot \sqrt{n} \rfloor, 1, \ldots, 1).$$

Clearly, $\sum_{\mathcal{C}_j} c_j = n$. In the following we will omit the floor and ceiling functions for the sake of readability reasons. First, we start by giving an upper bound on the number of nodes which change their color away from color 2. Now recall that $\Delta_{2\bar{2}}$ follows a binomial

distribution $\Delta_{2\bar{2}} \sim \mathsf{Bin}(b, \sum_{Cj \neq B} c_j^2/n^2)$ with expected value

$$\begin{aligned}\mathbb{E}[\,\Delta_{2\bar{2}}\,] &= c_2 \cdot \frac{c_1^2 + k - 2}{n^2} = (n' - z\cdot\sqrt{n}) \cdot \frac{(n' + z\cdot\sqrt{n})^2 + k - 2}{n^2} \\ &\leq \frac{(n' + z\cdot\sqrt{n})^3 + k - 2}{n^2} \leq \frac{n}{8} + 4z\sqrt{n}\ .\end{aligned}$$

Applying Chernoff bounds to $\Delta_{2\bar{2}}$ gives us

$$\mathbb{P}\Big[\,\Delta_{2\bar{2}} \geq \Big(1 + \sqrt{3/\mathbb{E}[\,\Delta_{2\bar{2}}\,]}\Big)\mathbb{E}[\,\Delta_{2\bar{2}}\,]\,\Big] \leq 1/e\ . \tag{10.8}$$

That is, with constant probability at least $1 - 1/e$ we have

$$\Delta_{2\bar{2}} \leq \Big(1 + \sqrt{3/\mathbb{E}[\,\Delta_{2\bar{2}}\,]}\Big)\mathbb{E}[\,\Delta_{2\bar{2}}\,] \leq \frac{n}{8} + 4z\sqrt{n} + \sqrt{3\mathbb{E}[\,\Delta_{2\bar{2}}\,]} \leq \frac{n}{8} + (4z+1)\sqrt{n}\ .$$

Secondly, we give the following lower bound on the number of nodes which change their color from color 1 to color 2 and $\Delta_{12} \sim \mathsf{Bin}(a, b^2/n^2)$ with expected value

$$\begin{aligned}\mathbb{E}[\,\Delta_{12}\,] &= (n' + z\cdot\sqrt{n}) \cdot \frac{(n' - z\cdot\sqrt{n})^2}{n^2} \geq \frac{(n' - z\cdot\sqrt{n})^3}{n^2} \\ &\geq \frac{(n/2 - (z+1/2)\sqrt{n})^3}{n^2} \geq \frac{n}{8} - 4z\sqrt{n}\end{aligned}$$

and variance

$$\mathrm{Var}[\,\Delta_{12}\,] = \mathbb{E}[\,\Delta_{12}\,] \cdot \left(1 - \frac{(n' - z\cdot\sqrt{n})^2}{n^2}\right) \geq \frac{n}{9}\cdot\frac{1}{2} = \frac{n}{18}\ .$$

We now apply Theorem A.6 to Δ_{12}. Let $x = \frac{\sqrt{18}}{2}(18z+4)$. We derive

$$\mathbb{P}\Big[\,\Delta_{12} \geq \mathbb{E}[\,\Delta_{12}\,] + x\cdot\sqrt{\mathrm{Var}[\,\Delta_{12}\,]}\,\Big] = \frac{1}{\sqrt{2\pi}\cdot x}\exp\Big(-x^2/2\Big) \pm \mathrm{o}(1) = \Omega(1)\ .$$

That is, we have with constant probability

$$\Delta_{12} \geq \mathbb{E}[\,\Delta_{12}\,] + x\cdot\sqrt{\mathrm{Var}[\,\Delta_{12}\,]} \geq \frac{n}{8} - 4z\sqrt{n} + x\cdot\sqrt{\frac{n}{18}}\ . \tag{10.9}$$

Finally, assume that in the worst-case every node of colors $3,\ldots,k$ changes to color 1 but not a single node changes away from color 1 to these colors $3,\ldots,k$. Observe that $\Delta_{2\bar{2}}$ is an upper bound on Δ_{21}. Therefore,

$$\begin{aligned}c_1' - c_2' &\leq (c_1 + k - 2 + \Delta_{21} - \Delta_{21}) - (c_2 + \Delta_{12} - \Delta_{2\bar{2}}) \leq c_1 - c_2 + k - 2 + 2\Delta_{2\bar{2}} - 2\Delta_{12} \\ &\leq 2z\cdot\sqrt{n} + k - 2 + 2\Delta_{2\bar{2}} - 2\Delta_{12} \leq (2z+1)\cdot\sqrt{n} + 2\Delta_{2\bar{2}} - 2\Delta_{12}\ .\end{aligned}$$

We plug in (10.8) and (10.9) to bound the random variables Δ_{12} and $\Delta_{2\overline{2}}$ and obtain with constant probability

$$\begin{aligned} c_1' - c_2' &\leq (2z+1)\cdot\sqrt{n} + 2\left(\frac{n}{8} + (4z+1)\sqrt{n}\right) - 2\left(\frac{n}{8} - 4z\sqrt{n} + x\cdot\sqrt{\frac{n}{18}}\right) \\ &= (2z+1+8z+2+8z-2x/\sqrt{18})\cdot\sqrt{n} = (18z+3-2x/\sqrt{18})\cdot\sqrt{n} \end{aligned}$$

which gives us $c_1' - c_2' < 0$ for $x = \frac{\sqrt{18}}{2}(18z+4)$. Therefore, we have $\mathbb{P}[\, c_1' < c_2' \,] = \Omega(1)$ and thus we conclude that color 2 wins with constant probability. □

Theorem 10.7 (Lower Bound on the Run Time)**.** *Assume the initial bias is exactly $z\sqrt{n\log n}$ for some constant z. The number of rounds required for the plurality consensus process defined in Algorithm 8 to converge is at least $\Omega(n/c_1 + \log n)$ with constant probability, even in absence of an adversary.*

Proof. Let $c_1(t)$ denote the size of color 1 in round t and of initial size $c_1(0) = n/k + z\cdot\sqrt{n\log n}$. Furthermore, assume that $k \geq 3\cdot z$. We show by induction on the rounds that $c_1(t) \leq c_1(0)\cdot(1+3\cdot c_1(0)/n)^t$ for $1 \leq t \leq n/(10\cdot c_1(0))$ with probability $1 - t/n$. First we note that

$$\begin{aligned} c_1(t) &\leq c_1(0)\cdot\left(1+3\cdot\frac{c_1(0)}{n}\right)^t \leq c_1(0)\cdot\left(1+3\cdot\frac{c_1(0)}{n}\right)^{n/(10\cdot c_1(0))} \leq c_1(0)\cdot\exp(1/2) \\ &\leq 2\cdot c_1(0) \end{aligned} \tag{10.10}$$

We now prove the induction claim. The base case holds trivially. Consider step $t+1$. By induction hypothesis we have with probability at least $1 - t/n$ that $c_1(t) \leq c_1(0)\cdot(1+3\cdot c_1(0)/n)^t$. Note that we have with high probability

$$c_1(t+1) \leq c_1(t) + \Delta_{\overline{1}1} \leq c_1(t) + \left(1 + \frac{\sqrt{3\log n}}{\sqrt{\mathbb{E}[\,\Delta_{\overline{1}1}\,]}}\right)\cdot\mathbb{E}[\,\Delta_{\overline{1}1}\,]\ ,$$

where the latter inequality follows by Chernoff bounds. Using (10.10) and $c_1(t) \leq c_1(0)$, we derive

$$\begin{aligned} c_1(t+1) &\leq c_1(t) + \left(1 + \frac{\sqrt{3\log n}}{\sqrt{c_1(t)^2/(2\cdot n)}}\right)\frac{c_1(t)^2}{n} \leq c_1(t) + \left(1 + \frac{\sqrt{3\log n}}{\sqrt{c_1(0)^2/(2\cdot n)}}\right)\frac{c_1(t)^2}{n} \\ &\leq c_1(t) + \frac{3}{2}\cdot\frac{c_1(t)^2}{n} = c_1(t)\cdot\left(1 + \frac{3}{2}\cdot\frac{c_1(t)}{n}\right) \leq c_1(t)\cdot\left(1+\frac{3\cdot c_1(0)}{n}\right)\ . \end{aligned}$$

From the induction hypothesis we therefore obtain

$$c_1(t+1) \leq c_1(0)\cdot\left(1+\frac{3\cdot c_1(0)}{n}\right)^t\cdot\left(1+\frac{3\cdot c_1(0)}{n}\right) = c_1(0)\cdot\left(1+\frac{3\cdot c_1(0)}{n}\right)^{t+1}\ .$$

Using a union bound to account for all errors, we derive that with probability at least $1-(t+1)/n$ we have $c_1(t+1) \leq c_1(0) \cdot (1+3 \cdot c_1(0)/n)^{t+1}$, which completes the proof of the induction and proves the lower bound of $\Omega(n/a)$.

In the remainder we establish the bound $\Omega(\log n)$. Assume only two colors 1 and color 2 where color 1 initial size $c_1(0) = n/2 + \sqrt{n}\log n$. We show by induction on the rounds that $c_1(t) \leq c_1(0) + 6^t\sqrt{n}\log n$ for $1 \leq t \leq \log n/20$ with probability $1-2t/n$. First we note that $c_1(t) \geq c_1(0)$ and

$$c_1(t) \leq c_1(0) + 6^t\sqrt{n}\log n \leq n/2 + n^{5/6} < n \ .$$

We now prove the induction claim. The base case holds trivially. Consider step $t+1$. By induction hypothesis we have with probability at least $1-2t/n$ that $c_1(t) \leq c_1(0) + 6^t\sqrt{n}\log n$. We have, using $c_1 = c_1(t)$ and $\sigma = 6^t\sqrt{n}\log n$,

$$\begin{aligned} n^2 \cdot \mathbb{E}[\,\Delta_{\bar{1}1} - \Delta_{1\bar{1}}\,] &= (n-1)c_1^2 - c_1 \cdot (n-c_1)^2 = (n-c_1)c_1(2c_1-n) \\ &\leq n/2 \cdot c_1 \cdot 2\sigma = n \cdot \sigma(n+\sigma) = n^2 \cdot \sigma + n \cdot \sigma^2 \ . \end{aligned}$$

Similar to before, we obtain by Chernoff bounds that with high probability

$$\begin{aligned} c_1(t+1) - c_1(t) = \Delta_{\bar{1}1} - \Delta_{1\bar{1}} &\leq \left(1 + \frac{\sqrt{3\log n}}{\sqrt{\mathbb{E}[\,\Delta_{\bar{1}1}\,]}}\right)\mathbb{E}[\,\Delta_{\bar{1}1}\,] - \left(1 - \frac{\sqrt{3\log n}}{\sqrt{\mathbb{E}[\,\Delta_{1\bar{1}}\,]}}\right)\mathbb{E}[\,\Delta_{1\bar{1}}\,] \\ &\leq \mathbb{E}[\,\Delta_{\bar{1}1} - \Delta_{1\bar{1}}\,] + 2\sqrt{3\log n} \cdot \sqrt{\mathbb{E}[\,\Delta_{\bar{1}1}\,]} \\ &\leq \sigma + \sigma^2/n + 2\sqrt{3\log n} \cdot \sqrt{n} \leq 3\sigma \ . \end{aligned}$$

From the induction hypothesis we therefore obtain

$$c_1(t+1) \leq c_1(0) + 6^t\sqrt{n}\log n + 3\sigma \leq c_1(0) + 6^{t+1}\sqrt{n}\log n \ ,$$

which completes the induction and yields the lower bound of $\Omega(\log n)$. □

10.5 Comparison with the 3-Majority Process

In this section we elaborate on the difference between the 2-Choices process and the 3-Majority rule [BCN+14b], where in the latter each node pulls the opinion of three random neighbors and adopts the majority opinion among those three, breaking ties uniformly at random. As mentioned before, the 3-Majority process of [BCN+14b] uses $O(\log k)$ memory bits and the authors prove a tight run time of $\Theta(k \cdot \log n)$ for this protocol, given a sufficiently high bias $c_1 - c_2$. Moreover, they show that if the bias is only of order $\sqrt{kn}$, then with constant probability the difference $c_1 - c_2$ decreases. This is fundamentally different in the 2-Choices process, which requires only a bias of $O(\sqrt{n\log n})$.

The reasons are twofold. First, the variance of the 3-Majority process can be orders of magnitude larger and second, the expected increase in the difference between the largest and second largest color in the 3-Majority process is only of order of the variance. As for the variance, consider an initial setting where all colors are of sublinear size and colors 1 and 2 are larger than all other colors, such that $c_1 = o(n)$ and

$$c_2 = c_1 - c\sqrt{n \log n} > c_j$$

and

$$c_j = (n - c_2 - c_1)/(k-2)$$

for all $3 \leq j \leq k$ with $k = n^{\varepsilon}$ for constants ε and c. Observe that the expected numbers of color switches differ significantly. In the 2-Choices process it is very unlikely for a node to pick the same color twice and the probability of switching is o(1). In contrast to this, the probability of switching in the 3-Majority process is $1 - \mathrm{o}(1)$.

More illustratively, consider the number of switches to color 2. By Lemma 2.1 of [BCN+14b], the probability that a node switches to color 2 in the 3-Majority process is $p \in [c_2/n, 2c_2/n]$ and the variance becomes $n \cdot p \cdot (1-p) \geq c_2/2$. However, in the 2-Choices process, the probability of switching to color 2 is $q = c_2^2/n^2$ and the variance is thus at most $n \cdot q \cdot (1-q) \leq n \cdot q = c_2^2/n$, which is considerably smaller than $c_2/2$. This high variance paired with the small expected increase in the difference between color 1 and color 2 easily becomes fatal. Again, by Lemma 2.1 of [BCN+14b], one can verify that $\mathbb{E}[\, c_1' - c_2' \,] \leq c_1 - c_2 + (c_1^2 - c_2^2)/n$. Now we have $\mathbb{P}[\, c_1' \leq \mathbb{E}[\, c_1' \,]\,] = \Omega(1)$ and, using the large variance, we obtain from Theorem A.6 that

$$\begin{aligned}\mathbb{P}\Big[\, c_2' \geq c_2 + (c_1^2 - c_2^2)/n \big| c_1' \leq \mathbb{E}[\, c_1' \,]\, \Big] &\geq \mathbb{P}\Big[\, c_2' \geq c_2 + (c_1^2 - c_2^2)/n \,\Big] \\ &\geq \mathbb{P}[\, c_2' \geq \mathbb{E}[\, c_2' \,] + \mathrm{Var}[\, c_2' \,]\,] = \Omega(1) \ .\end{aligned}$$

Thus the distance between color 1 and 2 *decreases* with constant probability, that is,

$$\mathbb{P}[\, c_1' - c_2' < c_1 - c_2 \,] = \Omega(1).$$

In comparison to this, we have seen in Section 10.3 that in the given setting the distance between color 1 and color 2 in the 2-Choices process increases with high probability.

Chapter 11

Rapid Asynchronous Consensus [EFK+16]

We consider the following plurality consensus process on the complete graph of size n. We consider the asynchronous setting (cf. e.g. [MS08]). In the *asynchronous model*, we assume that each node has a random clock which ticks according to a Poisson distribution, once per unit of time in expectation [MS08] (note that this model is equivalent to the continuous-time model of [AF02], page 26 (ii), according to which random walks move along the vertices of a graph). Again, upon activation a node updates its opinion according to a sample of its neighborhood.

Regardless of the underlying model of synchronicity, if eventually all nodes agree on one opinion, we say this opinion *wins*, and the process *converges*. Typically, one would demand from such a voting procedure to run accurately, that is, the opinion with the largest initial support should win with decent probability $(1-o(1))$, and to be efficient, that is, the voting process should converge within as few communication steps as possible.

Our goal is to design a simple algorithm which reaches quickly plurality consensus.

Model

In the following section, we will introduce formally the model which we consider in the remainder of this chapter. We give a formal definition of the consensus process in the synchronous and the asynchronous model followed by an overview of our results in Section 10.1.

We consider the following plurality consensus process on the complete graph of size n.

Synchronous Model

In the synchronous model we assume that the protocol operates in discrete rounds. In each round, the nodes may simultaneously sample other nodes uniformly at random and then simultaneously change their opinion as a function of the observed samples. An example

here is the 2-Choices process where in each round every node samples two nodes chosen uniformly at random, with replacement. If the chosen nodes' colors coincide, then the node adopts this color. We denote this process as the *plurality consensus process with two choices*. Our first two results will be shown w.r.t. this synchronous model.

Parallel Asynchronous Model

In the asynchronous model we consider, every node v is equipped with a random clock which ticks according to a Poisson distribution with parameter $\lambda = 1$. Whenever a node ticks, it may sample nodes chosen uniformly at random and update its opinion based on the sampled values.

As discussed in [BGPS06, MS08], this model is equivalent to the setting, in which a global clock ticks according to a Poisson process of rate n. If T_i denotes the i'th tick of this clock, then the time differences $T_i - T_{i-1}$ are identically distributed independent random variables which have exponential distribution with parameter n. At each tick of this global clock, a node is chosen uniformly at random, and the global tick corresponds to the local tick at this node. As shown before, there is a strong correspondence between the number of ticks and the absolute time, meaning that with probability $1 - e^{-\Omega(n)}$ the k'th tick of the global clock occurs at some time $\Theta(k/n)$ for any $k = \Omega(n)$.

Sequential Asynchronous Model

While the parallel model described above represents real-world processes for which event frequencies are commonly modeled by Poisson clocks, we give in the following a more theoretical model. Instead of considering the asynchronous parallel process in continuous time, we rather analyze the process in the sequential model. In this sequential model, we assume that a discrete time is given by the sequence of ticks, and at any of the discrete time steps, a node is selected uniformly at random from the set of all nodes to perform its task. According to the global clock model described above, and using the fact that for any $k = \Omega(n)$ the k'th tick of the global clock occurs at some time $\Theta(k/n)$, with probability $1 - e^{-\Omega(n)}$, we can relate the number of ticks in the sequential model to the continuous time in the asynchronous model. Since k ticks in the global clock model occur within time $\Theta(k/n)$ for any $k = \Omega(n)$ with probability $1 - e^{-\Omega(n)}$, the number of ticks divided by the number of nodes in the sequential model and the time in the global clock model are asymptotically the same with probability $1 - e^{-\Omega(n)}$.

11.1 Results

Our main contribution is an efficient algorithm for plurality consensus in the asynchronous setting. As discussed below in more detail, a straight-forward observation is that in the

sequential asynchronous model many nodes may remain *unselected* for up to $O(\log n)$ time, which implies that no algorithm can converge in $o(\log n)$ time. Thus, our aim is to construct a protocol that solves plurality consensus in $O(\log n)$ time. We show that if the difference between the numbers of the largest two opinions is at least $\Omega(c_2)$, where c_2 is the size of the second largest opinion, and $k = n^{O(1/\log\log n)}$, then our algorithm solves plurality consensus and achieves the best possible run time of $O(\log n)$, provided a node is allowed to communicate with at most constantly many other nodes in a step.

The key to the rapidity of our protocol is that we pair a phase in which all nodes execute the 2-Choices process with a phase in which successful opinions are propagated quickly–much like in broadcasting. For this to work it is crucial to separate the two phases. While this is trivial in the synchronous setting, it is impossible in the asynchronous setting. The number of activations of different nodes can easily differ by $\Theta(\log n)$, rendering any attempt of full synchronization futile if one aims for a run time of $O(\log n)$. Thus, we restrict ourselves to the concept of *weak synchronicity* as follows. At any time we only require that a $(1 - o(1))$-fraction of nodes are *almost* synchronous. To cope with the influence of the remaining nodes, we rely on a toolkit of gadgets, which we believe are interesting in their own right. Our result is formally stated in the following theorem.

Theorem 11.1. *Consider the asynchronous model. Let $G = K_n$ be the complete graph with n nodes. Let $k = O(\exp(\log n/\log\log n))$ be the number of opinions. Let $\varepsilon_{\text{bias}} > 0$ be a constant. Assume $c_1 \geq (1 + \varepsilon_{\text{bias}}) \cdot c_i$ for all $i \geq 2$, then the asynchronous plurality consensus process defined in Section 11.2 on G converges within time $\Theta(\log n)$ to the majority opinion $\mathcal{C}_1$, with high probability.*

The result of Theorem 11.1 can be extended to the following asynchronous model that takes into account latency and response times when a node is contacted. In this extended model, every node ticks according to a Poisson process with parameter $\lambda = 1$, and the response time of a node has exponential distribution with some constant parameter. Furthermore, we believe that the result of the theorem, and the methods used in its proof, can be extended beyond the Poisson clock process–see Section 11.11 for a discussion.

11.2 Approach and Technical Contributions

In this section, we introduce our asynchronous consensus algorithm and give the intuition behind the proof of Theorem 11.1. To simplify the presentation we start by discussing our synchronous consensus protocol. A detailed description and its proofs can be found in [EFK+16]. Afterwards, we introduce a framework allowing us to adapt the algorithm to the asynchronous setting.

An Ideal World: Synchronous Consensus A simple algorithm for reaching consensus is 2-Choices. As we show in detail in the full version, a bias of $c_1 - c_2 = O(\sqrt{n \log n})$ between

the two largest colors suffices to guarantee that the largest color 1 wins after $O(n/c_1 \cdot \log n)$ rounds with high probability.

In order to achieve a poly-logarithmic run time we do the following. We combine the guarantee of the 2-Choices process to reach plurality consensus with the speed of broadcasting. More specifically, the protocol consists of $\Theta(\log(n/c_1) + \log\log n)$ phases which in turn consist of two sub-phases: (i) one round of the 2-Choices process and (ii) several rounds of the so-called Bit-Propagation sub-phase in which each node that changed its opinion during the preceding two-choice step broadcasts its new opinion.

More precisely, we equip each node with an additional bit of memory which is set to True if and only if it changed its opinion in the 2-Choices sub-phase. In the Bit-Propagation sub-phase, each node u samples nodes randomly until a node v with its bit set to True is found. Then u adopts v's opinion and sets its own bit to True, which means that subsequently any node sampling u will set their bit as well.

The first sub-phase ensures that after the 2-Choices round the number of nodes holding opinion $\mathcal{C}_1$ and having their bit set to True is concentrated around c_1^2/n. After the Bit-Propagation sub-phase all nodes will have their bit set, and the distribution and the size of $\mathcal{C}_j$'s support is concentrated around $c_j^2/x(1)$, where $x(1)$ is the total number of bits set after the 2-Choices sub-phase. This is enough to show that after $O(\log(n/c_1))$ rounds the distance between $\mathcal{C}_1$ and any opinion $\mathcal{C}_j \neq \mathcal{C}_1$ increases quadratically, that is, $c_1'/c_j' \geq (1-o(1))c_1^2/c_j^2$. Due to the quadratic growth in the distance between $\mathcal{C}_1$ and every other opinion, the number of required phases is only of order $\Theta(\log(n/a) + \log\log n)$. We assume that every node is aware of (upper bounds on) n and k, allowing us to use these values within the algorithm, and in particular to run it in multiple phases of length $\Theta(\log k + \log\log n)$ each.

Towards an Asynchronous Algorithm. We now introduce our asynchronous protocol to solve plurality consensus. In the sequential asynchronous model we assume that a sequence of discrete time steps is given, where at each time step one node is chosen uniformly at random to perform its tick.

Algorithm 9: Distributed Voting Protocol with Two Choices

```
C_1 lgorithm two-choices(G = (V, E), color : V → C) for round t = 1 to |C| · log |V| do
    C_1 tNode v let u_1, u_2 ∈ N(v) uniformly at random;
    if color(u_1) = color(u_2) then
        color(v) ← color(u_1);
```

The key to the speed of the synchronous algorithm is the combination of the two-choice process with an information dissemination process. However, this interweaving of these processes requires that the nodes execute the sub-phases simultaneously. While this is trivially the case in the synchronous setting, it is extremely unlikely in the asynchronous setting, since the numbers of ticks of different nodes may differ by up to $O(\log n)$. Therefore, any attempt to reach full synchronization is futile if one aims for a run time of $O(\log n)$.

To overcome this restriction, we adopt the following weaker notion of synchronicity. At any time we only require a $(1 - o(1))$ fraction of the nodes to be *almost synchronous*. This relaxes full synchronicity in three ways: First, nodes are only almost synchronous, meaning that for any two nodes their working times may differ by up to $\Delta = \Theta(\log n / \log \log n)$. Secondly, we allow $o(n)$ nodes to be poorly synchronized. Finally, we require this to hold only with high probability.

The above notion does not require the nodes to synchronize actively per se, since their number of ticks is to some extent concentrated even without active synchronization. However, it turns out that without synchronizing perpetually, the number of poorly synchronized nodes in each phase will become larger than the initial bias towards the plurality opinion $c_1 - c_2$ and could therefore influence the consensus significantly. We thus actively synchronize nodes at the end of each phase to decrease the fraction of poorly synchronized nodes such that their number is in $o(c_1 - c_2)$, resulting in a negligible influence of those nodes.

Once several technical challenges are resolved, the resulting weak synchronicity allows us to reuse the high-level structure of the synchronous algorithm(`OneExtraBit`). As in the synchronous case, the asynchronous protocol consists of one 2-Choices sub-phase and one Bit-Propagation sub-phase, the latter of which propagates the choices of the 2-Choices phase to all nodes in the network. In addition to these sub-phases we have a third sub-phase in which we synchronize nodes.

After executing the first two sub-phases, the relative difference between $\mathcal{C}_1$ and any opinion $\mathcal{C}_j \neq A$ increases quadratically and thus we only require $O(\log \log n)$ such phases. Each of the sub-phases has a length of $O(\log n / \log \log n)$, amounting to a total run-time of $O(\log n)$. While superficially the asynchronous version looks very similar to the synchronous protocol (`OneExtraBit`), the analysis differs greatly from the synchronous case, in both approach and technical execution.

11.3 The Asynchronous Protocol

Our asynchronous protocol consists of two parts, Part 1 defined in Algorithm 10 later in this section and Part 2 defined in Algorithm 11 in Section 11.8. In these formal definitions, we specify the operations that each node performs when selected to tick. The goal of the first part is to increase the number of nodes of color 1 to at least $c_1 \geq (1 - \varepsilon_{\text{Part1}}) \cdot n$ for some small constant $\varepsilon_{\text{Part1}}$. Once the execution of the first part has finished, the nodes execute a simple 2-Choices algorithm in an asynchronous manner. We will show that after the second part, $\mathcal{C}_1$ wins with high probability. Our main contribution is the analysis of the first part. For the sake of completeness, we formally analyzethe second part in Section 11.8.

In contrast to the formal definitions, it is more convenient and instructive to represent the algorithm executed by each node in a graphical way. This graphical representation for a single phase of the first part is shown in Figure 11.1. In this graphical representation, the

Algorithm 10: Part 1 of the asynchronous protocol to solve plurality consensus. Both variables `realtime` and `workingtime` are initialized to 0, and `samples` is initially the empty set. The variables κ and ℓ denote large constants. The goal of the algorithm is to increase the plurality opinion $\mathcal{C}_1$ such that $c_1 \geq (1 - \varepsilon_{\text{Part1}}) \cdot n$ for a small constant $\varepsilon_{\text{Part1}}$.

Algorithm `asynchronous(`*node* v`)` **(Part 1)**

let $T = \kappa \cdot \log n / \log\log n$;
let $t =$ `workingtime(`v`)` $\bmod T$;

if $t = T/10$ **then**
  let $u_1, u_2 \in N(v)$ uniformly at random;
  if `color(`u_1`)` $=$ `color(`u_2`)` **then**
    `intermediate(`v`)` $\leftarrow$ `color(`u_1`)`;
  else
    `intermediate(`v`)` $\leftarrow$ Null;
else if $t = 2 \cdot T/10$ **then**
  if `intermediate(`v`)` $\neq$ Null **then**
    `color(`v`)` $\leftarrow$ `intermediate(`v`)`;
    `bit(`v`)` $\leftarrow$ True;
  else
    `bit(`v`)` $\leftarrow$ False;
else if $t \in [3 \cdot T/10,\ 7 \cdot T/10]$ **then**
  if `bit(`v`)` $=$ False **then**
    let $u \in N(v)$ uniformly at random;
    if `bit(`u`)` $=$ True **then**
      `bit(`v`)` $\leftarrow$ True;
      `color(`v`)` $\leftarrow$ `color(`u`)`;
⋮

⋮
else if $t \in [8 \cdot T/10,\ 9.5 \cdot T/10]$ **then**
  increase all values in `samples(`v`)` by 1;
  if $t \in \left[8 \cdot T/10,\ 8 \cdot T/10 + \log^3 \log n\right]$ **then**
    let $u \in N(v)$ uniformly at random;
    `samples(`v`)` $\leftarrow$ `samples(`v`)` $\cup$ {`realtime(`u`)`};
  if $t = 9.5 \cdot T/10$ **and** `samples(`v`)` $\neq \emptyset$ **then**
    `workingtime(`v`)` $\leftarrow$ `median(samples(`v`))`;
    `samples(`v`)` $\leftarrow \emptyset$;
else
  do nothing;

`realtime(`v`)` $\leftarrow$ `realtime(`v`)` $+ 1$;
`workingtime(`v`)` $\leftarrow$ `workingtime(`v`)` $+ 1$;
if `workingtime(`v`)` $\geq \kappa \cdot \ell \cdot \log n$ **then**
  continue with Algorithm 11;

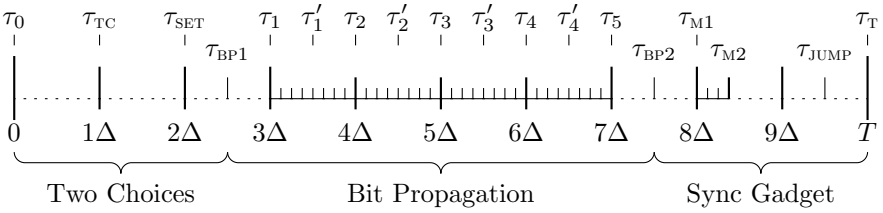


Figure 11.1: Graphical representation of one phase of Algorithm 10. Each phase consists of $T = 10 \cdot \Delta$ ticks.

instructions are drawn on a line from left to right, starting with the first instruction at the left endpoint.

As in the synchronous case, the asynchronous algorithm operates in multiple phases. Each of these phases is split into three sub-phases. Each sub-phase consists of multiple blocks of length Δ each. During these sub-phases, according to Algorithm 10, there are multiple blocks of instructions where nodes for a long time literally *do nothing*. These

do-nothing-blocks are used, in combination with the following result on synchronicity, to ensure that a large fraction of nodes executes critical instructions at almost the same time. That is, for a large fraction of nodes we will show that these nodes execute instructions as if they were bulk synchronized, which they clearly are not.

The first phase is the 2-Choices sub-phase, which consists of two instructions, the 2-Choices step and the commit step. In the 2-Choices step, every node samples two neighbors uniformly at random. If and only if these neighbors' colors coincide, the node sets an `intermediate` color to the neighbors' colors. In the commit step, nodes change their color if they have their intermediate color set and then set their bit accordingly. The second phase is the Bit-Propagation sub-phase, which closely resembles the synchronous counter part. Finally, in the third phase, all nodes execute the so-called *Sync Gadget*. In this gadget, nodes adjust their *working time* in order to synchronize. Our perpetual synchronization mechanism is described after the following definitions.

For the analysis of the asynchronous algorithm we will use the following notation and definitions.

Definitions. Let κ and ℓ denote sufficiently large positive constants. We refer to a series of n consecutive time steps as a *period*, and we combine $T = \kappa \cdot \log n / \log\log n$ periods to a *phase*. The first part of the asynchronous protocol consists of $\ell \cdot \log\log n$ phases. Intuitively, a period is the number of time steps during which each node ticks in expectation once. We define a *reference point* τ to be a time step which marks the end of a period τ. In particular, at reference point τ there have been $\tau \cdot n$ time steps, and each node has ticked in expectation τ times.

- *Let $T_v(t)$ denote the random variable for the* real time*, the number of ticks of node v after the first $t \cdot n$ time steps. That is, $T_v(t)$ denotes the number of times v was scheduled during the first $t \cdot n$ ticks.*
- *Let $T'_v(t)$ denote the random variable for the* working time*, the current instruction counter of node v after the first $t \cdot n$ time steps. Note that $T'_v(t)$ can differ from $T_v(t)$ since the working time is adjusted with the goal of synchronization in Algorithm 10.*

At the beginning of the algorithm, both, the real time and the working time are initialized to 0. Since at each time step one node is chosen to tick independently and uniformly at random, $T_v(\tau)$ has a binomial distribution $T_v(\tau) \sim \mathsf{Bin}(\tau \cdot n,\ 1/n)$ with expected value $\mathbb{E}[\,T_v(\tau)\,] = \tau$. It will prove convenient to regard a reference point as the one instruction in the algorithm which would be executed in the corresponding period if every node ticked exactly once in every period.

Weak Perpetual Synchronization. In the asynchronous algorithm, when a node is selected to tick, all operations are performed based on the node's current working time. In

contrast, the real time of a node is used to always the total number of ticks performed so far by this node. In the Sync Gadget, the working time T'_v of a node v, denoted as `workingtime` in Algorithm 10, is adjusted as follows.

The Sync Gadget consists of a sampling sub-phase $[\tau_{\text{M1}}, \tau_{\text{M2}}]$ and a jump step τ_{JUMP}. The sampling sub-phase of the Sync Gadget consists of $\log^3 \log n$ ticks. During these ticks, every node samples a neighbor uniformly at random and collects the real time T_u of the sampled neighbor u. Additionally, the node increments all real times sampled so far by 1 until the jump step is executed. At the jump step, the node sets its working time to the median of the samples.

During the entire phase, according to Algorithm 10, there are multiple blocks of instructions where nodes literally *do nothing*. These blocks are used, in combination with the following result on synchronicity, to ensure that a large fraction of nodes executes critical instructions at almost the same time. That is, for a large fraction of nodes we will show that these nodes execute instructions as if they were bulk synchronized, which they clearly are not.

11.4 The Key Lemmas

The use of the Sync Gadget and the following definition of Δ-closeness allow us to show Proposition 11.3 which forms the basis for our adaption of the synchronous protocol to the asynchronous setting.

Definition 11.2. *We say a node is Δ-close to a reference point τ w.r.t. the real time T_v or the working time T'_v, if $|T_v(\tau) - \tau| \leq \Delta$ or $|T'_v(\tau) - \tau| \leq \Delta$, respectively. If we say a node is Δ-close without specifying a reference point, we mean that it is Δ-close to the expected number of ticks.*

Proposition 11.3. *Let $\mathcal{S}$ be set of* synchronized nodes *that are $(\Delta/2)$-close w.r.t. the working time throughout the entire process. With high probability,*

$$|\mathcal{S}| \geq n \cdot (1 - \exp(-8 \log n / \log \log n)).$$

The proof idea is as follows. We first observe that roughly $n \cdot \Big(1 - \exp\Big(-\log n / \log^2 \log n\Big)\Big)$ nodes are $(\Delta/16)$-close throughout the execution of the algorithm. As argued before, the resulting number of poorly synchronized nodes is too large and could tip the balance. Furthermore, we show, by careful induction, that thanks to the perpetual synchronization in each phase, a large fraction $f = (1 - \exp(-9 \log n / \log \log n))$ of the nodes which were $(\Delta/2)$-close throughout the first i phases, will remain $(\Delta/2)$-close in phase $i + 1$: (i) a fraction f of these nodes will tick equally often in each interval in this phase, up to an error of $\Delta/16$, and (ii) among these nodes again a fraction f will adapt their working time by selecting the median of a sample of nodes. That median will be $(\Delta/16)$-close. Accounting

for numerous other sources of error we obtain overall $(\Delta/2)$-closeness for a large fraction of nodes.

Equipped with Proposition 11.3 we analyze the 2-Choices and Bit-Propagation sub-phases. Instead of describing the distribution of colors after every 2-Choices and Bit-Propagation sub-phase, we restrict ourselves to the distribution of colors among the well-synchronized nodes in $\mathcal{S}$. In fact, throughout the analysis, we assume for all other nodes in $(V \setminus \mathcal{S})$ the worst-case. However, based on the Sync Gadget and Proposition 11.3, their number is small enough such to prevent them from tipping the balance.

Our next key-lemma is Proposition 11.4 which establishes that the number of nodes which pick up a bit for color $\mathcal{C}_j$ is with high probability concentrated around the expectation.

Analogously to the synchronous case, we consider in the following definitions and propositions an arbitrary but fixed phase of Algorithm 10. Let $\hat{c}_j(\tau)$ denote the number of nodes belonging to $\mathcal{S}$ having color $\mathcal{C}_j$ at reference point τ, that is, at time step $\tau \cdot n$. Let furthermore $x_j(\tau)$ denote the set of nodes belonging to $\mathcal{S}$ having color $\mathcal{C}_j$ and having their bit set at reference point τ and let finally $x(\tau) = \sum_j x_j(\tau)$.

Proposition 11.4. *Assume $|\mathcal{S}| \geq n \cdot (1 - \exp(-8 \log n / \log\log n))$. Let $\mathcal{C}_j$ be an arbitrary but fixed color. With high probability, the number of nodes in $\mathcal{S}$ having a bit set for color $\mathcal{C}_j$ after the 2-Choices sub-phase at reference point τ_{BP1} is bounded as follows.*

$$x_1(\tau_{\text{BP1}}) \geq \frac{\hat{c}_j(\tau_0)^2}{n}(1 - \mathrm{o}(1)) \quad \textit{and} \quad x_i(\tau_{\text{BP1}}) \leq \frac{\hat{c}_j(\tau_0)^2}{n}(1 + \mathrm{o}(1)) + \mathrm{O}\Big(n^{1-14/\log\log n}\Big) \ .$$

Building on the concentration of bits given by Proposition 11.4 at τ_{BP1}, the following proposition bounds the number of nodes of each color after the Bit-Propagation sub-phase at τ_{BP2}. As before, we only characterize those nodes which are part of $\mathcal{S}$.

Proposition 11.5. *Assume $|\mathcal{S}| \geq n \cdot (1 - \exp(-8 \log n / \log\log n))$. Let $\mathcal{C}_j$ be an arbitrary but fixed color. With high probability, the number of nodes in $\mathcal{S}$ of color $\mathcal{C}_j$ after the Bit-Propagation sub-phase is bounded as follows.*

$$\hat{c}_1(\tau_{\text{BP2}}) \geq \frac{\hat{c}_1(\tau_0)^2}{x(\tau_{\text{BP1}})} \cdot (1 - \mathrm{o}(1)) \quad \textit{and} \quad \hat{c}_j(\tau_{\text{BP2}}) \leq \frac{\hat{c}_j(\tau_0)^2}{x(\tau_{\text{BP1}})} \cdot (1 + \mathrm{o}(1)) + \mathrm{O}\Big(n^{1-4/\log\log n}\Big) \ .$$

In the proof we analyze the Bit-Propagation by the means of the Pólya urn process. In particular, we show that the fraction of nodes supporting each color $\mathcal{C}_j$ remains concentrated throughout the Bit-Propagation sub-phase. The proofs can be found in Section 11.5, Section 11.6, and Section 11.7, respectively.

11.5 Concentration of the Clocks: Proof of Proposition 11.3

In the following we show that throughout the entire process there do not exist nodes which perform more than $\mathrm{O}(\log n)$ ticks, with high probability.

Observation 11.6. *For any reference point τ we have that the working time of any node is bounded by the minimum and maximum real times, that is, for all $u \in V$ and $\tau \in \mathbb{N}$ we have*

$$T'_v(\tau) \in \left[\min_{u\in V} T_u(\tau), \max_{u\in V} T_u(\tau)\right] \ . \tag{11.1}$$

Let $\mathfrak{T}$ denote the total number of time steps until all nodes have completed the execution of Part 1 of the asynchronous protocol defined in Algorithm 10 w.r.t. their working time. With high probability, we have

$$\mathfrak{T} \leq 3/2 \cdot \kappa \cdot \ell \cdot n \log n \ . \tag{11.2}$$

Furthermore, we have with high probability that

$$\max_{v\in V}\{T_v(\mathfrak{T})\} < 2 \cdot \kappa \cdot \ell \cdot \log n \qquad \textit{and} \qquad \max_{v\in V}\{T'_v(\mathfrak{T})\} < 2 \cdot \kappa \cdot \ell \cdot \log n \ . \tag{11.3}$$

Proof Sketch. The proof idea is the following. (11.1) follows from the fact that at every tick the working time and the real time are simultaneously increased by one, and whenever the working time is set to the median of the sampled real times, which are also incremented upon each tick, the property also holds. For the proof of (11.2) and (11.3), observe that according to Algorithm 10 a node completes the execution of the algorithm when T'_v reaches $\kappa \cdot \ell \cdot \log n$. The proof of (11.2) and (11.3) follows, for $\kappa \cdot \ell$ large enough, from an application of Chernoff bounds to $T_v(\mathfrak{T})$ and union bound over all nodes, where we use (11.1) to show the second part of (11.3). □

We proceed to show that *most* nodes are *almost synchronous* at carefully chosen reference points. Intuitively, a huge fraction of nodes has a number of ticks that is concentrated around the expected value and therefore most nodes will execute instructions which are *close* together. We formalize this concept in the following lemma which is based on Definition 11.2. The lemma establishes in its first part that $n \cdot \left(1 - \exp\left(-\Theta\left(\log n / \log^2 \log n\right)\right)\right)$ nodes will be $(\Delta/6)$-close w.r.t. the real time over the course of the algorithm.

In the second statement we consider shorter intervals of the length of a phase and claim that a much larger number of nodes, to be specific, $n \cdot (1 - \exp(-9 \log n / \log \log n))$ nodes, will be selected to tick for the same number of times up to an error of $\Delta/16$.

Lemma 11.7. *Let $\Delta \geq c_\Delta \log n / \log \log n$, for some large enough constant c_Δ. Let τ be a reference point with $\tau \leq c \cdot \log n$, and let $Y(\tau)$ be the random variable for the number of nodes which are $(\Delta/16)$-close to τ w.r.t. T_v. We have*

$$Y(\tau) \geq n \cdot \left(1 - \exp\left(-\Omega\left(\log n / \log^2 \log n\right)\right)\right) \ .$$

Furthermore, consider an arbitrary interval consisting of t consecutive ticks. Fix a subset $Y \subseteq V$ and let $Y' \subset Y$ be the subset of nodes which receive at least $t/n - \Delta/16$ ticks and at

most $t/n + \Delta/16$ ticks out of the t ticks. We have

$$|Y'| \geq |Y| \cdot (1 - \exp(-10 \log n / \log\log n)) - \tilde{O}(\sqrt{n}) \ .$$

Proof. Let $\mathcal{E}_v(\tau)$ be the event that a node v is $(\Delta/16)$-close to τ, that is,

$$\mathcal{E}_v(\tau) = \left[\tau - \Delta/16 \leq \ T_v(\tau) \ \leq \ \tau + \Delta/16\right] \ .$$

We apply Chernoff bounds to $T_v(t)$ and obtain

$$\mathbb{P}[\,\mathcal{E}_v(\tau)\,] \geq 1 - \exp\Big(-\Omega\Big(\frac{\log n}{\log^2 \log n}\Big)\Big) \ , \tag{11.4}$$

Let in the following $Y_v(\tau)$ be an indicator random variable for a node v and a reference point τ defined as

$$Y_v(\tau) = \begin{cases} 1, & \text{if } \mathcal{E}_v(\tau) \ , \\ 0, & \text{otherwise.} \end{cases}$$

Summing up over all nodes gives us $Y(\tau) = \sum_{v \in V} Y_v(\tau)$. By linearity of expectation, we have $\mathbb{E}[\,Y(\tau)\,] \geq n \cdot \Big(1 - \exp\Big(-\Theta\Big(\log n/(\log^2 \log n)\Big)\Big)\Big)$. Note that the random variables $T_v(\tau)$, and therefore also the random variables $Y_v(\tau)$, are not independent. We thus consider the process of uncovering $Y_v(\tau)$ one node after the other in order to obtain the Doob martingale of $Y(\tau)$ as follows. We define the sequence $Z_j(\tau)$ as $Z_j(\tau) = \mathbb{E}[\,Y(\tau)|T_j(\tau), \ldots, T_1(\tau)\,]$ with $Z_0(\tau) = \mathbb{E}[\,Y(\tau)\,]$. We have

$$\mathbb{E}[\,Z_j(\tau)|T_{j-1}(\tau), \ldots, T_1(\tau)\,] = \mathbb{E}[\,\mathbb{E}[\,Y(\tau)|T_j(\tau), \ldots, T_1(\tau)\,]|T_{j-1}(\tau), \ldots, T_1(\tau)\,]$$

which, applying the tower property, gives us that

$$\mathbb{E}[\,Z_j(\tau)|T_{j-1}(\tau), \ldots, T_1(\tau)\,] = \mathbb{E}[\,Y(\tau)|T_{j-1}(\tau), \ldots, T_1(\tau)\,] = Z_{j-1}(\tau) \ .$$

Therefore $Z_j(\tau)$ is indeed the Doob martingale of $Y(\tau)$.

According to Observation 11.6 each node ticks at most $2c \cdot \log n$ times, that is, $|T_{j+1}(\tau) - T_j(\tau)| \leq 2c \cdot \log n$. This holds with high probability in the original process P and with probability 1 in the coupled process P'. Since at most $2c \cdot \log n$ of the random variables $Y_{j+1}(\tau), \ldots, Y_n(\tau)$ differ, we have

$$\begin{aligned} |Z_{j+1}(\tau) - Z_j(\tau)| = \big|&\mathbb{E}[\,Y_n(\tau) + \cdots + Y_1(\tau)|T_{j+1}(\tau), \ldots, T_1(\tau)\,] \\ &- \mathbb{E}[\,Y_n(\tau) + \cdots + Y_1(\tau)|T_j(\tau), \ldots, T_1(\tau)\,]\big| \leq 2c \cdot \log n \ . \end{aligned}$$

Applying the Azuma-Hoeffding bound to $Y(\tau) = \sum_{v \in V} Y_v(\tau)$ gives us

$$\mathbb{P}\Big[\, |Y(\tau) - \mathbb{E}[\, Y(\tau)\,]| \geq \sqrt{c^3 \cdot n \cdot \log^3 n}\,\Big] \leq \exp\left(-\frac{c^3 \cdot n \cdot \log^3 n}{2 \cdot \sum_{j=1}^{n} (2c \cdot \log n)^2}\right) ,$$

which for sufficiently large c yields $|Y(\tau) - \mathbb{E}[\, Y(\tau)\,]| \leq \sqrt{c^3 \cdot n \cdot \log^3 n}$ with high probability. Observe that $\sqrt{c^3 \cdot n \cdot \log^3 n} \leq n \cdot \exp\Big(-\Theta\Big(\log n / \log^2 \log n\Big)\Big)$. We finally conclude that, with high probability, at least $n \cdot \Big(1 - \exp\Big(-\Theta\Big(\log n / \log^2 \log n\Big)\Big)\Big)$ nodes are synchronous up to a deviation of at most $\Delta = \Theta(\log n / \log \log n)$ ticks from the expected number of ticks at the given reference point τ.

We now turn to the second part of the statement. Recall that $\Delta = c_\Delta \log n / \log \log n$ and c_Δ is a large enough constant. Observe that, by definition of our algorithm, $T = 10\Delta$. The proof of the second part follows in a similar way as before. We define an analogous event $\mathcal{E}'_v(\tau_1)$ for node v to hold, then the number of ticks it receives $t/n \pm \Delta/16$ out of t ticks. We have

$$\mathbb{P}[\, \mathcal{E}'_v(\tau)\,] \geq 1 - \exp\left(-\frac{10 \log n}{\log \log n}\right).$$

Observe that this is bound is much stronger than (11.4). Similarly, as before, $|Y' - \mathbb{E}[\, Y'\,]| \leq \sqrt{c^3 \cdot n \cdot \log^3 n}$ with high probability. Thus,

$$|Y'(\tau_1)| \geq |Y| \cdot (1 - \exp(-10 \log n / \log \log n)) - \sqrt{c^3 \cdot n \cdot \log^3 n}$$

yielding the claim. □

In the following we show that the median taken will be concentrated around the expected real time.

Lemma 11.8. *The median real-time of a uniform sample of $\Omega(\log^2 \log n)$ nodes is $(\Delta/16)$-close with high probability at any reference point $\tau \leq \kappa \cdot \ell \cdot \log n$.*

Proof. In this proof we assume for simplicity that the $c'' \log^2 \log n$ sampled nodes are taken in one single step. First, we show that the median of the sampled times is close to the average of all (real) times, with high probability. The median real-time of the sample is no $(\Delta/16)$-close if at least half of the sample contained nodes which were not $(\Delta/16)$-close. By Lemma 11.7, we know that for some constant $c > 0$ there are with high probability at most

$$L = n \exp\Big(-c(\log n / \log^2 \log n\Big))$$

nodes u which are not $(\Delta/16)$-close w.r.t. T_u during any point of the execution of the algorithm.

Let $\mathcal{G}$ be the set of these *bad* nodes. Let Z denote the number of samples drawn which are bad. Thus, by Theorem A.4 with parameters $\alpha = 1/2$ and $p = L/n$, we derive

$$\begin{aligned}
\mathbb{P}\Big[\, Z \geq \alpha c'' \log^2 \log n/2 \,\Big] &\leq \Big((2p)^{1/2}(2(1-p))^{1/2}\Big)^{c'' \log^2 \log n} \\
&\leq 2^{c'' \log^2 \log n} \cdot (p^{1/2}(1-p))^{c'' \log^2 \log n} \\
&\leq \sqrt{n} \cdot (L/n)^{c'' \log^2 \log n/2} \\
&= \sqrt{n} \cdot n^{-\frac{c \cdot c'' \log^2 \log n}{2 \log^2 \log n}} \\
&\leq 1/n^2,
\end{aligned}$$

for large enough c''. □

Proof of Proposition 11.3. For every phase $s = \mathrm{O}(\log \log n)$, let J_s be the set of nodes which are

1. $(5\Delta/16)$-close w.r.t. the working time at any reference point $\tau = s \cdot T$ and
2. $(\Delta/2)$-close w.r.t. the working time at any reference point in $[(s-1) \cdot T, s \cdot T]$.

In the following, we show by induction that with high probability

$$|J_s| \geq n\Big(1 - T^2 \cdot s \cdot \exp(-9 \log n / \log \log n)\Big) \ .$$

For $s = 0$ this holds trivially since $|J_0| = n$. Suppose the claims holds for phase s and consider phase $s+1$. We seek to show that the claim holds in the interval $[s \cdot T, (s+1) \cdot T]$. Let $\tau_{\mathrm{L}}, \tau_{\mathrm{R}}$ with $\tau_{\mathrm{L}} < \tau_{\mathrm{R}}$ be an arbitrary pair of reference points with $\tau_{\mathrm{L}} \geq s \cdot T$ and $\tau_{\mathrm{R}} \leq (s+1) \cdot T$. Let furthermore $J' \subset J_s$ denote the set of nodes which are selected to tick $\tau_{\mathrm{R}} - \tau_{\mathrm{L}} \pm \Delta/16$ times in any interval $[\tau_{\mathrm{L}}, \tau_{\mathrm{R}}]$. By Part 2 of Lemma 11.7, we have

$$|J'| \geq |J_s|(1 - \exp(-9 \log n / \log \log n)) \ . \tag{11.5}$$

Let J'_s be the set of nodes which are selected $\tau_{\mathrm{R}} - \tau_{\mathrm{L}} \pm \Delta/16$ times to tick in *every* interval $[\tau_{\mathrm{L}}, \tau_{\mathrm{R}}]$. Since there are at most T^2 such intervals, we get by (11.5) that with high probability

$$|J'_s| \geq |J_s|\Big(1 - T^2 \cdot \exp(-9 \log n / \log \log n)\Big) \ .$$

Let v be an arbitrary but fixed node. Let ϑ_v be the exact time step at which v jumps and observe that ϑ_v is a random variable. Let furthermore τ_v denote the first reference point after time step ϑ_v, that is, $\tau_v = \lceil \vartheta_v / n \rceil$. Consider the number of times v is selected to tick in the interval of time steps $[\vartheta_v, \tau_v \cdot n]$. By a standard balls-into-bins argument [RS98], we can argue that with high probability

$$|T(\tau_v) - T(\vartheta_v/n)| \leq \Delta/16 \ . \tag{11.6}$$

Let τ' be any reference point in $[\tau_v, (s+1)\cdot T]$. Since the working time increases afterwards whenever v is selected to tick, we have

$$T'_v(\tau') - T'_v(\vartheta_v/n) = T_v(\tau') - T_v(\vartheta_v/n) \ . \tag{11.7}$$

We now show that every node $v \in J'_s$ jumps exactly once. Recall that τ_{JUMP} is the instruction at which every node executes the jump step. That is, if any nodes has a working time of $s \cdot T + \tau_{\text{JUMP}}$, then that node jumps We claim that every node $v \in J'_s$ must have jumped prior to $(s+1)\cdot T$, that is, we have $\tau_v \leq (s+1)\cdot T$. To see this, assume that v didn't jump. By (11.7),

$$\begin{aligned} T'_v((s+1)\cdot T) &= T_v((s+1)\cdot T) - T_v(s\cdot T) + T'_v(s\cdot T) \\ &\geq (s+1)\cdot T - s\cdot T - \Delta/16 + T'_v(s\cdot T) \\ &\geq (s+1)\cdot T - s\cdot T - \Delta/16 + s\cdot T - 5\Delta/16 \\ &> (s+1)\cdot T - \Delta/2 \geq s\cdot T + \tau_{\text{JUMP}} \ , \end{aligned}$$

where the first inequality follows from the definition of J'_s and the second inequality follows from the induction hypothesis. The the above inequality implies that v must have executed the jump instruction and thus must have jumped.

Symmetrically, we claim that every node $v \in J'_s$ will jump at most once per phase with high probability. It suffices to show that no node of J'_s jumps before reference point $\tau' := \tau_{\text{M2}} + \Delta/2$, since, informally speaking, at reference point τ' all nodes of J'_s will have a real time exceeding τ_{M2} (similarly as before, this can be shown using the definition of J'_s and the induction hypothesis). Thus, by Lemma 11.8 and the due to the immense size of J'_s, node v will set its working time to the median of sampled real times which will be larger than τ_{M2}. Node v will not execute the jump instruction again in this phase. To show this claim we need to show that $T'_v(\tau') < s\cdot T + \tau_{\text{JUMP}}$, which is true since (11.7),

$$\begin{aligned} T'_v(\tau') &= T_v(\tau') - T_v(s\cdot T) + T'_v(s\cdot T) \\ &\leq \tau_{\text{M2}} + \Delta/2 + \Delta/16 + T'_v(s\cdot T) \\ &\leq \tau_{\text{M2}} + \Delta/2 + \Delta/16 + s\cdot T - 5\Delta/16 \\ &\leq (s+1)\cdot T - \Delta/2 = s\cdot T + \tau_{\text{JUMP}} \ , \end{aligned}$$

where the first inequality follows from the definition of J'_s and the second inequality follows from the induction hypothesis. Thus, v jumped at most once. We therefore conclude that every node $v \in J'_s$ jumps exactly once.

We will now argue the following. For every $v \in J'_s$ chooses with high probability

$$|T'_v(\vartheta_v/n) - \vartheta_v/n| \leq 2\Delta/16 + 1 \ . \tag{11.8}$$

To see this, first observe that, by Lemma 11.8, the median taken from $\log^3 \log n$ samples of the real time is $(\Delta/6)$-close. Second, we need to account for the fact that median is not taken directly, but rather over time. If all samples were taken directly before jumping, then the median would indeed be $(\Delta/6)$-close. However, since $v \in J'_s$, it holds that the value of any sample is $(\Delta/6)$-close w.r.t. the value it would have if it were sampled directly before v jumps. Accounting for all errors, using triangle inequality and that $\tau_u = \lceil \vartheta_v/n \rceil$, (11.8) follows.

We proceed by showing that after $v \in J'_s$ jumps its working-time well-concentrated, that is,

$$|T'_v(\tau') - \tau'| \leq 5\Delta/16 \ , \tag{11.9}$$

for any reference point τ' in $[\tau_v, (s+1) \cdot T]$. We have

$$\begin{aligned}
T'_v(\tau') &\overset{(11.7)}{=} T'_v(\vartheta_v/n) + T_v(\tau') - T_v(\vartheta_v/n) \\
&\overset{(11.8)}{\leq} \vartheta_v/n + 2\Delta/16 + 1 + T_v(\tau') - T_v(\vartheta_v/n) \\
&\overset{(11.6)}{\leq} \vartheta_v/n + 2\Delta/16 + 1 + T_v(\tau') - T_v(\tau_v) + \Delta/16 \\
&\overset{\text{def. } J'_s}{\leq} \vartheta_v/n + 2\Delta/16 + 1 + ((\tau' - \tau_v) + \Delta/16) + \Delta/16 \\
&\overset{\text{def. } \tau_v}{\leq} \tau_v + 1 + 2\Delta/16 + 1 + ((\tau' - \tau_v) + \Delta/16) + \Delta/16 \\
&\leq \tau' + 5\Delta/16,
\end{aligned}$$

Symmetrically, we have

$$\begin{aligned}
T'_v(\tau') &\overset{(11.7)}{=} T'_v(\vartheta_v/n) + T_v(\tau') - T_v(\vartheta_v/n) \\
&\overset{(11.8)}{\geq} \vartheta_v/n - 2\Delta/16 - 1 + T_v(\tau') - T_v(\vartheta_v/n) \\
&\overset{(11.6)}{\geq} \vartheta_v/n - 2\Delta/16 - 1 + T_v(\tau') - T_v(\tau_v) - \Delta/16 \\
&\overset{\text{def. } J'_s}{\geq} \vartheta_v/n - 2\Delta/16 - 1 + ((\tau' - \tau_v) + \Delta/16) - \Delta/16 \\
&\overset{\text{def. } \tau_v}{\geq} \tau_v - 1 - 2\Delta/16 - 1 + ((\tau' - \tau_v) + \Delta/16) - \Delta/16 \\
&\geq \tau' - 5\Delta/16 \ ,
\end{aligned}$$

This shows (11.9). Define $J_{s+1} = J'_s$. This shows that $v \in J_{s+1}$ is $(5\Delta/16)$-close at $(s+1) \cdot T$. Furthermore, at reference point $s \cdot T$, v was, by induction hypothesis, $(5\Delta/16)$-close and, since $J_{s+1} = J'_s$, at every reference point τ before u jumped we can derive $|T'_v(\tau) - \tau| \leq 5\Delta/16 + \Delta/16 \leq \Delta/2$. Furthermore, (11.9) implies that v was also $(\Delta/2)$-close after jumping and thus v was $\Delta/2$ at each reference point in $[s \cdot T, (s+1) \cdot T]$.

We now show that $|J_{s+1}|$ is large enough. Using the induction hypothesis, we have

$$\begin{aligned}|J_{s+1}| = |J'_s| &\geq |J_s|\Big(1 - T^2 \cdot \exp(-9\log n/\log\log n)\Big) \\ &\geq n\Big(1 - sT^2 \cdot \exp(-9\log n/\log\log n)\Big)\Big(1 - T^2 \cdot \exp(-9\log n/\log\log n)\Big) \\ &\geq n\Big(1 - (s+1)T^2 \cdot \exp(-9\log n/\log\log n)\Big)\ .\end{aligned}$$

This finishes the induction step. Finally, observe that for any $s = \mathrm{O}(\log\log n)$ we have

$$n \cdot \Big(1 - s \cdot T^2 \cdot \exp(-9\log n/\log\log n)\Big) \geq n(1 - \exp(-8\log n/\log\log n))\ . \qquad \square$$

11.6 Analysis of the 2-Choices sub-phase: Proof of Proposition 11.4

Proof of Proposition 11.4. Recall that $\mathcal{S}$ is the set of nodes v that are $(\Delta/2)$-close w.r.t. $T'(v)$ throughout the entire process. By Proposition 11.3, $|\mathcal{S}| \geq n - \mathcal{E}$, with $\mathcal{E} \leq n \cdot \exp(-8\log n/\log\log n)) = n^{1-8/\log\log n}$. When a node of $\mathcal{S}$ samples two nodes, then by definition the working time of all nodes of $\mathcal{S}$ is larger than τ_0 and smaller than τ_{set}. Let u be a node of $\mathcal{S}$. Then, u samples at two nodes (that is, when its working time is τ_{TC}), then its probability of sampling two nodes of color $\mathcal{C}_j$ with probability at least $(\hat{c}_j(\tau_0)/n)^2$ and at most $((\hat{c}_j(\tau_0) + \mathcal{E})/n)^2$.

By Chernoff bounds,

$$x_1(\tau_{\text{BP1}}) \geq |\mathcal{S}| \cdot (\hat{c}_j(\tau_0)/n)^2 - \sqrt{n}\log n \geq \frac{\hat{c}_j(\tau_0)^2}{n}(1 - o(1)),$$

where we used the fact that all nodes of $\mathcal{S}$ must have executed the instruction at τ_{set} at reference point τ_{BP1}.

We now distinguish between two cases. If $\hat{c}_j(\tau_0) \leq n^{1-7/\log\log n}$ we have, $\hat{c}_j(\tau_0) + \mathcal{E} = \mathrm{O}\Big(n^{1-7/\log\log n}\Big)$. Thus, by Chernoff bounds, with high probability

$$x_j(\tau_{\text{BP1}}) \leq n \cdot ((\hat{c}_j(\tau_0) + \mathcal{E})/n)^2 + \sqrt{n}\log n = \mathrm{O}\Big(n^{1-14/\log\log n}\Big).$$

Otherwise, $\hat{c}_j(\tau_{\text{BP1}}) > n^{1-7/\log\log n}$ and we have $\hat{c}_j(\tau_0) + \mathcal{E} = \hat{c}_j(\tau_0)(1 + o(1))$. Thus, by Chernoff bounds, we obtain with high probability that

$$x_j(\tau_{\text{BP1}}) \leq n \cdot ((\hat{c}_j(\tau_0) + \mathcal{E})/n)^2 = \hat{c}_j(\tau_0)^2/n \cdot (1 + o(1)).$$

This finishes the proof. $\square$

11.7 Analysis of the Bit-Propagation Sub-Phase: Proof of Proposition 11.5

We now focus on the analysis of the Bit-Propagation sub-phase. Similar to the analysis of the synchronous case, we first analyze the number of bits which are set during the Bit-Propagation sub-phase without taking their color into consideration. The following lemma is based on the observation that the Bit-Propagation can be modeled by a simple asynchronous randomized-gossip-based information dissemination process.

Lemma 11.9. *Consider an arbitrary but fixed phase and let $x(\tau)$ be the number of nodes in $\mathcal{S}$ which have a bit set at reference point τ in that phase. Assume that $|\mathcal{S}| \geq n \cdot (1 - \exp(-8 \log n / \log\log n))$ and that $x(\tau_{\text{BP1}}) \geq n/(2k)$. Then we have $x(\tau_{\text{BP2}}) = |\mathcal{S}|$ with high probability.*

Proof. We split the proof into three parts, in each of which we will rely on the fact that at each reference point the nodes of $\mathcal{S}$ are $(\Delta/2)$-close. We argue that with high probability (i) $x(\tau_2') \geq n/2$, (ii) $x(\tau_4) \geq |S| \cdot \left(1 - n^{-2/\log\log n}\right)$, and (iii) $x(\tau_{\text{BP2}}) = |\mathcal{S}|$.

Part (i). To show the first part, we first consider a sequence of Δ periods from τ_1' to τ_2'. Recall that each period consists of n consecutive time steps. We will show by induction over $i \in [\tau_1', \tau_2')$ that

$$x(i) \geq \min\left\{\frac{n}{2},\ \frac{n}{2k} \cdot \left(1 + \frac{1}{5}\right)^i\right\} .$$

Let i be an arbitrary but fixed period in $[\tau_1', \tau_2')$ and assume that $x(i-1) < n/2$. Note that by definition of $\mathcal{S}$ at any reference point $\tau \in [\tau_1', \tau_2']$ all nodes of $\mathcal{S}$ are in $[\tau_1, \tau_3]$. Let $H(i) \subseteq \mathcal{S}$ be the set of nodes in $\mathcal{S}$ which did not have their bit set after period $i-1$. By assumption, $|H(i)| \geq |\mathcal{S}| - n/2 = n/2 \cdot (1 - \mathrm{o}(1))$. Let furthermore $A(i)$ be the set of *active* nodes which tick in period i at least once. By a standard balls-into-bins arguments [RS98], we have that $|A(i)|$ has size at least $n/2$ with high probability. Observe that each node is equally likely to tick, independently of whether the bit is set or not. Therefore, $A(i)$ and $H(i)$ are independent, and any node in $H(i)$ ticks at least once with probability at least $n/2$, independently. Hence, $|A(i) \cap H(i)| \geq n/4 \cdot (1 - \mathrm{o}(1))$ with high probability, where the concentration follows from Chernoff bounds.

For a node $v \in A(i) \cap H(i)$ in period i, we define X_v to be the indicator random variable for the event that v sets the bit. Note that all X_v are independent and $\mathbb{P}[\, X_v = 1\,] \geq x(i-1)/n$. Let $X = \sum X_i$. By Chernoff bounds, $X \geq |A(i) \cap H(i)| \cdot x(i-1)/n \cdot (1 - \mathrm{o}(1)) \geq x(i-1)/5$ with high probability. We therefore get that with high probability

$$x(i) \geq x(i-1) + X \geq x(i-1)\left(1 + \frac{1}{5}\right) \overset{\text{IH}}{\geq} \frac{n}{2k} \cdot \left(1 + \frac{1}{5}\right)^i ,$$

which completes the induction. We now obtain, using $\tau'_2 - \tau'_1 \geq 4 \log k$, that

$$x(\tau'_2) \geq \frac{n}{2k}\left(1 + \frac{1}{5}\right)^{\tau'_2 - \tau'_1} \geq \frac{n}{2k} \cdot k = n/2 \ .$$

This completes the proof of Part (i).

Part (ii). Let $H(\tau'_2) \subseteq \mathcal{S}$ be the set of nodes in $\mathcal{S}$ which do not have a bit set at reference point τ'_2. We consider an arbitrary but fixed node $v \in H(\tau'_2)$ at reference point τ_4. Since v is in $\mathcal{S}$ and thus $(\Delta/2)$-close at both, τ'_2 and τ_4, we observe that it ticked at least $\tau_4 - \tau'_2 - 2 \cdot \Delta/2 = \Delta/2$ times between time steps $\tau'_2 \cdot n$ and $\tau_4 \cdot n$ corresponding to these reference points. The probability that the node v never sampled a node with the bit set is thus at most $2^{-\Delta/2}$. Hence, by using independence and Chernoff bounds, the number of nodes remaining in $H(\tau_4)$ is, for Δ large enough, at most $|\mathcal{S}| \cdot n^{-2/\log\log n}$ with high probability.

Part (iii). As before, let $H(\tau_4) \subseteq \mathcal{S}$ be the set of nodes in $\mathcal{S}$ which do not have a bit set at reference point τ_4. We again consider an arbitrary but fixed node $v \in H(\tau_4)$. Since v is in $\mathcal{S}$ and thus $(\Delta/2)$-close at both, τ_4 and τ_{BP2}, we observe that it performed at least $\tau_5 - \tau'_4 = \Delta/2$ Bit-Propagation ticks. The probability that v samples in one of these ticks a node in $\mathcal{S}$ without the bit set or that v samples a node not in $\mathcal{S}$ is at most $n^{-2/\log\log n} + n^{-8/\log\log n} \leq n^{-1/\log\log n}$. Therefore, the probability that this node never obtains the bit is at most $\left(n^{-1/\log\log n}\right)^{\Delta/2} \leq n^{-\omega(1)}$. From union bound we derive that all nodes in $\mathcal{S}$ therefore have the bit set at reference point τ_{BP2}. □

In the following we analyze the individual colors during the Bit-Propagation sub-phase. Our main observation is that the Bit-Propagation process can be modeled by so-called Pólya urns [JK77]. In this model, we are given an urn containing marbles of two colors, black and white. In every step, one marble is drawn uniformly at random from the urn. Its color is observed, the marble is returned to the urn and one more marble of the same color is added. For any color, the ratio of marbles with that given color over the total number of marbles is a martingale. We will use this urn process to model the Bit-Propagation sub-phase, which then can be analyzed by means of martingale techniques. Formally, the Pólya urn process is defined as follows.

Definition 11.10 (Pólya Urn Process)**.** *Let* Pólya(α_1, α_2) *with* $\alpha_1, \alpha_2 \in \mathbb{Z}_0^+$ *be the following urn process. At the beginning there are* α_1 *black marbles and* α_2 *white marbles in the urn. The process runs in multiple steps where* $\alpha_1(i)$ *and* $\alpha_2(i)$ *denote the number of black and white marbles in the urn, respectively, for every time step* i*. In every time step* i*, a black marble is added with probability* $\alpha_1(i)/(\alpha_1(i) + \alpha_2(i))$*, and with remaining probability* $\alpha_2(i)/(\alpha_1(i) + \alpha_2(i))$ *a white marble is added.*

We now use this urn model to show our main result for the Bit-Propagation sub-phase, Proposition 11.5. We start by performing a worst-case analysis for color 1 in order to give a lower bound on the number of nodes of color 1 after the Bit-Propagation sub-phase. Similarly, we will upper bound any *large* color $\mathcal{C}_j$. Then we will show that after each phase the gap between color 1 and $\mathcal{C}_j$ grows quadratically. We will use bounds resulting from Proposition 11.4 for the numbers of nodes with bits and their color distribution among $\mathcal{S}$. For the worst-case analysis, we will assume that any node which is not in $\mathcal{S}$ has color $\mathcal{C}_j$ and its bit set. We now give the formal proof.

Proof of Proposition 11.5. We consider an arbitrary but fixed Bit-Propagation sub-phase which we model by Pólya(α_1, α_2) as follows. Initially, we place for each node in $\mathcal{S}$ of color 1 which has its bit set at reference point τ_{BP1} a black marble in the urn, that is, $\alpha_1 = x_1(\tau_{\text{BP1}})$. Additionally, we add for each node in $\mathcal{S}$ which has its bit set for any color $\mathcal{C}_j \neq \mathcal{C}_1$ a white marble in the urn. Finally, in order to perform a worst-case analysis, we add a white marble for any node which is not in $\mathcal{S}$, that is, we add an additional number of $|V \setminus \mathcal{S}|$ white marbles. We therefore have $\alpha_1 + \alpha_2 = x(\tau_{\text{BP1}}) + |V \setminus \mathcal{S}|$. We now consider only those time steps of the Bit-Propagation sub-phase, where a node in $\mathcal{S}$ without bit samples another node with bit. We couple these very steps with the Pólya urn process, where we assume that a marble is added based on the adopted color in the Bit-Propagation process, that is, if a node newly adopts a bit for color 1, we add a black marble, and if otherwise a node adopts a bit for color $\mathcal{C}_j \neq \mathcal{C}_1$, we add a white marble. For the worst-case analysis we assume in the Bit-Propagation process that all nodes in $V \setminus \mathcal{S}$ have a bit set for a color $\mathcal{C}_j \neq \mathcal{C}_1$ throughout the entire process. This corresponds to the additional $|V \setminus \mathcal{S}|$ white marbles initially added to the urn.

As before, we will use the notation that $x(\tau)$ denotes the number of nodes in $\mathcal{S}$ which have a bit set at reference point τ and $x_j(\tau)$ denotes the number of nodes in $\mathcal{S}$ of color $\mathcal{C}_j$ which have a bit set at reference point τ. Let M be a lower bound on $x(\tau_{\text{BP1}})$, the number of bits set at the beginning of the Bit-Propagation sub-phase, and recall that according to the proof of Proposition 11.4 we have with high probability

$$M \geq n/(2k) \ . \tag{11.10}$$

We now consider the Pólya urn process. Let $F(i)$ be the fraction of black marbles in step i of the Pólya urn process. As mentioned before, this fraction of black marbles in the Pólya urn process is a martingale. Observe furthermore that $|F(i) - F(i-1)| \leq 1/M$ throughout the entire urn process. Let $\mathcal{I}$ be the last step of the Pólya urn process and observe that

$\mathcal{I} \leq n$. Applying Azuma's inequality to $F(i)$ for any $i \leq \mathcal{I}$ gives us

$$\begin{aligned}\mathbb{P}[\,|F(i) - F(1)| \geq \delta\,] &\leq 2 \cdot \exp\left(-\frac{\delta^2}{2 \cdot \sum_{j=1}^{i} 1/M^2}\right) \\ &\leq 2 \cdot \exp\left(-\frac{\delta^2 \cdot M^2}{2 \cdot i}\right) \ .\end{aligned}$$

We set $\delta = 4 \cdot k \cdot \sqrt{\log n/n}$ and obtain using (11.10)

$$\begin{aligned}\mathbb{P}\left[\,|F(i) - F(1)| \geq 4 \cdot k \cdot \sqrt{\log n/n}\,\right] &\leq 2 \cdot \exp\left(-\frac{2 \cdot k^2 \cdot M^2 \cdot \log n}{n \cdot i}\right) \\ &\leq 2 \cdot \exp(-2 \cdot \log n) \ , \end{aligned} \tag{11.11}$$

where we used that $x(\tau_{\text{BP1}}) \geq n/(2k)$ with high probability.

From the calculation above we see that with high probability the fraction of black marbles in the urn remains concentrated around the initial value. To derive a lower bound on the absolute number of black marbles at the end of the process we first bound $F(1)$. By Proposition 11.3, we have $|V \setminus \mathcal{S}| \leq n^{1-8/\log\log n}$ and thus

$$F(1) \geq \frac{x_1(\tau_{\text{BP1}})}{x(\tau_{\text{BP1}}) + |V \setminus \mathcal{S}|} \geq \frac{x_1(\tau_{\text{BP1}})}{x(\tau_{\text{BP1}}) + n^{1-8/\log\log n}} = \frac{x_1(\tau_{\text{BP1}})}{x(\tau_{\text{BP1}})} \cdot (1 - \mathrm{o}(1)) \tag{11.12}$$

Using (11.11), we get for the end of the Bit-Propagation sub-phase that at reference point τ_{BP2} with high probability

$$\begin{aligned}F(\mathcal{I}) &\geq F(1) - 4 \cdot k \cdot \sqrt{\log n/n} = \frac{x_1(\tau_{\text{BP1}})}{x(\tau_{\text{BP1}})} \cdot (1 - \mathrm{o}(1)) - 4 \cdot n^{1/\log\log n}\sqrt{\log n/n} \\ &= \frac{x_1(\tau_{\text{BP1}})}{x(\tau_{\text{BP1}})} \cdot (1 - \mathrm{o}(1)) \ ,\end{aligned}$$

where we used that $x_1(\tau_{\text{BP1}}) \geq n/(2k^2) \geq n^{1-3/\log\log n}$ with high probability and $x(\tau_{\text{BP}}) \leq n$. Hence,

$$x_1(\tau_{\text{BP2}}) \geq x(\tau_{\text{BP2}})\frac{x_1(\tau_{\text{BP1}})}{x(\tau_{\text{BP1}})} \cdot (1 - \mathrm{o}(1)) \tag{11.13}$$

It remains to establish an upper bound on $x_j(\tau_{\text{BP2}})$ for every other large color $\mathcal{C}_j \neq \mathcal{C}_1$. We will use a symmetric argument. Let $\mathcal{C}_j \neq A$ be an arbitrary but fixed color and let $F'(i)$ be the fraction of black marbles in another Pólya urn process which we use to bound the size of color $\mathcal{C}_j$. As before, we use the black marbles to represent $\mathcal{C}_j$, the color under investigation, and the white marbles to represent all other colors $\mathcal{C}_i \neq \mathcal{C}_j$. For the worst-case analysis, we again assume that all nodes of $V \setminus \mathcal{S}$ have their bit set for color $\mathcal{C}_j$. We apply a similar

computation as before and observe, now for color $\mathcal{C}_j$, that

$$\begin{aligned} F'(1) &\leq \frac{x_j(\tau_{\text{BP1}}) + |V \setminus \mathcal{S}|}{x(\tau_{\text{BP1}}) + |V \setminus \mathcal{S}|} \leq \frac{x_j(\tau_{\text{BP1}}) + |V \setminus \mathcal{S}|}{x(\tau_{\text{BP1}})} \\ &\leq \frac{x_j(\tau_{\text{BP1}}) + n^{1-8/\log\log n}}{x(\tau_{\text{BP1}})} \\ &\leq \frac{x_j(\tau_{\text{BP1}})}{x(\tau_{\text{BP1}})} + \frac{n^{1-8/\log\log n}}{n^{1-3/\log\log n}} \\ &\leq \frac{x_j(\tau_{\text{BP1}})}{x(\tau_{\text{BP1}})} + n^{-5/\log\log n} \ . \end{aligned}$$

Again using (11.11), we get with high probability

$$\begin{aligned} F'(\mathcal{I}) &\leq F'(1) + 4 \cdot k \cdot \sqrt{\log n/n} = \frac{x_1(\tau_{\text{BP1}})}{x(\tau_{\text{BP1}})} + n^{-5/\log\log n} \quad + n^{-1/3} \\ &\leq \frac{x_1(\tau_{\text{BP1}})}{x(\tau_{\text{BP1}})} + 2n^{-5/\log\log n} \ . \end{aligned}$$

Thus, using that $x(\tau_{\text{BP2}})/x(\tau_{\text{BP2}}) \leq 2k$ with high probability we get

$$\begin{aligned} x_j(\tau_{\text{BP2}}) &\leq x(\tau_{\text{BP2}}) \cdot \frac{x_j(\tau_{\text{BP1}})}{x(\tau_{\text{BP1}})} + 2n^{-1/\log\log n} \cdot 2n^{-5/\log\log n} \\ &= x(\tau_{\text{BP2}}) \cdot \frac{x_j(\tau_{\text{BP1}})}{x(\tau_{\text{BP1}})} + 4n^{-5/\log\log n} \ . \end{aligned}$$

Furthermore, from the calculation above and (11.13) we obtain for all $\mathcal{C}_j$ that with high probability

$$x_j(\tau_{\text{BP2}}) = x_j(\tau_{\text{BP1}}) \cdot \frac{x(\tau_{\text{BP2}})}{x(\tau_{\text{BP1}})} \cdot (1 \pm \mathrm{o}(1)) + \mathrm{O}\Big(n^{-5/\log\log n}\Big) \ .$$

By Proposition 11.4, we have that with high probability

$$x_j(\tau_{\text{BP1}}) = \frac{\hat{c}_j(\tau_{set})^2}{n}(1 \pm \mathrm{o}(1)) + \mathrm{O}\Big(n^{1-5/\log\log n}\Big) \ .$$

Moreover, by Lemma 11.9 and Definition 11.2, we have

$$x(\tau_{\text{BP2}}) \in [n \cdot (1 - \mathrm{o}(1)), n] \ .$$

Putting everything together, we derive that with high probability

$$x_j(\tau_{\text{BP2}}) = \frac{\hat{c}_j(\tau_0)^2}{x(\tau_{\text{BP1}})}(1 \pm \mathrm{o}(1)) + \mathrm{O}\Big(n^{1-4/\log\log n}\Big) \ . \qquad \square$$

Algorithm 11: Part 2 of the asynchronous protocol to solve plurality consensus. At ticks in $[\tau_{\text{END0}}, \tau_{\text{END1}}]$, the nodes do not perform any action.

Algorithm `asynchronous`(*node* v) **(Part 2)**

```
if τ_END1 ≤ workingtime(v) ≤ τ_END4 then
    let u_1, u_2 ∈ N(v) uniformly at random;
    if color(u_1) = color(u_2) then
        color(v) ← color(u_1);
workingtime(v) ← workingtime(v) + 1;
```

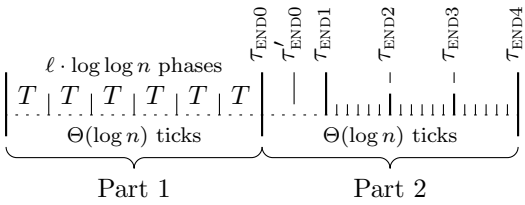


Figure 3: graphical representation of the asynchronous protocol, showing Part 1 (Algorithm 10) and Part 2 (Algorithm 11)

11.8 The Endgame: Taking a from $(1-\varepsilon_{\text{Part1}})\cdot n$ to n

In this section we analyze Part 2 of the asynchronous algorithm defined in Algorithm 11. As we will argue in the proof of Theorem 11.1, we assume at for Part 2 that at τ_{END1} we have with high probability $c_1 \geq (1-\varepsilon_{\text{Part1}})\cdot n$, where $\varepsilon_{\text{Part1}}$ is a small constant. Observe that Part 2 is executed after Part 1 defined in Algorithm 10. Therefore, $\tau_{\text{END0}} = \kappa \cdot \ell \cdot \log n$. We define the following reference points for Part 2.

$$\tau'_{\text{END0}} = 3/2 \cdot \tau_{\text{END0}} \quad \tau_{\text{END1}} = 2 \cdot \tau_{\text{END0}} \quad \tau_{\text{END2}} = 3 \cdot \tau_{\text{END0}} \quad \tau_{\text{END3}} = 4 \cdot \tau_{\text{END0}} \quad \tau_{\text{END4}} = 5 \cdot \tau_{\text{END0}}$$

Observe that according to the definition of Part 2 given in Algorithm 11 we only consider the working time (and not the real time). As Observation 11.6 Part 1 suggests, the working times of the nodes are sandwiched by the real time of the nodes and thus if we bound the real times of nodes, we get bounds on the working times as well.

From Observation 11.6 we obtain that all nodes have finished Part 1 at time step $\mathfrak{T}$ after at most $\mathfrak{T} \leq 3/2 \cdot \kappa \cdot \ell \cdot \log n = \tau'_{\text{END0}}$ ticks w.r.t. the working time. Furthermore, also due to Observation 11.6, we have that no node has yet reached τ_{END1} w.r.t. the working time at time step $\mathfrak{T}$. Therefore, we conclude that all nodes have completed Part 1 before any node starts the two choices process of Part 2 at reference point τ_{END1}. More precisely, all nodes are with high probability in $[\tau_{\text{END0}}, \tau_{\text{END1}}]$ before the first node passes τ_{END1}.

Since the real times are sandwiched, we get from Chernoff bounds that when the first node reaches τ_{END2}, all nodes are with high probability in $[\tau_{\text{END1}}, \tau_{\text{END2}}]$ w.r.t. the real time. We assume that nodes which are in $[\tau_{\text{END0}}, \tau_{\text{END4}}]$ respond, when queried, with the color they last set, possibly in Part 1 of the algorithm.

The remainder of this section is structured as follows. In Lemma 11.11 we give a lower bound on the size of $\mathcal{C}_1$ throughout the execution of Algorithm 9. This lower bound on $\mathcal{C}_1$ allows us to show that the number of nodes having any other color $\mathcal{C}_j \neq \mathcal{C}_1$ decreases quickly in expectation. This expected drop lets us apply a standard drift theorem, Theorem A.12, to obtain a bound on the required time until $\mathcal{C}_1$ prevails and all other colors vanish. Finally,

this will allow us to show that with high probability all nodes have set their color to $\mathcal{C}_1$ by the end of the the asynchronous algorithm at τ_{END4}.

For the next two lemmas, we will use the following notation. Consider an arbitrary but fixed time step t. Let a_t and b_t be the number of nodes of color 1 and $\mathcal{B}$ at time step t, respectively.

Lemma 11.11. *Assume that all nodes have a working time in $[\tau_{\text{END0}}, \tau_{\text{END4}}]$ during the time steps in $[n \cdot \tau'_{\text{END0}}, n \cdot \tau_{\text{END3}}]$. Assume furthermore that at time step $t = n \cdot \tau'_{\text{END0}}$ we have $a_t \geq 19n/20$. Then for any later tick t' in $[n \cdot \tau'_{\text{END0}}, n \cdot \tau_{\text{END4}}]$ we have $a_{t'} \geq 4n/5$, with high probability.*

Proof. To show the claim, we split Part 2 of the asynchronous algorithm into phases of $n/100$ consecutive time steps each. Based on these phases, we show the claim by an induction over every phase $i \in [100 \cdot \tau'_{\text{END0}}, 100 \cdot \tau_{\text{END4}}]$. By induction, we will show that we have with high probability at time step $t_i = i \cdot 100 \cdot n$

$$a_{t_i} \geq 17n/20 - i \cdot \sqrt{n} \cdot \log n \ .$$

Let now i be an arbitrary but fixed phase. We distinguish two cases.

Case 1: $a_{t_i} \geq 18n/20$**.** In this case the induction step holds trivially, since in the worst-case $a_{t_{i+1}} \geq a_{t_i} - (t_{i+1} - t_i) = 18n/20 - n/100 > 17n/20$.

Case 2: $a_{t_i} \leq 18n/20$**.** Observe that we have, by induction hypothesis, that for every $t \in [t_i, t_{i+1}]$ that $a_t \geq 17n/20 - i \cdot \sqrt{n} \cdot \log n - n/100 \geq 16.5n/20$. Furthermore, by assumption of the lemma we have $a_t \geq 19n/20$ at time step $t = n \cdot \tau'_{\text{END0}}$. We conclude that there are at least $n/20$ nodes that have already passed τ_{END2} and changed their color away from $\mathcal{C}_1$. However, by assumption of the lemma, these nodes have not yet passed τ_{END4}. These nodes can thus switch to $\mathcal{C}_1$ if they are selected to tick and choose two nodes of color 1.

We define the random variable X_t as 1 when a node of color $\mathcal{C}_j \neq \mathcal{C}_1$ is selected to tick and changes its color to $\mathcal{C}_1$ and as -1 if a node of color $\mathcal{C}_1$ is selected to tick and changes its color to any other color $\mathcal{C}_j \neq \mathcal{C}_1$. If neither of these cases apply, we define X_t to be zero. Observe, that the probability for X_t to be negative is maximized when $b_t = n - a_t$. Therefore, we have

$$X_t = \begin{cases} 1 & \text{with probability at least } 1/20 \cdot (16.5n/20)^2/n^2 = 272.25/20^3 \\ -1 & \text{with probability at most } 19/20 \cdot (3.5n/20)^2/n^2 = 232.75/20^3 \\ 0 & \text{otherwise.} \end{cases}$$

We now define Y_t as $Y_t = \sum_{k \leq t} X_k$ and show that Y_t is a sub-martingale.

$$\begin{aligned}\mathbb{E}[\, Y_t | Y_{t-1}, \ldots, Y_1 \,] &= Y_{t-1} + \mathbb{E}[\, X_t | Y_{t-1}, \ldots, Y_1 \,] \\ &\geq Y_{t-1} - 19/20 \cdot (3.5n/20)^2/n^2 + 1/20 \cdot (16.5n/20)^2/n^2 \\ &\geq Y_{t-1} \ .\end{aligned}$$

Since $|Y_t - Y_{t-1}| \leq 1$, applying the Azuma-Hoeffding bound to Y_t gives us

$$\mathbb{P}[\, Y_{t_{i+1}} - Y_{t_i} \geq -\sqrt{n} \cdot \log n \,] \leq \exp\left(-\frac{n \cdot \log^2 n}{2 \cdot n/100}\right) \ ,$$

which yields that the induction steps hold with high probability. This completes the proof. □

In the following (Lemma 11.12) we make use of multiplicative drift theorem (Theorem A.12 in Appendice A) which will allow us to derive a bound on the number of required periods until all nodes agree on one opinion.

Lemma 11.12. *Assume that all nodes have a working time in* $[\tau_{\text{END1}}, \tau_{\text{END4}}]$ *during the time steps in* $[n \cdot \tau_{\text{END2}}, n \cdot \tau_{\text{END3}}]$. *Furthermore assume that* $a_t \geq 4n/5$ *for any time step* $t \in [n \cdot \tau_{\text{END2}}, n \cdot \tau_{\text{END3}}]$. *Then at reference point* τ_{END3} *all nodes have opinion* $\mathcal{C}_1$ *with high probability, that is,* $a_{\tau_{\text{END3}}} = n$.

Proof. W.l.o.g. let $b_t = n - a_t$. We have

$$\begin{aligned}\mathbb{E}[\, b_{t+1} - b_t | \mathcal{F}_t \,] &= (+1)\frac{a_t}{n} \cdot \frac{b_t^2}{n^2} + (-1)\frac{b_t}{n}\frac{a_t^2}{n^2} \\ &= \frac{a_t \cdot b_t(b_t - a_t)}{n^3} \leq \frac{a_t \cdot b_t \cdot (-3/5)n}{n^3} \leq -\frac{4/5n \cdot b_t \cdot 3/5n}{n^3} \\ &= -\frac{12 \cdot b_t}{25n} \ .\end{aligned}$$

Let $\delta = 12/(25n)$ and define $\Phi(x_t) = b_t$. Note that $\Phi(x_{max}) \leq n$ and at any time step t we have $\mathbb{E}[\, \Phi(x_{t+1}) | \Phi(x_t) \,] \leq (1 - \delta)\Phi(x_t)$. Let $\mathcal{T}$ be the first point in time where all nodes agree on color 1, that is, $\mathcal{T} = \min\{t \geq 0 \colon \Phi(x_t) = 0\}$. We derive from Theorem A.12 with parameters δ and $k = 5 \log n$ that $\mathbb{P}[\, \mathcal{T} \geq 20/\delta \cdot \ln n \,] \leq n^{-5}$, where we used the Taylor series approximation for $\log(1 - \delta)$. Since $\tau_{\text{END3}} - \tau_{\text{END2}} \geq 20/\delta \ln n$, the claim follows. □

11.9 Putting Everything Together: Proof of Theorem 11.1

We use Proposition 11.5 (which builds on Proposition 11.4) and Lemma 11.12 to show Theorem 11.1, which is restated as follows.

Theorem 11.1. *Consider the asynchronous model. Let $G = K_n$ be the complete graph with n nodes. Let $k = \mathrm{O}(\exp(\log n/\log\log n))$ be the number of opinions. Let $\varepsilon_{\text{bias}} > 0$ be a constant. Assume $c_1 \geq (1+\varepsilon_{\text{bias}})\cdot c_i$ for all $i \geq 2$, then the asynchronous plurality consensus process defined in Section 11.2 on G converges within time $\Theta(\log n)$ to the majority opinion $\mathcal{C}_1$, with high probability.*

Proof. By Proposition 11.5 we have

$$x_j(\tau_{\text{BP2}}) = \frac{\hat{c}_j(\tau_0)^2}{x(\tau_{\text{BP1}})} \cdot (1 \pm \mathrm{o}(1)) + \mathrm{O}\Big(n^{1-4/\log\log n}\Big) \ .$$

Observe that due to the definition of x_j and $\mathcal{S}$, we have $x_j(\tau_{\text{T}}) = x_j(\tau_{\text{BP2}})$. Furthermore, note that $\hat{c}_1(\tau_0) \geq n/k \geq n^{1-1/\log\log n}$ and hence

$$\frac{\hat{c}_1(\tau_0)^2}{x(\tau_{\text{BP1}})} \geq n^{1-2/\log\log n} = \omega\Big(n^{1-4/\log\log n}\Big)$$

Let $a' := \hat{c}_1(\tau_0 + T)$ the nodes of color 1 belonging to $\mathcal{S}$ at the the beginning of the next round. Define b' analogously for color $\mathcal{B}$. We consider the ratio between and show a quadratic growth w.r.t. $\hat{c}_1(\tau_0)^2/\hat{c}_2(\tau_0+T)^2$. We derive

$$\frac{a'}{b'} \geq \frac{\frac{\hat{c}_1(\tau_0)^2}{x(\tau_{\text{BP1}})} \cdot (1-o(1))}{\frac{\hat{c}_2(\tau_0)^2}{x(\tau_{\text{TC}})} \cdot (1+o(1)) + \mathrm{O}(n^{1-4/\log\log n}))} \geq \frac{\hat{c}_1(\tau_0)^2}{\hat{c}_2(\tau_0)^2} \cdot (1 - \mathrm{o}(1)).$$

Hence, for sufficiently large constant ℓ, we have after $\ell \cdot \log\log n$ phases

$$\hat{c}_1 \geq 19n/20 \ . \tag{11.14}$$

As mentioned before (see Observation 11.6), using Chernoff bounds, we can show that with high probability:

1. All nodes have a working time in $[\tau_{\text{END0}}, \tau_{\text{END1}})$ at reference point τ'_{END0}. This implies that no node starts with two choices phase before all nodes finished Part 1 (Algorithm 10).

2. All nodes have a working time in $[\tau_{\text{END0}}, \tau_{\text{END4}}]$ during the reference points in $[\tau'_{\text{END0}}, \tau_{\text{END3}}]$. This together with above statement and (11.14) are the assumptions of Lemma 11.11.

3. All nodes have a working time in $[\tau_{\text{END1}}, \tau_{\text{END4}}]$ during the reference points in $[\tau_{\text{END2}}, \tau_{\text{END3}}]$. This is the assumption required by Lemma 11.12.

Thus, by Lemma 11.11 and Lemma 11.12, with high probability all nodes agree on $\mathcal{C}_1$ at τ_{END3}. Clearly, no node can change to any other color afterwards and, by Chernoff

bounds, after additional $\Theta(\log n)$ periods all nodes will have completed the execution of Algorithm 11. Thus the total run time is $\Theta(\log n)$. □

11.10 Increasing the Number of Opinions

In our proofs we considered the setting $k \leq \exp(\log n / \log\log n)$.

However, it is possible to allow for any $k = \mathrm{O}(n^{\varepsilon})$ (we still require that $a \geq (1+\varepsilon)b$). This requires the algorithm to have a bound on k so that the length of block Δ is adapted to $\Delta = \Theta(\log k + \log n / \log\log n)$. This is sufficient to get an equivalent notion of weak synchronicity. Due to the quadratic doubling, the algorithm requires $\mathrm{O}(\log\log n)$ phases. The length of the second part of the algorithm remains untouched resulting in a run time of $\mathrm{O}(\log k \cdot \log\log n + \log n)$.

11.11 Conclusions and Further Work

We introduced an algorithm to solve the plurality consensus in the asynchronous setting. Our algorithm achieves the best the possible asymptotic run time in the setting where the number of opinions k is bounded by $\exp(\log n / \log\log n)$. It remains an open question whether there exists an algorithm with the same run time allowing for $k = \mathrm{O}(n^{\varepsilon})$ opinions; we note that even in the synchronous setting this questions is open.

We believe that the concept of weak synchronicity (including the Sync Gadget and the tactical waiting) as well as our analysis techniques may well prove to be of independent interest.

We showed our main result assuming independent Poisson clocks with parameter 1. However, our techniques should carry over to a much more general setting where the nodes' clocks follow distributions satisfying the following properties: in $\Theta(\log n / \log\log n)$ periods $n - n/e^{O(\log n/\log\log n)}$ nodes tick $\Theta(\log n / \log\log n)$ times with high probability; a message is spread to $n - n/e^{O(\log n/\log\log n)}$ nodes within $\Theta(\log n / \log\log n)$ periods; at each (global) time t at least $n - n/e^{\Omega(\log n/\log^2\log n)}$ nodes ticked $t \pm O(\log n / \log\log n)$ many times; and the two-choice protocol converges within $\mathrm{O}(\log n)$ steps if the majority opinion is supported by 19/20 of the nodes. We believe we may also be able to relax the assumption that the nodes' clocks tick independently.

Moreover, we assumed that once a node contacts another node, it receives that node's response without any delay. This assumption, however, might be unrealistic in real networks (or other models of asynchronicity). We may address this issue by extending our model to allow for response delays following some exponential distribution with constant parameter (which need not be 1, but must be independent of n).

Finally, we feel that the ideas presented here may be applicable to the adaptation of synchronous protocols to asynchronous settings for a much wider class of problems, perhaps even eventually leading to a generic framework.

Chapter 12

Consensus via Load Balancing [BFK+16b]

In this chapter we consider plurality consensus on arbitrary connected and undirected graphs and a wide range of communication modes: From anything between simple sequential communication with a single neighbor (often used in biological settings as a simple variant of asynchronous communication [AR07]) to fully parallel communication where all nodes communicate with all their neighbors simultaneously (e.g. broadcasting models in distributed computing). This diversity turns out to be a major obstacle for algorithm design, since protocols (and their analysis) to a large degree depend upon the employed communication mechanism.

We present two simple protocols for the plurality consensus problem called Shuffle and Balance. Both protocols work in a very general discrete-time communication model. The communication partners are determined by a (possibly randomized) sequence $(\boldsymbol{M}_t)_{t\geq 0}$ of *communication matrices*, where we assume[1] N to be some suitably large polynomial in n. That is, nodes u and v can communicate in round t if and only if $\boldsymbol{M}_t[u,v] = 1$. In that case, we call the edge $\{u, v\}$ *active* (see [AKL08, SS12] for related graph models). Our results allow for a wide class of communication patterns (which can even vary over time) as long as the communication matrices have certain "mixing" properties (cf. Section 12.3).

In fact, load balancing is the source of inspiration for our protocols. Initially, each node creates a suitably chosen number of tokens labeled with its own opinion. Our Balance protocol then performs discrete load balancing on these tokens, allowing each node to get an estimate on the total number of tokens for each opinion. The Shuffle protocol keeps the number of tokens on every node fixed, but shuffles tokens between communication partners. By keeping track of how many tokens of their own opinion (label) were exchanged in total,

[1] For simplicity and without loss of generality; our protocols run in polynomial time in all considered models.

nodes gain an estimate on the total (global) number of such tokens. Together with a simple broadcast routine, all nodes can determine the plurality opinion.

The running time of our protocols is the smallest time t where all nodes have stabilized on the plurality opinion. That is, all nodes have determined the plurality opinion and will not change. This time depends on the network G, the communication pattern $(\boldsymbol{M}_t)_{t\geq 0}$, and the initial bias towards the plurality opinion (cf. Section 12.3). For both protocols we show a strong correlation between their running time, the mixing time of certain random walks, both of which are used in the analysis of recent load balancing results [SS12].

To give some more concrete examples of our results, let $T := O(\log n/(1-\lambda_2))$, where $1-\lambda_2$ is the spectral gap of G. If the bias is sufficiently high, then both our protocols Shuffle and Balance determine the plurality opinion in time

- $n \cdot T$ in the *sequential model* (only one pair of nodes communicates per time step);
- $d \cdot T$ in the *balancing circuit model* (communication partners are chosen according to d (deterministic) perfect matchings in a round-robin fashion); and
- T in the *diffusion model* (all nodes communicate with all their neighbors at once).

To the best of our knowledge, these match the best known bounds in the corresponding models. For an arbitrary bias (in particular, *arbitrarily small* bias), the protocols differ in their time and space requirements. More details of our results can be found in Section 12.1.

12.1 Results

We introduce two protocols for plurality consensus, called Shuffle and Balance. Both solve plurality consensus under a diverse set of (randomized or adversarial) communication patterns in arbitrary graphs for any positive bias. We continue with a detailed description of our results.

Shuffle. Our main result is the Shuffle protocol. In the first time step each node generates γ tokens labeled with its initial opinion. During round t, any pair of nodes connected by an active edge (as specified by the communication pattern $(\boldsymbol{M}_t)_{t\leq N}$) exchanges tokens. We show that Shuffle solves plurality consensus and allows for a trade-off between running time and memory.

More exactly, let the number of tokens be $\gamma = O\big(\log n/(\alpha^2 \cdot T)\big)$, where T is a parameter to control the trade-off between memory and running time[2]. Moreover, let t_{mix} be such that any time interval $[t, t+t_{\text{mix}}]$ is *ε-smoothing*[3] (cf. Section 12.3). Given knowledge of

[2]The protocol works for any integral choice of γ (this fixes the trade-off parameter T).

[3]Intuitively, this means that the communication pattern has good load balancing properties during any time window of length t_{mix}. This coincides with the worst-case mixing time of a lazy random walk on active edges.

the *maximum number degree* $\mathrm{d}_{\max}$ and the *mixing time* t_{mix} of the underlying communication pattern[4], SHUFFLE lets all nodes agree on the plurality opinion in $O(T \cdot t_{\mathrm{mix}})$ rounds (w.h.p.), using $O(\log n/(\alpha^2 T) \cdot \log k + \log(T \cdot t_{\mathrm{mix}}))$ memory bits per node.

This implies, for example, that plurality consensus on expanders in the sequential model is achieved in $O(T \cdot n \log n)$ time steps and with $O(\log n \cdot \log k/T + \log(Tn))$ memory bits (assuming a constant initial bias). For arbitrary graphs, arbitrary bias, and many natural communication patterns (e.g., communicating with all neighbors in every round or communicating via random matchings), the time for plurality consensus is closely related to the spectral gap of the underlying communication network (cf. Corollary 12.2).

Balance. The previous protocol, SHUFFLE, allows for a nice trade-off between running time and memory. If the number of opinions is relatively small, our much simpler BALANCE protocol gives better results.

In BALANCE, each node u maintains a k-dimensional load vector. Where j denotes u's initial opinion, the j-th dimension of this load vector is initialized with $\gamma \in \mathbb{N}$ (a sufficiently large value) and any other dimension is initialized with zero. In each time step, all nodes perform a simple, discrete load balancing on each dimension of these load vectors. Our results imply, for example, that plurality consensus on expanders in the sequential model is achieved in only $O(n \cdot \log n)$ time steps with $O(k)$ memory bits per node (assuming a constant initial bias).

BALANCE can be thought of as a (slightly simplified) version of [AGV15] or [KDG03] that generalizes naturally to $k \geq 2$ and arbitrary (even dynamic) graphs. In the setting of [AGV15] (but as opposed to [AGV15] for arbitrary k), it achieves plurality consensus with probability $1 - o(1)$ in parallel time $O(\log n)$ and uses $O(k \cdot \log(1/\alpha)) = O(k \cdot \log n)$ bits per node (Corollary 12.10), an improvement by a $\log(n)$ factor.

12.2 Approach and Technical Contributions

While our protocol SHUFFLE is relatively simple, the analysis is quite involved. The idea is to observe that after t_{mix} time steps, each single token is on any node with (roughly) the same probability; the difficulty is that token movements are not independent. The main ingredients for our analysis are Lemma 12.5 and Lemma 12.6, which generalize a result by Sauerwald and Sun [SS12] (we believe that this generalization is interesting in its own right). These lemmas show that the joint distribution of token locations is negatively correlated, allowing us to derive a suitable Chernoff bound. Once this is proven, nodes can "count" tokens every t_{mix} time steps, building up over time an estimate of the total number of tokens

[4]For static graphs, $\mathrm{d}_{\max}$ is the maximal degree which can be easily computed in a distributed way, see for example [BGPS06]. For t_{mix}, good bounds are known for many static graphs [AF02, Chapter 5].

labeled with their own opinion. By broadcasting these estimates, all nodes determine the plurality opinion.

On the other hand protocol BALANCE analysis' is straightforward: Even if the bias of the plurality color just one (the second largest color has support which is smaller by one), by construction, there will be $\geq \gamma = \text{poly}(n)$ many more tokens of the plurality token. The analysis of [SS12] then shows that sufficiently many time steps, the load discrepancy between any pair of nodes is so small that these additional γ tokens ensure that each node has strictly more tokens of the plurality color than of any other color.

12.3 Communication Model and Notation

We consider an undirected graph $G = (V, E)$ of $n \in \mathbb{N}$ nodes and let $1 - \lambda_2$ denote the eigenvalue (or spectral) gap of G. Each node u is assigned an *opinion* $o_u \in \{1, 2, \ldots, k\}$. For $i \in \{1, 2, \ldots, k\}$, we use $n_i \in \mathbb{N}$ to denote the number of nodes which have initially opinion i. Without loss of generality (w.l.o.g), we assume $n_1 > n_2 \geq \cdots \geq n_k$, such that 1 is the opinion that is initially supported by the largest subset of nodes. We also say that 1 is the *plurality opinion*. The value $\alpha := \frac{n_1 - n_2}{n} \in [1/n, 1]$ denotes the *initial bias* towards the plurality opinion. In the *plurality consensus problem*, the goal is to design simple, distributed protocols that let all nodes agree on the plurality opinion. Time is measured in discrete rounds, such that the (randomized) running time of our protocols is the number of rounds it takes until all nodes are aware of the plurality opinion. Further to the running time we also consider the total number of memory bits per node that are required by our protocols. All our statements and proofs assume n to be sufficiently large.

Communication Model. In any given round, two nodes u and v can communicate if and only if the edge between u and v is *active*. We use $\boldsymbol{M}_t$ to denote the symmetric *communication matrix* at time t, where $\boldsymbol{M}_t[u, v] = \boldsymbol{M}_t[v, u] = 1$ if $\{u, v\}$ is active and $\boldsymbol{M}_t[u, v] = \boldsymbol{M}_t[v, u] = 0$ otherwise. We assume (w.l.o.g) $\boldsymbol{M}_t[u, u] = 1$ (allowing nodes to "communicate" with themselves). Typically, the sequence $\boldsymbol{M} = (\boldsymbol{M}_t)_{t \in \mathbb{N}}$ of communication matrices (the *communication pattern*) is either randomized or adversarial, and our statements merely require that $\boldsymbol{M}$ satisfies certain smoothing properties (see below). For the ease of presentation, we restrict ourselves to polynomial number of time steps and consider only communication patterns $\boldsymbol{M} = (\boldsymbol{M}_t)_{t \geq 0}$ where $N = N(n)$ is an arbitrarily large polynomial. Let us briefly mention some natural and common communication models covered by such patterns:

- *Diffusion Model:* In every round t, all edges of the graph are activated.
- *Random matching model:* In every round t, the active edges are given by a random matching. We require that random matchings from different rounds are mu-

tually independent[5]. Results for the random matching model dependent on $p_{\min} \coloneqq \min_{t\in\mathbb{N},\{u,v\}\in E} \mathbb{P}[\boldsymbol{M}_t[u,v]=1]$.

- *Balancing Circuit Model:* There are d perfect matchings $\boldsymbol{M}_0, \boldsymbol{M}_1, \ldots, \boldsymbol{M}_{d-1}$ given. They are used in a round-robin fashion, such that for $t \geq d$ we have $\boldsymbol{M}_t = \boldsymbol{M}_{t \bmod d}$.
- *Sequential Model:* In each round t an edge $\{u,v\} \in E$ is activated uniformly random.

Notation. We use $\|\boldsymbol{x}\|_\ell$ to denote the ℓ-norm of vector $\boldsymbol{x}$, where the ∞-norm is the vector's maximum absolute entry. In general, bold font indicates vectors and matrices, and $x(i)$ refers to the i-th component of $\boldsymbol{x}$. The *discrepancy* of $\boldsymbol{x}$ is defined as $\operatorname{disc}(\boldsymbol{x}) \coloneqq \max_i x(i) - \min_i x(i)$. For $i \in \mathbb{N}$, we define $[i] \coloneqq \{1,2,\ldots,i\}$ as the set of the first i integers. We use $\log x$ to denote the binary logarithm of $x \in \mathbb{R}_{>0}$. We write $a \mid b$ if a divides b. For any node $u \in V$, we use $\deg_u$ to denote u's degree in G and $\deg_u(t) \coloneqq \sum_v \boldsymbol{M}_t[u,v]$ to denote its *active degree* at time t (i.e., its degree when restricted to active edges). Similarly, $N(u)$ and $N_t(u)$ refer to u's (active) neighborhood respectively. Moreover, $\mathrm{d}_{\max} \coloneqq \max_{t,u} \deg_u(t)$ is the maximum active degree of any node. We assume knowledge of $\mathrm{d}_{\max}$. On static graphs it can be computed efficiently in a distributed manner [BGPS06] and it is given by many dynamic graph models (e.g., 1 for the sequential model, d for balancing circuits). Let $c(t)$ be the configuration (state) of the all nodes at time t. We say an event happens with high probability (w.h.p.) if its probability is at least $1 - 1/n^c$ for $c \in \mathbb{N}$.

Random Walks. The running time of our protocols is closely related to the running time ("smoothing time") of diffusion load balancing algorithms, which in turn is a function of the mixing time of a random walk on G (see also [AKL08, SS12]). More exactly, we consider a random walk on G that is restricted to the active edges in each time step. As indicated in Section 12.1, this random walk should converge towards the uniform distribution over the nodes of G. This leads to the following definition of the random walk's transition matrices $\boldsymbol{P}_t$ based on the communication matrices $\boldsymbol{M}_t$:

$$\boldsymbol{P}_t[u,v] \coloneqq \begin{cases} \frac{1}{2\,\mathrm{d}_{\max}} & \text{if } \boldsymbol{M}_t[u,v]=1 \text{ and } u \neq v, \\ 1 - \frac{\deg_u(t)}{2\,\mathrm{d}_{\max}} & \text{if } \boldsymbol{M}_t[u,v]=1 \text{ and } u = v, \\ 0 & \text{if } \boldsymbol{M}_t[u,v]=0. \end{cases} \tag{12.1}$$

Obviously, $\boldsymbol{P}_t$ is doubly stochastic for all $t \in \mathbb{N}$. Moreover, note that the random walk is trivial in any matching-based model, while we get $\boldsymbol{P}_t[u,v] = \frac{1}{2d}$ for every edge $\{u,v\} \in E$ in the diffusion model on a d-regular graph. We are now ready to define the required mixing property.

[5]Note that there are several simple, distributed protocols to obtain such matchings [GM96, BGPS06].

Generalized Mixing Time Consider a fixed sequence $(\boldsymbol{M}_t)_{t\geq 1}$ of communication matrices.

In the following we generalize the standard definition for the mixing time to our general communication patterns. Let $d(t) = \max_{\boldsymbol{x},t_1} \|\boldsymbol{x} \cdot \prod_{t'=t_1}^{t_1+t} \boldsymbol{P}_{t'} - \boldsymbol{\pi}\|_{\text{TV}}$ and $\bar{d}(t) = \max_{\boldsymbol{x},\boldsymbol{y},t_1} \|\boldsymbol{x} \cdot \prod_{t'=t_1}^{t_1+t} \boldsymbol{P}_{t'} - \boldsymbol{y} \cdot \prod_{t'=t_1}^{t_1+t} \boldsymbol{P}_{t'}\|_{\text{TV}}$, where $\|\cdot\|_{\text{TV}}$ denotes the total variation distance. We define the *mixing time* to be $t_{\text{mix}}(\varepsilon) = \min\{t \geq 0 : \bar{d}(t) \leq \varepsilon\}$. For convenience we adapt the definition to only include $t_1 \in \text{poly}(n)$; otherwise, the mixing time might be arbitrary large for many communication patterns such as the sequential model.

The mixing time can be seen as the worst-case time required by a random walk to get "close" to the uniform distribution. If the parameter ε is not explicitly stated, we consider $t_{\text{mix}} \coloneqq t_{\text{mix}}(n^{-5})$.

12.4 Protocol Shuffle - Theorem 12.1

Our main result is the following theorem, stating the correctness as well as the time and space-efficiency of Shuffle. The protocol is described in Section 12.4.1, followed by its analysis in Section 12.4.2.

Theorem 12.1. *Let $\alpha = \frac{n_1-n_2}{n} \in [1/n, 1]$ denote the initial bias. Consider a fixed communication pattern $(\boldsymbol{M}_t)_{t\geq 1}$ and let $T \in \mathbb{N}$. Protocol* Shuffle *ensures that all nodes know the plurality opinion after $O(T \cdot t_{\text{mix}}(n^{-5}))$ rounds[6] (w.h.p.) and requires $\left(12 \cdot \frac{\log(n)}{\alpha^2 \cdot T} + 2\Delta + 4\right) \cdot \log(k) + 4\log\left(\frac{12 \cdot \log(n)}{\alpha^2}\right) + \log(T \cdot t_{\text{mix}})$ memory bits per node.*

The parameter T in the statement serves as a lever to trade running time for memory. Since $t_{\text{mix}}(n^{-5})$ depends on the graph and communication pattern, Theorem 12.1 might look a bit unwieldy. The following corollary gives a few concrete examples for common communication patterns on general graphs.

Corollary 12.2. *Let G be an arbitrary d-regular graph.* Shuffle *ensures that all nodes agree on the plurality opinion (w.h.p.) using $\left(12 \cdot \frac{\log(n)}{\alpha^2 \cdot T} + 2\Delta + 4\right) \cdot \log(k) + 4\log\left(\frac{12 \cdot \log(n)}{\alpha^2}\right) + \log(T \cdot t_{\text{mix}})$ bits of memory in time*

- $O\left(T \cdot \frac{\log(n)}{1-\lambda_2}\right)$ *in the diffusion model,*
- $O\left(\frac{T}{d \cdot p_{min}} \cdot \frac{\log(n)}{1-\lambda_2}\right)$ *in the random matching model,*
- $O\left(T \cdot d \cdot \frac{\log(n)}{1-\lambda_2}\right)$ *in the balancing circuit model, and*
- $O\left(T \cdot n \cdot \frac{\log(n)}{1-\lambda_2}\right)$ *in the sequential model.*

[6] This state is then maintained for a $\text{poly}(n)$ many rounds. For an exponential number of time steps, one would require larger counters to guarantee correctness.

12.4.1 Protocol Description

We continue to explain the SHUFFLE protocol given in Listing 12. Our protocol consists of three parts that are executed in each time step: the *shuffle* part, the *broadcast* part, and the *update* part.

Every node u is initialized with $\gamma \in \mathbb{N}$ tokens labeled with u's opinion o_u. Our protocol sends $2\,\mathrm{d}_{\max}$ tokens chosen uniformly at random (without replacement) over each edge $\{u, v\} \in E$ with $\boldsymbol{M}_t[u, v] = 1$.

Here, $\gamma \geq 2\,\mathrm{d}_{\max}$ is a parameter depending on T and α to be fixed during the analysis[7]. SHUFFLE maintains the invariant that, at any time, all nodes have exactly γ tokens.

In addition to storing the tokens, each node maintains a set of auxiliary variables. The variable c_u is increased during the update part of the protocol and counts tokens labeled o_u. The variable pair (dom_u, e_u) is a temporary guess of the plurality opinion and its frequency. During the broadcast part of the protocol, nodes broadcast these pairs, replacing their own pair whenever they observe a pair with higher frequency. Finally, the variable plu_u represents the opinion currently believed to be the plurality opinion. The shuffle and broadcast parts of the protocol are executed in each time step, while the update part is executed only every $t_{\mathrm{mix}}(n^{-5})$ time steps

Waiting $t_{\mathrm{mix}}(n^{-5})$ time steps for each update gives the broadcast enough time to inform all nodes and ensures that the tokens of each opinion are well distributed. The latter implies that, if we consider a node u with opinion $o_u = i$ at time $T \cdot t_{\mathrm{mix}}(n^{-5})$, the value c_u is a good estimate of $T \cdot \gamma n_i/n$ (which is maximized for the plurality opinion). When we reset the broadcast (Line 11), the subsequent $t_{\mathrm{mix}}(n^{-5})$ broadcast steps ensure that all nodes get

[7] SHUFFLE needs not to know α, it works for any choice of γ; such a choice merely fixes the trade-off parameter T.

to know the pair (o_u, c_u) for which c_u is maximal. Thus, if we can ensure that c_u is a good enough approximation of $T \cdot \gamma n_i/n$, all nodes get to know the plurality.

Algorithm 12: Protocol SHUFFLE as executed by node u at time t. At time zero, each node u creates γ tokens labeled o_u and sets $c_u := 0$ and $(dom_u, e_u) := (o_u, c_u)$.

Algorithm `Shuffle`

```
    // shuffle sub-phase
1   for { u, v } ∈ E with M_t[u, v] = 1 do
2       send 2 d_max tokens chosen u.a.r. (without replacement) to v

    // broadcast sub-phase
3   for { u, v } ∈ E with M_t[u, v] = 1 do
4       send (dom_u, e_u);
5       receive (dom_v, e_v)
6   v := w with e_w ≥ e_w'   ∀w, w' ∈ N_t(u) ∪ { u };
7   (dom_u, e_u) := (dom_v, e_v);

    // update sub-phase
8   if t ≡ 0 (mod t_mix)(n^-5) then
9       increase c_u by the number of tokens labeled o_u held by u ;
10      plu_u := dom_u ;              // plurality guess: last broadcast's dom. op.
11      (dom_u, e_u) := (o_u, c_u) ; // reset broadcast
```

12.4.2 Analysis of Shuffle

Fix a communication pattern $(\boldsymbol{M}_t)_{t\geq 0}$ and an arbitrary parameter $T \in \mathbb{N}$. Remember that $t_{\text{mix}} := t_{\text{mix}}(n^{-5})$ ensure that a random walk starting at any node and for any time step t and run for $t_{\text{mix}}(n^{-5})$ time steps will be n^{-5}-close to the stationary distribution. We set the number of tokens stored in each node to $\gamma := \left\lceil c \cdot \frac{\log n}{\alpha^2 T} \right\rceil$, where c is a suitable constant.

The analysis of SHUFFLE is largely based on Lemma 12.8, which states that, after $O(T \cdot t_{\text{mix}}(n^{-5}))$ time steps, the counter values c_u can be used to reliably separate the plurality opinion from any other opinion. The main technical difficulty is the dependency between the tokens' movements, rendering standard Chernoff-bounds inapplicable. Instead, we show that certain random variables satisfy the negative regression condition (Lemma 12.5), which allows us to majorize the token distribution by a random walk (Lemma 12.6) and to derive the Chernoff type bound in Lemma 12.7. This Chernoff type bound can be used to show that all counter values are concentrated which is the main pillar of the proof of Theorem 12.1.

Majorizing Shuffle by Random Walks

While our SHUFFLE protocol assumes that $2\,\mathrm{d}_{\max}$ divides γ, here we assume the slightly weaker requirement that $\boldsymbol{P}_t[u, v] \cdot \gamma \in \mathbb{N}$ for any $u, v \in V$ and $t \in \mathbb{N}$. Let us first introduce

some notation for the shuffle part of our protocol at time t. To ease the discussion, we consider u as a neighbor of itself and speak of $\deg_u(t)+1$ neighbors. For $i \in [\deg_u(t)+1]$, let $N_t(u,i) \in V$ denote the i-th neighbor of u (in an arbitrary order). Fix a node u and let u's tokens be numbered from 1 to γ. Our assumption on γ allows us to partition the tokens into $\deg_u(t)+1$ disjoint subsets (*slots*) $S_i \subseteq [\gamma]$ of size $\boldsymbol{P}_t[u,v] \cdot \gamma$ each, where $v = N_t(u,i)$. Let $\pi_{t,u} \colon [\gamma] \to [\gamma]$ be a random permutation of u's tokens at time t. All tokens j of node u at time t with $\pi_{t,u}(j) \in S_i$ are sent to u's i-th neighbor. To ease notation, we drop the time index t and write π_u instead of $\pi_{t,u}$ (and, similarly for $\deg_u$ and $N(u,i)$).

A *configuration* $\mathbf{c}$ describes the location of *all* γn tokens at a given point in time. For a token $j \in [\gamma n]$ we use $u_j \in V$ to denote its location in configuration $\mathbf{c}$ (which will always be clear from the context). For each such token j we define a random variable $X_j \in [\deg_{u_j}+1]$ with $X_j = i$ if and only if $\pi_{u_j}(j) \in S_i$. In other words, X_j indicates to which of u_j's neighbors token j is sent. Our key technical lemma (Lemma 12.5) establishes the *negative regression condition* for these $X = (X_j)_{j \in [\gamma n]}$ variables.

Formally, negative regression is defined as follows:

Definition 12.3 (Neg. Regression [DR98, Def. 21])**.** *For $n \in \mathbb{N}$, a vector $(X_1, X_2, \ldots, X_n)$ of random variables is said to satisfy the* negative regression condition (NRC) *if*

$$\mathbb{E}[\, f(X_l, l \in \mathscr{L}) | X_r = x_r, r \in \mathscr{R}\,]$$

is non-increasing in each x_r for any disjoint $\mathscr{L}, \mathscr{R} \subseteq [n]$ and for any non-decreasing function f.

The intuition is as follows. Consider two disjoint subsets $\mathscr{R}$ and $\mathscr{L}$ of $[\gamma n]$ and let $X_{\mathscr{R}}$, $X_{\mathscr{L}}$ be the corresponding sets of random variables. The higher the values of the random variables $X_{\mathscr{R}}$ we condition on, the (monotoinically) smaller the expectation of the random variables $X_{\mathscr{L}}$.

Lemma 12.4 ([DR98, Lemma 26])**.** *Let $n \in \mathbb{N}$ and assume $(X_1, X_2, \ldots, X_n)$ satisfy the negative regression condition. Consider an arbitrary index set $I \subseteq [n]$ as well as any family of non-decreasing functions f_i ($i \in I$). Then, we have*

$$\mathbb{E}\left[\prod_{i \in I} f_i(X_i)\right] \leq \prod_{i \in I} \mathbb{E}[\, f_i(X_i)\,]. \tag{12.2}$$

Lemma 12.5 (NRC)**.** *Fix a configuration $\mathbf{c}$ and consider the random variables $(X_j)_{j \in [\gamma n]}$. Then $(X_j)_{j \in [\gamma n]}$ satisfies the negative regression condition.*

Proof. Recall that u_j is the location of token j in configuration $\mathbf{c}$ and that $X_j \in [\deg_{u_j}+1]$ indicates the neighbor of u that token j is sent in the next step. We show for any $u \in V$ that $(X_j)_{j \colon u_j = u}$ satisfies the NRC. The lemma's statement follows since the π_u are chosen

independently: if two independent vectors (X_j) and (Y_j) satisfy the NRC, then so do both together.

Fix a node u and disjoint subsets $\mathscr{L}, \mathscr{R} \subseteq \{\, j \in [\gamma n] \colon u_j = u \,\}$ of tokens on u. Define $d \coloneqq \deg_u$ and let $f \colon [d+1]^{|\mathscr{L}|} \to \mathbb{R}$ be an arbitrary non-decreasing function. We have to show that $\mathbb{E}[\, f(X_l, l \in \mathscr{L}) | X_r = x_r, r \in \mathscr{R} \,]$ is non-increasing in each x_r (cf. Definition 12.3). That is, we need to show

$$\mathbb{E}[\, f(X_l, l \in \mathscr{L}) | X_r = x_r, r \in \mathscr{R} \,] \leq \mathbb{E}[\, f(X_l, l \in \mathscr{L}) | X_r = \tilde{x}_r, r \in \mathscr{R} \,], \tag{12.3}$$

where $x_r = \tilde{x}_r$ holds for all $r \in \mathscr{R} \setminus \{\, \hat{r} \,\}$ and $x_{\hat{r}} > \tilde{x}_{\hat{r}}$ for a fixed index $\hat{r} \in \mathscr{R}$.

We prove Inequality (12.3) via a coupling of the processes on the left-hand side (LHS process) and right-hand side (RHS process) of that inequality. Since $x_{\hat{r}} \neq \tilde{x}_{\hat{r}}$, these processes involve two slightly different probability spaces Ω and $\tilde{\Omega}$, respectively. To couple these, we employ a common uniform random variable $U_i \in [0, 1)$. By partitioning $[0, 1)$ into $d + 1$ suitable slots for each process (corresponding to the slots S_i mentioned above), we can use the outcome of U_i to set the X_j in both Ω and $\tilde{\Omega}$. We first explain how to handle the case $x_{\hat{r}} - \tilde{x}_{\hat{r}} = 1$. The case $x_{\hat{r}} - \tilde{x}_{\hat{r}} > 1$ follows from this by a simple reordering argument.

So assume $x_{\hat{r}} - \tilde{x}_{\hat{r}} = 1$. We reveal the yet unset random variables X_j (i.e., $j \notin \mathscr{R}$) one by one in order of increasing indices. To ease the description assume (w.l.o.g.) that the tokens from $\mathscr{R}$ are numbered from 1 to $|\mathscr{R}|$. When we reveal the j-th variable (which indicates the new location of the j-th token), note that the probability $p_{j,i}$ that token j is assigned to $N(u, i)$ (the i'th neighbour of u) depends solely on the *number* of previous tokens $j' < j$ that were assigned to $N(u, i)$. Thus, we can denote by $p_{j,i} \colon \mathbb{N} \to [0, 1]$ a function mapping $x \in \mathbb{N}$ to the probability that j is assigned to $N(u, i)$ conditioned on the event that exactly x previous tokens were assigned to $N(u, i)$. We observe that $p_{j,i}$ is non-increasing.

For a vector $\boldsymbol{x} \in \mathbb{N}^{d+1}$, we define a threshold function $T_{j,i} \colon \mathbb{N}^{d+1} \to [0, 1]$ by

$$T_{j,i}(\boldsymbol{x}) \coloneqq \sum_{i' \leq i} p_{j,i'}(x_{i'})$$

for each $i \in [d + 1]$. To define our coupling, let

$$\beta_{j,i} \coloneqq |\{\, j' < j \mid X_{j'} = i \,\}|$$

denote the number of already revealed variables with value i in the LHS process and define, similarly,

$$\tilde{\beta}_{j,i} \coloneqq |\{\, j' < j \mid \tilde{X}_{j'} = i \,\}|$$

for the RHS process. We use $\boldsymbol{\beta_j}, \tilde{\boldsymbol{\beta}}_{\boldsymbol{j}} \in \mathbb{N}^{d+1}$ to denote the corresponding vectors. Now, to assign token j we consider a uniform random variable $U_j \in [0, 1)$ and assign j in both processes using customized partitions of the unit interval. To this end, let $T_{j,i} \coloneqq T_{j,i}(\boldsymbol{\beta_j})$

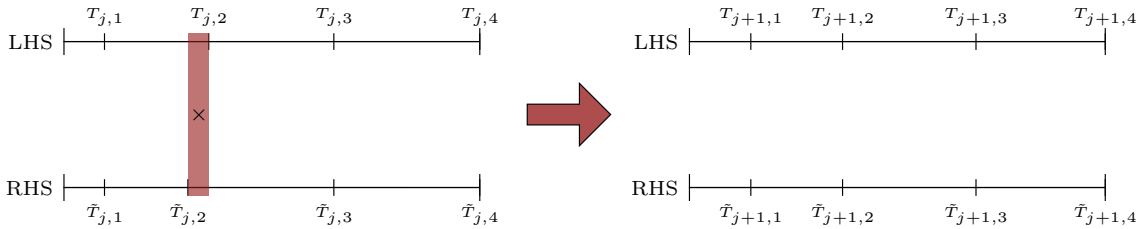


Figure 12.1: The figure illustrates the coupling between the two process for the token j and the token $j+1$. The first figure depicts the situation after the random decision of the j-th token is made and the second figure depicts the coupling for the subsequence token $j+1$. In this example, there are $d+1=4$ different slots for the LHS and RHS process, $x_{\hat{r}}=3$ and $\tilde{x}_{\hat{r}}=2$. On the left, the uniform random variable U_j falls into slot $[T_{j,1}, T_{j,2})$ for the LHS process (causing j to be sent to node $N(u,2)$) and into slot $[\tilde{T}_{j,2}, \tilde{T}_{j,3})$ for the RHS process (causing j to be sent to node $N(u,3)$).

and $\tilde{T}_{j,i} \coloneqq T_{j,i}(\tilde{\boldsymbol{\beta}}_{\boldsymbol{j}})$ for each $i \in [d+1]$. We assign X_j in the LHS and RHS process as follows:

- **LHS Process:** $X_j = x_j = i$ if and only if $U_j \in [T_{j,i-1}, T_{j,i})$,
- **RHS Process:** $X_j = \tilde{x}_j = i$ if and only if $U_j \in [\tilde{T}_{j,i-1}, \tilde{T}_{j,i})$.

See Figure 12.1 for an illustration. Our construction guarantees that the coupling is valid (i.e., considered in isolation, both the LHS and RHS process behave correctly).

At the beginning of this coupling, only the variables X_r corresponding to tokens $r \in \mathscr{R}$ are set, and these differ in the LHS and RHS process only for the index $\hat{r} \in \mathscr{R}$, for which we have $X_{\hat{r}} = x_{\hat{r}}$ (LHS) and $X_{\hat{r}} = \tilde{x}_{\hat{r}} = x_{\hat{r}} - 1$ (RHS). For the first revealed token $j = |\mathscr{R}| + 1$, this implies $\beta_{j,x_{\hat{r}}} = \tilde{\beta}_{j,x_{\hat{r}}} + 1$, $\beta_{j,x_{\hat{r}}-1} = \tilde{\beta}_{j,x_{\hat{r}}-1} - 1$, and $\beta_{j,i} = \tilde{\beta}_{j,i}$ for all $i \notin \{x_{\hat{r}}, x_{\hat{r}} - 1\}$. By the definitions of the slots for both processes, we get $T_{j,i} = \tilde{T}_{j,i}$ for all $i \neq x_{\hat{r}} - 1$ and $T_{j,x_{\hat{r}}-1} > \tilde{T}_{j,x_{\hat{r}}-1}$ (cf. Figure 12.1). Thus, the LHS and RHS process behave differently if and only if $U_i \in [\tilde{T}_{j,x_{\hat{r}}-1}, T_{j,x_{\hat{r}}-1})$. If this happens, we get $x_j < \tilde{x}_j$ (i.e., token j is assigned to a smaller neighbor in the LHS process). This implies $\boldsymbol{\beta_{j+1}} = \tilde{\boldsymbol{\beta}}_{\boldsymbol{j+1}}$ and both processes behave identical from now on. Otherwise, if $U_i \notin [\tilde{T}_{j,x_{\hat{r}}-1}, T_{j,x_{\hat{r}}-1})$, we have $\tilde{\boldsymbol{\beta}}_{\boldsymbol{j+1}} - \boldsymbol{\beta_{j+1}} = \tilde{\boldsymbol{\beta}}_{\boldsymbol{j}} - \boldsymbol{\beta_j}$ and we can repeat the above argument. Thus, after all X_j are revealed, there is at most one $j \in \mathscr{L}$ for which $x_j \neq \tilde{x}_j$, and for this we have $x_j < \tilde{x}_j$. Since f is non-decreasing, this guarantees Inequality (12.3). To handle the case $x_{\hat{r}} - \tilde{x}_{\hat{r}} > 1$, note that we can reorder the intervals $[T_{j,i-1}, T_{j,i})$ used for the assignment of the variables such that the corresponding slots for $x_{\hat{r}}$ and $\tilde{x}_{\hat{r}}$ are neighboring. Formally, this merely changes in which order we consider the neighbors in the definition of the functions $T_{j,i}$. With this change, the same arguments as above apply. □

Before proving the majorization of tokens with random walks (Lemma 12.6) we require further notation. Let $\mathscr{S}$ denote our random Shuffle process, and $\mathscr{W}$ the random walk process in which each of the γn tokens performs an independent random walk according to the sequence of random walk matrices $(\boldsymbol{P}_t)_{t\in\mathbb{N}}$ (i.e., a token on u uses $P_t(u,\cdot)$ for the

transition probabilities). We use $w_j^{\mathscr{P}}(t)$ to denote the position of token j after t steps of a process $\mathscr{P}$. We assume (w.l.o.g.) $w_j^{\mathscr{S}}(0) = w_j^{\mathscr{W}}(0)$ for all j. While there are strong correlations between the tokens' movements in $\mathscr{S}$ (e.g., not all tokens can move to the same neighbor), Lemma 12.6 shows that these correlations are negative.

Lemma 12.6 (Majorizing RWs)**.** *Consider a time $t \geq 0$, a token j, and node v. Let $B \subseteq [\gamma n]$ and $D \subseteq V$ be arbitrary subsets of tokens and nodes, respectively. The following holds:*

1. $\mathbb{P}\Big[\, w_j^{\mathscr{S}}(t) = v \,\Big] = \mathbb{P}\Big[\, w_j^{\mathscr{W}}(t) = v \,\Big]$ *and*
2. $\mathbb{P}\Big[\, \bigcap_{j\in B}\Big(w_j^{\mathscr{S}}(t) \in D\Big) \,\Big] \leq \mathbb{P}\Big[\, \bigcap_{j\in B}\Big(w_j^{\mathscr{W}}(t) \in D\Big) \,\Big] = \prod_{j\in B} \mathbb{P}\Big[\, w_j^{\mathscr{W}}(t) \in D \,\Big]$.

Proof. The first statement follows immediately from the definition of our process. For the second statement, note that the equality on the right-hand side holds trivially, since the tokens perform independent random walks in $\mathscr{W}$. To show the inequality, we define the intermediate process $\mathscr{S}\mathscr{W}(t')$ ($t' \leq t$) that performs t' steps of $\mathscr{S}$ followed by $t - t'$ steps of $\mathscr{W}$. By this definition, $\mathscr{S}\mathscr{W}(0)$ is identical to $\mathscr{W}$ restricted to t steps and, similar, $\mathscr{S}\mathscr{W}(t)$ is identical to $\mathscr{S}$ restricted to t steps. Define

$$\mathscr{E}_{t'} \coloneqq \bigcap_{j\in B}\Big(w_j^{\mathscr{S}\mathscr{W}(t')}(t) \in D\Big) \tag{12.4}$$

(the event that all tokens from B end up at nodes from D under process $\mathscr{S}\mathscr{W}(t')$). The lemma's statement is equivalent to $\mathbb{P}[\,\mathscr{E}_t\,] \leq \mathbb{P}[\,\mathscr{E}_0\,]$. To prove this, we show $\mathbb{P}[\,\mathscr{E}_{t'+1}\,] \leq \mathbb{P}[\,\mathscr{E}_{t'}\,]$ for all $t' \in \{\,0, 1, \ldots, t-1\,\}$. Combining these inequalities yields the desired result.

Fix an arbitrary $t' \in \{\,0, 1, \ldots, t-1\,\}$ and note that $\mathscr{S}\mathscr{W}(t')$ and $\mathscr{S}\mathscr{W}(t'+1)$ behave identical up to and including step t'. Hence, we can fix an arbitrary configuration (i.e., the location of each token) $\mathbf{c}(t') = \mathbf{c}$ immediately before time step $t'+1$. Remember that $u_j \in V$ denotes the location of j in configuration c. The auxiliary functions $h_j \colon [\deg_{u_j} + 1] \to [0, 1]$ describe the probability that a random walk starting at time $t'+1$ from u_j's i-th neighbor ends up in a node from D. Formally,

$$h_j(i) \coloneqq \mathbb{P}\Big[\, w_j^{\mathscr{W}}(t) \in D | w_j^{\mathscr{W}}(t'+1) = N(u_j, i) \,\Big]. \tag{12.5}$$

We can assume (w.l.o.g.) that all h_j are non-decreasing (by reordering the neighborhood of u_j).

Now, by Lemma 12.5 the variables $(X_j)_{j\in B}$ satisfy the negative regression condition. Thus, we can apply Lemma 12.4 (a well-known characterization of negative regression) to the functions h_j. Using another simple auxiliary result ((12.6) and (12.7)) we can relate the (conditioned) probabilities of the events $\mathscr{E}_{t'}$ and $\mathscr{E}_{t'+1}$ to the expectations over the different $h_j(X_j)$. We assume the following. Fix a time $t' \in \{\,0, 1, \ldots, t-1\,\}$ and consider an arbitrary

configuration c. Then the following identities hold:

$$\mathbb{P}[\,\mathscr{E}_{t'+1}|\mathbf{c}(t')=\mathbf{c}\,]=\mathbb{E}\left[\prod_{j\in B}h_j(X_j)|\mathbf{c}(t')=\mathbf{c}'\right] \tag{12.6}$$

and

$$\mathbb{P}[\,\mathscr{E}_{t'}|\mathbf{c}(t')=\mathbf{c}'\,]=\prod_{j\in B}\mathbb{E}[\,h_j(X_j)|\mathbf{c}(t')=\mathbf{c}'\,]. \tag{12.7}$$

We defer the proof the end of the lemma. Using this assumption, we compute

$$\begin{aligned}\mathbb{P}[\,\mathscr{E}_{t'+1}|\mathbf{c}(t')=\mathbf{c}\,]&\overset{(12.6)}{=}\mathbb{E}\left[\prod_{j\in B}h_j(X_j)|\mathbf{c}(t')=\mathbf{c}\right]\overset{\textit{Lemma 12.4}}{=}\prod_{j\in B}\mathbb{E}[\,h_j(X_j)|\mathbf{c}(t')=\mathbf{c}\,]\\&\overset{(12.7)}{=}\mathbb{P}[\,\mathscr{E}_{t'}|\mathbf{c}(t')=\mathbf{c}\,].\end{aligned}$$

Using the law of total probability, we conclude $\mathbb{P}[\,\mathscr{E}_{t'+1}\,]\leq\mathbb{P}[\,\mathscr{E}_{t'}\,]$, as required.

In the remainder we prove (12.6) and (12.7). Remember the definitions from Lemma 12.6 and its proof. We use the shorthand $\deg_{u_j}=\deg_{u_j}(t'+1)$. Remember that each X_j indicates to which of the $\deg_{u_j}+1$ neighbors of u_j (where u_j is considered a neighbor of itself) a token j moves during time step $t'+1$. Thus, given the configuration $\mathbf{c}(t')=\mathbf{c}'$ immediately before time step $t'+1$, there is a bijection between any possible configuration $\mathbf{c}(t'+1)$ and outcomes of the random variable vector $\boldsymbol{X}=(X_j)_{j\in[\gamma n]}$. Let $\mathbf{c}_{\boldsymbol{x}}$ denote the configuration corresponding to a concrete outcome $\boldsymbol{X}=\boldsymbol{x}\in[\deg_{u_j}+1]^{\gamma n}$. Thus, we have $\mathbb{P}[\,\mathbf{c}(t'+1)=\mathbf{c}_{\boldsymbol{x}}|\mathbf{c}(t')=\mathbf{c}'\,]=\mathbb{P}[\,\boldsymbol{X}=\boldsymbol{x}|\mathbf{c}(t')=\mathbf{c}'\,]$, and conditioning on $\mathbf{c}(t'+1)$ is equivalent to conditioning on $\boldsymbol{X}$ and $c(t')$. For the claim's first statement, we calculate

$$\begin{aligned}&\mathbb{P}[\,\mathscr{E}_{t'+1}|\mathbf{c}(t')=\mathbf{c}'\,]=\\&\quad\overset{(a)}{=}\sum_{\mathbf{c}_{\boldsymbol{x}}}\mathbb{P}[\,\mathscr{E}_{t'+1}|\mathbf{c}(t'+1)=\mathbf{c}_{\boldsymbol{x}}\,]\cdot\mathbb{P}[\,\mathbf{c}(t'+1)=\mathbf{c}_{\boldsymbol{x}}|\mathbf{c}(t')=\mathbf{c}'\,]\\&\quad\overset{(b)}{=}\sum_{\mathbf{c}_{\boldsymbol{x}}}\prod_{j\in B}\mathbb{P}\left[\,w_j^{\mathscr{SW}(t'+1)}(t)\in D|\boldsymbol{X}=\boldsymbol{x},\mathbf{c}(t')=\mathbf{c}'\right]\cdot\mathbb{P}[\,\boldsymbol{X}=\boldsymbol{x}|\mathbf{c}(t')=\mathbf{c}'\,]\\&\quad\overset{(c)}{=}\sum_{\mathbf{c}_{\boldsymbol{x}}}\prod_{j\in B}h_j(x_j)\cdot\mathbb{P}[\,\boldsymbol{X}=\boldsymbol{x}|\mathbf{c}(t')=\mathbf{c}'\,]\\&\quad=\sum_{\boldsymbol{x}}\prod_{j\in B}h_j(x_j)\cdot\mathbb{P}[\,\boldsymbol{X}=\boldsymbol{x}|\mathbf{c}(t')=\mathbf{c}'\,]=\mathbb{E}\left[\prod_{j\in B}h_j(X_j)|\mathbf{c}(t')=\mathbf{c}'\right],\end{aligned}$$

where (a) follows from law of total probability, (b) follows by using the bijection between $\mathbf{c}(t'+1)$ and $\boldsymbol{X}$ (if $c(t')$ is given) and that the process $\mathscr{SW}(t'+1)$ consists of independent

random walks if $\mathbf{c}(t'+1)$ is fixed, (c) we use the definition of the auxiliary functions $h_j(i)$, which equal the probability that a random walk starting at time $t'+1$ from u_j's i-th neighbor reaches a node from D.

For the claim's second statement, we do a similar calculation for the process $\mathscr{SW}(t')$. By definition, this process consists already from time t' onward of a collection of independent random walks. For a fixed configuration $\mathbf{c}c = \mathbf{c}c(t')$ let $\tilde{\boldsymbol{X}}$ denote the vector obtained after each token performed a step of an independent random walk. Thus, following the same arguments as before we obtain,

$$\begin{aligned}
&\mathbb{P}\big[\,\mathscr{E}_{t'}|\mathbf{c}(t') = \mathbf{c}'\,\big] = \\
&\quad= \sum_{\mathbf{c}_{\tilde{\boldsymbol{x}}}} \mathbb{P}\big[\,\mathscr{E}_{t'}|\mathbf{c}(t'+1) = \mathbf{c}_{\tilde{\boldsymbol{x}}}\,\big] \cdot \mathbb{P}\big[\,\mathbf{c}(t'+1) = \mathbf{c}_{\tilde{\boldsymbol{x}}}|\mathbf{c}(t') = \mathbf{c}'\,\big] \\
&\quad= \sum_{\mathbf{c}_{\tilde{\boldsymbol{x}}}} \prod_{j\in B} \mathbb{P}\Big[\, w_j^{\mathscr{SW}(t')}(t) \in D|\tilde{\boldsymbol{X}} = \tilde{\boldsymbol{x}}, \mathbf{c}(t') = \mathbf{c}'\,\Big] \cdot \mathbb{P}\Big[\,\tilde{\boldsymbol{X}} = \tilde{\boldsymbol{x}}|\mathbf{c}(t') = \mathbf{c}'\,\Big] \\
&\quad= \sum_{\mathbf{c}_{\tilde{\boldsymbol{x}}}} \prod_{j\in B} h_j(\tilde{x}_j) \cdot \mathbb{P}\Big[\,\tilde{\boldsymbol{X}} = \tilde{\boldsymbol{x}}|\mathbf{c}(t') = \mathbf{c}'\,\Big] \\
&\quad= \sum_{\tilde{\boldsymbol{x}}} \prod_{j\in B} h_j(\tilde{x}_j) \cdot \mathbb{P}\Big[\,\tilde{\boldsymbol{X}} = \tilde{\boldsymbol{x}}|\mathbf{c}(t') = \mathbf{c}'\,\Big] \\
&\quad= \mathbb{E}\left[\prod_{j\in B} h_j(\tilde{X}_j)|\mathbf{c}(t') = \mathbf{c}'\right].
\end{aligned}$$

The difference to before is that the $\tilde{X}_j$ are independent and thus we obtain,

$$\begin{aligned}
\mathbb{P}\big[\,\mathscr{E}_{t'}|\mathbf{c}(t') = \mathbf{c}'\,\big] = \mathbb{E}\left[\prod_{j\in B} h_j(\tilde{X}_j)|\mathbf{c}(t') = \mathbf{c}'\right] &= \prod_{j\in B} \mathbb{E}\Big[\, h_j(\tilde{X}_j)|\mathbf{c}(t') = \mathbf{c}'\,\Big] \\
&= \prod_{j\in B} \mathbb{E}\big[\, h_j(X_j)|\mathbf{c}(t') = \mathbf{c}'\,\big],
\end{aligned}$$

where the last equality stems from the fact that if consider only one token, then its distribution is the same in both processes.

□

Separating the Plurality via Chernoff

We rely on a bound of Theorem A.5 by [ABKU99] which allows us to majorize a sequence of "weakly dependent" variables by a sequence of binomially distributed variables.

We are finally able to prove the following Chernoff-like bound.

Lemma 12.7 (Token Concentration)**.** *Consider any subset B of tokens, a node $u \in V$, and an integer T. Let $X \coloneqq \sum_{1\le t\le T}\sum_{j\in B} X_{j,t}$, where $X_{j,t}$ is 1 if token j is on node u at time $t \cdot t_{\text{mix}}$. With $\mu \coloneqq (1/n + 1/n^5) \cdot |B| \cdot T$, we have $\mathbb{P}[\, X \ge (1+\delta)\cdot\mu\,] \le e^{\delta^2\mu/3}$.*

Proof. Let $v_{j,t}$ denote the location of token j at time $(t-1) \cdot t_{\text{mix}}$. For all $t \in \{1, \ldots, T\}$ and $\ell \in \mathbb{N}$ define the random indicator variable $Y_{j,t}$ to be 1 if and only if the random walk starting at $v_{j,t}$ is at node u after t_{mix} time steps. By Lemma 12.6 we have for each $B' \subseteq B$ and $t \in \{1, \ldots, T\}$ that

$$\mathbb{P}\left[\bigcap_{i \in B'} X_{j,t} = 1\right] \leq \prod_{j \in B'} \mathbb{P}[\, Y_{j,t} = 1\,]. \tag{12.8}$$

Hence for all $t \leq T$ and $\ell \in \mathbb{N}$ we have $\mathbb{P}\Big[\sum_{j \in B} X_{j,t} \geq \ell\Big] \leq \mathbb{P}\Big[\sum_{j \in B} Y_{j,t} \geq \ell\Big]$ and

$$\mathbb{P}[\, X \geq \ell\,] = \mathbb{P}\left[\sum_{1 \leq t \leq T} \sum_{j \in B} X_{j,t} \geq \ell\right] \leq \mathbb{P}\left[\sum_{1 \leq t \leq T} \sum_{j \in B} Y_{j,t} \geq \ell\right]. \tag{12.9}$$

Let us define $p \coloneqq 1/n + 1/n^5$. By the definition of t_{mix}, we have for all $j \in B$ and $t \leq T$ that

$$\mathbb{P}\Big[\, Y_{j,t} = 1 | Y_{1,1}, Y_{2,1}, \ldots, Y_{|B|,1}, Y_{1,2}, \ldots, Y_{j-1,t}\,\Big] \leq p. \tag{12.10}$$

Combining our observations with Theorem A.5 (see above), we get $\mathbb{P}[\, X \geq \ell\,] \leq \mathsf{Bin}(T \cdot |B|, p)$. Recall that $\mu = T \cdot |B| \cdot p$. Thus, by applying standard Chernoff bounds we get

$$\mathbb{P}[\, X \geq (1+\delta)\mu\,] \leq \left(\frac{e^\delta}{(1+\delta)^{1+\delta}}\right)^\mu \leq e^{\delta^2 \mu/3}, \tag{12.11}$$

which yields the desired statement. □

Together, these lemmas generalize a result given in [SS12] to a setting with considerably more dependencies. Equipped with this Chernoff bound, we prove concentration of the counter values.

Lemma 12.8 (Counter Separation)**.** *Let $c \geq 12$. For every time $t \geq c \cdot T \cdot t_{\text{mix}}$ there exist values $\ell_\top > \ell_\perp$ such that*

1. *For all nodes w with $o_w \geq 2$ we have (w.h.p.) $c_w \leq \ell_\perp$.*
2. *For all nodes v with $o_v = 1$ we have (w.h.p.) $c_v \geq \ell_\top$.*

For (a) we show that for any given node the number of tokens received is not much larger than its exception. See Figure 12.2 for an illustration. Showing (b) is slightly more involved. Due to the negative association, we refrain from showing directly that the number of tokens received by a node u of opinion $o_v = 1$ is lower bounded by its expectation minus some deviation. Instead, we show (b) by arguing that out of the total number of tokens considered which do *not* end up on node v with $o_v = 1$ is concentrated. This gives a lower bound on the number of tokens which did land on v.

Proof. Let v and w be two nodes with $o_v = 1$ and $o_w \geq 2$. We define for all $i \in \{2, \ldots, k\}$

$$\mu_i \coloneqq (1/n + 1/n^5)c \cdot T \cdot \gamma \cdot n_i \text{ and } \mu' \coloneqq (1/n + 1/n^5)c \cdot T \cdot \gamma \cdot (n - n_1).$$

For $i \in \{2, \ldots, k\}$ define

$$\ell_\perp(i) \coloneqq \mu_i + \sqrt{c^2 \cdot \log n \cdot T \cdot \gamma \frac{n_i}{n}} \text{ and } \ell_\top \coloneqq c \cdot T\gamma - \mu' - \sqrt{c^2 \cdot \log n \cdot T \cdot \gamma \frac{n - n_1}{n}}.$$

We set $\ell_\perp \coloneqq \ell_\perp(2)$. We first show that $\ell_\top > \ell_\perp$: recall that $\alpha = \frac{n_1 - n_2}{n} \geq 1/n$ and $\gamma = \left\lceil c_\gamma \cdot \frac{\log n}{\alpha^2 T} \right\rceil$, where c_γ is a suitable constant.

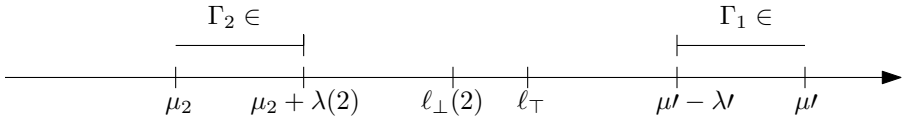


Figure 12.2: The figure depicts the random variables in the proof of Lemma 12.8 for the case of two nodes where node 1 supports opinion 1 and node 2 supports opinion 2. The total expected number of tokens counter by a node supporting the most prominent color 1 (2, respectively) is approximatively μ' (μ_2, respectively) up to an second order error accounting for the difference between the distribution of a random walk after t_{mix} time steps and the stationary distribution. We will show that w.h.p. the number of tokens counted will be in $[\mu' - \lambda', \infty]$ and in $[0, \mu_2 + \lambda(2)]$ for node 2. These intervals are separated by the carefully chosen quantities $\ell_\perp(2)$ and $\ell_\top$ where $\ell_\perp(2) < \ell_\top$. This establishes the separation of the counters.

We have

$$\begin{aligned}
\frac{\ell_\top - \ell_\perp}{c} &\geq T\gamma - (1/n + 1/n^5) \cdot T \cdot \gamma \cdot (n - n_1) - \sqrt{\cdot \log n \cdot T \cdot \gamma \frac{n - n_1}{n}} \\
&\quad - (1/n + 1/n^5)T \cdot \gamma \cdot n_2 - \sqrt{\cdot \log n \cdot T \cdot \gamma \frac{n_2}{n}} \\
&\geq T\gamma \cdot (\tfrac{n_1}{n} - \tfrac{n - n_1}{n^5}) - \sqrt{\log n \cdot T \cdot \gamma \tfrac{n - n_1}{n}} - T\gamma \cdot (\tfrac{n_2}{n} + \tfrac{n_2}{n^5}) - \sqrt{\log n \cdot T \cdot \gamma \tfrac{n_2}{n}} \\
&\geq \sqrt{T\gamma} \cdot \left(\sqrt{T\gamma} \cdot \left(\tfrac{n_1 - n_2}{n} - \tfrac{2}{n^4} \right) - \sqrt{\log n} \cdot \left(\sqrt{\tfrac{n - n_1}{n}} + \sqrt{\tfrac{n_2}{n}} \right) \right) \\
&\geq \sqrt{T\gamma} \cdot \left(\sqrt{\tfrac{c_\gamma \log n}{\alpha^2}} \left(\alpha - \tfrac{2}{n^4} \right) - \sqrt{\log n} \cdot \left(\sqrt{\tfrac{n - n_1}{n}} + \sqrt{\tfrac{n_2}{n}} \right) \right) \\
&\geq \sqrt{T\gamma} \cdot \left(\sqrt{\tfrac{c_\gamma \log n}{\alpha^2}} \left(0.9\alpha + 0.1/n - \tfrac{2}{n^4} \right) - \sqrt{\log n} \cdot \left(\sqrt{\tfrac{n - n_1}{n}} + \sqrt{\tfrac{n_2}{n}} \right) \right) \\
&\geq \sqrt{T\gamma} \cdot \left(0.9\sqrt{c_\gamma \log n} - 2\sqrt{\log n} \right) > 0.
\end{aligned}$$

where the last inequality holds for $c_\gamma \geq 5$. Now, let all γn tokens be labeled from 1 to γn. We proceed by showing the lemma's statements:

- For the first statement, consider a node w with $o_w \geq 2$ and set

$$\lambda(o_w) \coloneqq \ell_\perp(o_w) - \mu_{o_w} = \sqrt{c^2 \cdot \log n \cdot T \cdot \gamma \cdot n_{o_w}/n}.$$

Set the random indicator variable $X_{i,t}$ to be 1 if and only if token i is on node w at time t and if i's label is o_w. Let $c_w = \sum_{j=1}^{c\cdot T} \sum_{i\in B} X_{i,j\cdot t_{\text{mix}}}$, where B is the set of all tokens $[\gamma n]$ with label o_w. We compute

$$\begin{aligned}\mathbb{P}[\, c_w \geq \ell_\perp \,] &\leq \mathbb{P}[\, c_w \geq \mu_{o_w} + \lambda(o_w) \,] = \mathbb{P}\left[\, c_w \geq \left(1 + \frac{\lambda(o_w)}{\mu_{o_w}}\right) \cdot \mu_{o_w} \,\right] \\ &\leq \exp\left(-\frac{\lambda^2(o_w)}{3\mu_{o_w}}\right) \leq \exp\left(-\frac{c}{6}\log n\right),\end{aligned} \tag{12.12}$$

where the last line follows by Lemma 12.7 applied to $c_w = \sum_{j=1}^{c\cdot T} \sum_{i\in B} X_{i,j\cdot t_{\text{mix}}}$ and setting B to the set of all tokens with label o_w. Hence, the claim follows for c large enough after taking the union bound over all $n - n_1 \leq n$ nodes w with $o_w \geq 2$.

- For the lemma's second statement, consider a node v with $o_v = 1$ and set

$$\lambda' \coloneqq c \cdot T\gamma - \mu' - \ell_\top = \sqrt{c^2 \cdot \log n \cdot T \cdot \gamma \frac{n - n_1}{n}}.$$

Define the random indicator variable $Y_{i,t}$ to be 1 if and only if token i is on node v at time t and if i's label is not 1. Set $Y = \sum_{j=1}^{c\cdot T} \sum_{i\in B'} Y_{i,j\cdot t_{\text{mix}}}$, where B' is the set of all tokens with opinion different than 1. Note that $c_v = c \cdot T\gamma - Y$. We compute

$$\begin{aligned}\mathbb{P}[\, c_v \leq \ell_\top \,] &= \mathbb{P}[\, c \cdot T\gamma - Y \leq \ell_\top \,] = \mathbb{P}[\, c \cdot T\gamma - Y \leq c \cdot T\gamma - \mu' - \lambda' \,] \\ &= \mathbb{P}[\, Y \geq \mu' + \lambda' \,] = \mathbb{P}\left[\, Y \geq \left(1 + \frac{\lambda'}{\mu'}\right) \cdot \mu' \,\right] \\ &\leq \exp\left(-\frac{\lambda'^2}{3\mu'}\right) \leq \exp\left(\frac{c}{6}\log n\right),\end{aligned}$$

where the first inequality follows by Lemma 12.7 applied to Y. Hence, the claim follows for c large enough after taking the union bound over all $n_1 \leq n$ nodes v with $o_u \geq 2$.

□

We now give the proof of our main theorem.

Theorem 12.1. Fix an arbitrary time $t \in [c \cdot T \cdot t_{\text{mix}}, N]$ with t_{mix} dividing t, where c is the constant from the statement of Lemma 12.8. From Lemma 12.8 we have that (w.h.p.) the node u with the highest counter c_u has $o_u = 1$ (ties are broken arbitrarily). In the following we condition on $o_u = 1$. We claim that at time $t' = t + t_{\text{mix}}$ all nodes $v \in V$ have $plu_v = 1$. This is because the counters during the "broadcast part" (Lines 3 to 7) propagate the highest counter received after time t. The time τ until all nodes $v \in V$ have $plu_v = 1$ is bounded by the $t_{\text{mix}}(n^{-5})$ by definition: In order for $[t, t']$ to be $1/n^5$-smoothing, the random walk starting at u at time t is with probability at least $1/n - 1/n^5$ on node v

and, thus, there exists a path from u to v (with respect to the communication matrices). If there is such a path for every node v, the counter of u was also propagated to that v and we have $\tau \leq t_{\text{mix}}$. Consequently, at time t' all nodes have the correct majority opinion. This implies the desired time bound. For the memory requirements, note that each node u stores γ tokens with a label from the set $[k]$ ($\gamma \cdot O(\log k)$ bits), three opinions (its own, its plurality guess, and the dominating opinion; $O(\log k)$ bits), the two counters c_u and e_u and the time step counter. The memory to store the counter c_u and e_u is $O(\gamma T)$. Finally, the time step counter is bounded by $O(\log(T \cdot t_{\text{mix}}))$ bits. This yields the claimed space bound. □

12.5 Protocol Balance - Theorem 12.9

Protocol Description. The idea of our BALANCE protocol is quite simple: Every node u stores a k-dimensional vector $\boldsymbol{\ell_t(u)}$ with k integer entries, one for each opinion. BALANCE performs an entry-wise load balancing on $\boldsymbol{\ell_t(u)}$ according to the communication pattern $\boldsymbol{M} = (\boldsymbol{M}_t)_{t\geq 0}$ and the corresponding transition matrices $\boldsymbol{P}_t$ (cf. Section 12.3). Once the load is properly balanced, the nodes look at their largest entry and assume that this is the plurality opinion (stored in the variable plu_u).

In order to ensure a low memory footprint, we must not send fractional loads over active edges. To this end, we use a rounding scheme from [BCF+15, SS12], which works as follows: Consider a dimension $i \in [k]$ and let $\ell_{i,t}(u) \in \mathbb{N}$ denote the current (integral) load at u in dimension i, then u sends $\lfloor \ell_{i,t}(u) \cdot \boldsymbol{P}_t[u,v] \rfloor$ tokens to all neighbors v with $\boldsymbol{M}_t[u,v] = 1$. This results in at most $\deg_u(t)$ remaining *excess tokens* ($\ell_{i,t}(u)$ minus the total number of tokens sent out). These are then randomly distributed (without replacement), where neighbor v receives a token with probability $\boldsymbol{P}_t[u,v]$. In the following we call the resulting balancing algorithm the VERTEX-BASED BALANCER algorithm.

The formal description of protocol BALANCE is given in Listing 13.

Algorithm 13: Protocol BALANCE as executed by node u at time t. At time zero, each node initializes $\ell_{o_u,0}(u) \coloneqq \gamma$ and $\ell_{j,0}(u) \coloneqq 0$ for all $j \neq o_u$. .

Algorithm `Shuffle`

- **for** $i \in [k]$ **do**
  - **for** $\{u, v\} \in E$ *with* $\boldsymbol{M}_t[u,v] = 1$*:* **do**
    - send $\lfloor \ell_{i,t}(u) \cdot \boldsymbol{P}_t[u,v] \rfloor$ tokens from dimension i to v
  - $x \coloneqq \ell_{i,t}(u) - \sum_{v:\ \boldsymbol{M}_t[u,v]=1} \lfloor \ell_{i,t}(u) \cdot \boldsymbol{P}_t[u,v] \rfloor$; `// excess tokens`
  - **forall the** *token* x **do**
    - randomly distribute x such that:;
    - every $v \neq v$ with $\boldsymbol{M}_t[u,v] = 1$ receives 1 token w.p. $\boldsymbol{P}_t[u,v]$ (and zero otherwise)
- $plu_u \coloneqq i$ with $\ell_{i,t}(u) \geq \ell_{j,t}(u) \quad \forall 1 \leq i,j \leq k$; `// plurality guess`

Analysis of Balance. Consider initial load vectors $\boldsymbol{\ell_0}$ with $\|\boldsymbol{\ell_0}\|_\infty \leq n^5$. Let $\tau := \tau(g, \boldsymbol{M})$ be the first time step when Vertex-Based Balancer under the (fixed) communication pattern $\boldsymbol{M} = (\boldsymbol{M}_t)_{t\geq 0}$ is able to balance any such vector $\boldsymbol{\ell_0}$ up to a g-discrepancy. With this, we show:

Theorem 12.9. *Let $\alpha = \frac{n_1-n_2}{n} \in [1/n, 1]$ denote the initial bias. Consider a fixed communication pattern $\boldsymbol{M} = (\boldsymbol{M}_t)_{t\geq 0}$ and let $\gamma \in [3 \cdot \frac{g}{\alpha}, n^5]$ be an arbitrary integer. Protocol* Balance *ensures that all nodes know the plurality opinion after $\tau(g, \boldsymbol{M})$ rounds and requires $k \cdot \log(\gamma)$ memory bits per node.*

Proof. Recall that $\gamma \geq 3\frac{g}{\alpha} = 3g \cdot \frac{n}{n_1-n_2}$. For $i \in [k]$ let $\bar{\ell}_i := n_i \cdot \gamma/n$. The definition of $\tau(g, \boldsymbol{M})$ implies $\ell_{1,t}(u) \geq \bar{\ell}_1 - g$ and $\ell_{i,t}(u) \leq \bar{\ell}_i + g$ for all nodes u and $i \geq 2$. Consequently, we get

$$\ell_{1,t}(u) - \ell_{i,t}(u) \geq \bar{\ell}_1 - \bar{\ell}_i - 2g = 3g \cdot \frac{n_1 - n_i}{n_1 - n_2} - 2g > 0. \tag{12.13}$$

Thus, every node u has the correct plurality guess at time t. □

The memory usage of Balance depends on the number of opinions (k) and on the number of tokens generated on every node (γ). The algorithm is very efficient for small values of k but it becomes rather impractical if k is large. Note that if one chooses γ sufficiently large, it is easy to adjust the algorithm such that every node knows the frequency of *all* opinions in the network. The next corollary gives a few concrete examples for common communication patterns on general graphs.

Corollary 12.10. *Let G be an arbitrary d-regular graph.* Balance *ensures that all nodes agree on the plurality opinion with probability $1 - e^{-(\log(n))^c}$ for some constant c*

1. *using $O(k \cdot \log n)$ bits of memory in time $O(\frac{\log n}{1-\lambda_2})$ in the diffusion model,*
2. *using $O(k \cdot \log n)$ bits of memory in time $O(\frac{1}{d \cdot p_{min}} \cdot \frac{\log n}{1-\lambda_2})$ in the random matching model,*
3. *using $O(k \cdot \log(\alpha^{-1}))$ bits of memory in time $O(d \cdot \frac{\log n}{1-\lambda_2})$ in the balancing circuit model, and*
4. *using $O(k \cdot \log(\alpha^{-1}))$ bits of memory in time $O(n \cdot \frac{\log n}{1-\lambda_2})$ in the sequential model.*

Proof. Part 1 follows directly from [SS12, Theorem 6.6] and Part 3 follows directly from [SS12, Theorem 1.1]. To show Part 2 and 4 we choose τ such that $\boldsymbol{M}_1, \boldsymbol{M}_2, \ldots, \boldsymbol{M}_\tau$ enable Vertex-Based Balancer to balance any vector $\boldsymbol{\ell_0}$ (with initial discrepancy of at most n^5) up to a *g-discrepancy*. The bound on τ then follows from [SS12, Theorem 1.1]. □

Chapter 13

Future Work - Distributed Consensus Processes

The results for consensus dynamics in Part II can be generalized in many ways—the most obvious being to general graphs. Another interesting open problem would be to analyze 3-Majority and faster protocols in presence of an adversary that is allowed to change the opinion of say $\sqrt{n}$ nodes per round—starting from all configurations.

In the following we elaborate on these research directions.

Fault Tolerance. As mentioned in Chapter 9, previous studies [BCN+14b, BCN+16, CER14, EFK+16] show that 2-Choices and 3-Majority are consensus protocols that can tolerate dynamic, worst-case adversarial faults. More specifically, the protocols work even in the presence of an adversary that can, in every round, corrupt the state of a bounded set of nodes. The goal in this setting is to achieve a stable regime in which "almost-all" nodes support the same *valid* color (i.e. a color initially supported by at least one non-corrupted node). The size of the corrupted set is one of the studied quality parameters and depends on the number k of colors and/or on the bias in the starting configuration. For instance, in [BCN+16] it is proven that, for $k = o\left(n^{1/3}\right)$, 3-Majority tolerates a corrupted sets of size $O\left(\sqrt{n}/(k^{5/2}\log n)\right)$. A natural important open issue is to investigate whether our framework for AC-processes can be used to make statements about fault-tolerance properties in this (or in similar) adversarial models. We moderately lean toward thinking that our analysis is sufficiently general and "robust" to be suitably adapted in order to cope with this adversarial scenario over a wider range of k and bias w.r.t. the relative previous analyses. Finally, we ask whether there exists any poly-log consensus time protocols agreeing on the plurality color in presence of an adversary?

Towards a Hierarchy. Consider the process functions of the general h-Majority process for arbitrary $h \in \mathbb{N}$. Intuitively, h-Majority should be (stochastically) slower than $(h+1)$-

Majority. We strongly believe this result holds. However, naïvely applying our machinery to prove this does not work and needs to be amended. Our conjecture that such a "hierarchy" for h$-Majority$ for different $h \in \mathbb{N}$ holds is backed by the proof of Proposition 9.10 (which shows this for $h \in \{1, 2, 3\}$, since the Voter process is actually equivalent to 1-Majority and 2-Majority).

Conjecture 13.1. *For $h \in \mathbb{N}$, we can couple h-Majority and $(h+1)$-Majority such that the latter never has more remaining colors than the former. In particular, $(h+1)$-Majority is stochastically faster than h-Majority.*

However, as we have shown in Section 9.5 via a counterexample, it turns out that Lemma 9.5 is not strong enough to derive Conjecture 13.1. In fact, our failed attempts in adapting our approach may suggest that similar counterexamples exist for any majorization attempt that uses a total order on vectors.

General Graphs. To obtain result for 3-Majority on general graphs, it seems natural to try generalizing the coupling of Proposition 9.10 to general graphs. However, there are graphs for which this is not true: versions of the stochastic block model seem to require a super-exponential number of rounds whereas Voter requires at most $O(n^3)$ rounds [KMS16]. Nevertheless, 3-Majority seems to perform at least as good as Voter on graphs like the cycle, the grid and the star. It is natural to ask whether there are any dynamics reaching consensus faster than Voter, 3-Majority, and 2-Choices on any graph.

The Right Model. The most flagrant and yet most difficult and long-term based challenge is to find the right model. We are far from a general understanding of the domain partially due to the vast choice of parameters: Synchronous vs asynchronous time steps, knowledge of the graph size n, execution starts for all nodes at the same time, presence of adversaries, trade-off consensus guarantees and space etc. Finding the correct model and assumptions could benefit greatly from industrial applications as well as applications in computational biology modelling the behaviour and communication of species. Alternatively, it would be very interesting to obtain a general understand of how results in one setting relate to another—is it possible to translate results for synchronous dynamics to asynchronous dynamics and vice versa?

Bibliography

[AD15] M. A. Abdullah and M. Draief: *Global majority consensus by local majority polling on graphs of a given degree sequence*. In *Discrete Applied Mathematics*, volume 180, 2015, pages 1–10.

[ABS98] M. Adler, P. Berenbrink, and K. Schröder: *Analyzing an Infinite Parallel Job Allocation Process*. In *Proceedings of the 6th Annual European Symposium on Algorithms*. ESA. Springer-Verlag, 1998, pages 417–428. URL: http://dl.acm.org/citation.cfm?id=647908.740138.

[ACMR98] M. Adler, S. Chakrabarti, M. Mitzenmacher, and L. Rasmussen: *Parallel randomized load balancing*. In *Random Structures & Algorithms*, volume 13 (2), 1998, pages 159–188. URL: http://dx.doi.org/10.1002/(SICI)1098-2418(199809)13:2%3C159::AID-RSA3%3E3.0.CO;2-Q.

[AF02] D. Aldous and J. Fill: *Reversible Markov Chains and Random Walks on Graphs*. Unpublished. http://www.stat.berkeley.edu/~aldous/RWG/book.html. 2002.

[Alf03] A. S. Alfa: *Algorithmic analysis of the BMAP/D/k system in discrete time*. In *Adv. in Appl. Probab.* volume 35 (4), 2003, pages 1131–1152. URL: http://dx.doi.org/10.1239/aap/1067436338.

[AAG17] D. Alistarh, J. Aspnes, and R. Gelashvili: *Space-Optimal Majority in Population Protocols*. In *ArXiv e-prints*, 2017. arXiv: 1704.04947 [cs.DC].

[AGV15] D. Alistarh, R. Gelashvili, and M. Vojnović: *Fast and Exact Majority in Population Protocols*. In *Proc. PODC*, 2015, pages 47–56.

[AAK+11] N. Alon, C. Avin, M. Koucký, G. Kozma, Z. Lotker, and M. R. Tuttle: *Many Random Walks Are Faster Than One*. In *Combinatorics, Probability & Computing*, 2011.

[Ang80] D. Angluin: *Local and Global Properties in Networks of Processors (Extended Abstract)*. In *Proceedings of the Twelfth Annual ACM Symposium on Theory of Computing*. STOC '80. ACM, 1980, pages 82–93. URL: http://doi.acm.org.proxy.lib.sfu.ca/10.1145/800141.804655.

[AAE08] D. Angluin, J. Aspnes, and D. Eisenstat: *A simple population protocol for fast robust approximate majority*. In *Distributed Computing*, volume 21 (2), 2008, pages 87–102.

[AAER07] D. Angluin, J. Aspnes, D. Eisenstat, and E. Ruppert: *The computational power of population protocols*. In *Distributed Computing*, volume 20 (4), 2007, pages 279–304.

[AFJ06] D. Angluin, M. J. Fischer, and H. Jiang: *Stabilizing consensus in mobile networks*. In *Proc. of Distributed Computing in Sensor Systems (DCOSS)*. LNCS, volume 4026, 2006, pages 37–50.

[AR07] J. Aspnes and E. Ruppert: *An Introduction to Population Protocols*. In *Bulletin of the EATCS*, volume 93, 2007, pages 98–117.

[ALNP15] C. Avin, Z. Lotker, Y. Nahum, and D. Peleg: *Core Size and Densification in Preferential Attachment Networks*. In *Automata, Languages, and Programming - 42nd International Colloquium, ICALP 2015*, 2015, pages 492–503.

[AKL08] C. Avin, M. Koucký, and Z. Lotker: *How to Explore a Fast-Changing World (Cover Time of a Simple Random Walk on Evolving Graphs)*. In *Automata, Languages and Programming, 35th International Colloquium, ICALP 2008, Reykjavik, Iceland, July 7-11, 2008, Proceedings, Part I: Tack A: Algorithms, Automata, Complexity, and Games*, 2008, pages 121–132. URL: http://dx.OPTdoi.org/10.1007/978-3-540-70575-8_11.

[ABKU99] Y. Azar, A. Z. Broder, A. R. Karlin, and E. Upfal: *Balanced Allocations*. In *SIAM J. Comput.* volume 29 (1), 1999, pages 180–200. URL: http://dx.doi.org/10.1137/S0097539795288490.

[Bai54] N. T. J. Bailey: *On Queueing Processes with Bulk Service*. In *Journal of the Royal Statistical Society. Series B (Methodological)*, volume 16 (1), 1954, pages 80–87. URL: http://www.jstor.org/stable/2984011.

[BA99] A. Barabási and R. Albert: *Emergence of scaling in random networks*. In *science*, volume 286 (5439), 1999, pages 509–512.

[BPS12] M. Barlow, Y. Peres, and P. Sousi: *Collisions of Random Walks*. In *Annales de l' Institut Henri Poincare*, volume 48 (4), 2012, pages 922–946.

[BCN+15a] L. Becchetti, A. E. F. Clementi, E. Natale, F. Pasquale, and G. Posta: *Self-Stabilizing Repeated Balls-into-Bins*. In *Proceedings of the 27th ACM on Symposium on Parallelism in Algorithms and Architectures, SPAA 2015, Portland, OR, USA, June 13-15, 2015*. ACM, 2015, pages 332–339. URL: http://doi.acm.org/10.1145/2755573.2755584.

[BCN+14a] L. Becchetti, A. E. F. Clementi, E. Natale, F. Pasquale, R. Silvestri, and L. Trevisan: *Simple dynamics for plurality consensus*. In *26th ACM Symposium on Parallelism in Algorithms and Architectures, (SPAA)*, 2014, pages 247–256.

[BCN+16] L. Becchetti, A. E. F. Clementi, E. Natale, F. Pasquale, and L. Trevisan: *Stabilizing Consensus with Many Opinions*. In *Proceedings of the Twenty-Sixth Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*. SIAM, 2016.

[BCN+15b] L. Becchetti, A. Clementi, E. Natale, F. Pasquale, and R. Silvestri: *Plurality Consensus in the Gossip Model*. In *Proceedings of the 26th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, 2015, pages 371–390.

[BCN+14b] L. Becchetti, A. Clementi, E. Natale, F. Pasquale, R. Silvestri, and L. Trevisan: *Simple Dynamics for Plurality Consensus*. In *Proc. SPAA*, 2014, pages 247–256.

[BTV09] F. Bénézit, P. Thiran, and M. Vetterli: *Interval consensus: From quantized gossip to voting*. In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing, ICASSP*. IEEE, 2009, pages 3661–3664.

[BTV11] F. Bénézit, P. Thiran, and M. Vetterli: *The Distributed Multiple Voting Problem*. In *J. Sel. Topics Signal Processing*, volume 5 (4), 2011, pages 791–804.

[BDDS10] O. Ben-Shahar, S. Dolev, A. Dolgin, and M. Segal: *Direction election in flocking swarms*. In *Proc. of the 6th Int. Workshop on Foundations of Mobile Computing (DIALM-POMC)*. ACM, 2010, pages 73–80.

[BCE+17] P. Berenbrink, A. E. F. Clementi, R. Elsässer, P. Kling, F. Mallmann-Trenn, and E. Natale: *Ignore or Comply?: On Breaking Symmetry in Consensus*. In *Proceedings of the ACM Symposium on Principles of Distributed Computing, PODC'17, Washington, DC, USA, July 25-27, 2017*, 2017, pages 335–344. URL: http://doi.acm.org/10.1145/3087801.3087817.

[BCF+15] P. Berenbrink, C. Cooper, T. Friedetzky, T. Friedrich, and T. Sauerwald: *Randomized diffusion for indivisible loads*. In *J. Comput. Syst. Sci.* volume 81 (1), 2015, pages 159–185.

[BCE+12] P. Berenbrink, A. Czumaj, M. Englert, T. Friedetzky, and L. Nagel: *Multiple-Choice Balanced Allocation in (Almost) Parallel*. English. In *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques*, volume 7408, Lecture Notes in Computer Science. Springer Berlin Heidelberg, 2012, pages 411–422. URL: http://dx.doi.org/10.1007/978-3-642-32512-0_35.

[BCFV00] P. Berenbrink, A. Czumaj, T. Friedetzky, and N. D. Vvedenskaya: *Infinite Parallel Job Allocation (Extended Abstract)*. In *Proceedings of the Twelfth Annual ACM Symposium on Parallel Algorithms and Architectures*. SPAA. ACM, 2000, pages 99–108. URL: http://doi.acm.org/10.1145/341800.341813.

[BCSV06] P. Berenbrink, A. Czumaj, A. Steger, and B. Vöcking: *Balanced Allocations: The Heavily Loaded Case*. In *SIAM Journal on Computing*, volume 35 (6), 2006, pages 1350–1385.

[BEMA14] P. Berenbrink, F. Ergün, F. Mallmann-Trenn, and E. S. Azer: *Palindrome Recognition In The Streaming Model*. In *31st International Symposium on Theoretical Aspects of Computer Science STACS*. Volume 25, 2014, pages 149–161. URL: http://drops.dagstuhl.de/opus/volltexte/2014/4454.

[BFGK16] P. Berenbrink, T. Friedetzky, G. Giakkoupis, and P. Kling: *Efficient Plurality Consensus, or: The Benefits of cleaning up from time to time*. In *Proceedings of the 43rd International Colloquium on Automata, Languages and Programming (ICALP)*. to appear, 2016.

[BFG03] P. Berenbrink, T. Friedetzky, and L. A. Goldberg: *The Natural Work-Stealing Algorithm is Stable*. In *SIAM J. Comput.* volume 32 (5), 2003, pages 1260–1279. URL: http://dx.doi.org/10.1137/S0097539701399551.

[BFK+16a] P. Berenbrink, T. Friedetzky, P. Kling, F. Mallmann-Trenn, L. Nagel, and C. Wastell: *Self-stabilizing Balls & Bins in Batches: The Power of Leaky Bins*. In *Proceedings of the 2016 ACM Symposium on Principles of Distributed Computing*. PODC'16, 2016, pages 83–92. URL: http://adoi.acm.org/10.1145/2933057.2933092.

[BFK+16b] P. Berenbrink, T. Friedetzky, P. Kling, F. Mallmann-Trenn, and C. Wastell: *Plurality Consensus in Arbitrary Graphs: Lessons Learned from Load Balancing*. In *24th Annual European Symposium on Algorithms ESA'16*. Volume 57. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, 2016, pages 10:1–10:18. URL: http://drops.dagstuhl.de/opus/volltexte/2016/6361.

[BFM+15] P. Berenbrink, T. Friedetzky, F. Mallmann-Trenn, S. Meshkinfamfard, and C. Wastell: *Threshold Load Balancing with Weighted Tasks*. In *2015 IEEE International Parallel and Distributed Processing Symposium, IPDPS'15*, 2015, pages 550–558. URL: http://dx.adoi.org/10.1109/IPDPS.2015.73.

[BGKM16] P. Berenbrink, G. Giakkoupis, A. Kermarrec, and F. Mallmann-Trenn: *Bounds on the Voter Model in Dynamic Networks*. In *43rd International Colloquium on Automata, Languages, and Programming, ICALP'16*, 2016, pages 146:1–146:15. URL: http://dx.adoi.org/10.4230/LIPIcs.ICALP.2016.146.

[BKSS13] P. Berenbrink, K. Khodamoradi, T. Sauerwald, and A. Stauffer: *Balls-into-bins with Nearly Optimal Load Distribution*. In *Proceedings of the Twenty-fifth Annual ACM Symposium on Parallelism in Algorithms and Architectures*. SPAA. ACM, 2013, pages 326–335. URL: http://doi.acm.org/10.1145/2486159.2486191.

[BKK+15] P. Berenbrink, R. Klasing, A. Kosowski, F. Mallmann-Trenn, and P. Uznanski: *Improved Analysis of Deterministic Load-Balancing Schemes*. In *Proceedings of the 2015 ACM Symposium on Principles of Distributed Computing, PODC'15*, 2015, pages 301–310. URL: http://adoi.acm.org/10.1145/2767386.2767413.

[BKM14] P. Berenbrink, B. Krayenhoff, and F. Mallmann-Trenn: *Estimating the number of connected components in sublinear time*. In *Inf. Process. Lett.* volume 114 (11), 2014, pages 639–642. URL: http://dx.adoi.org/10.1016/j.ipl.2014.05.008.

[BDJ98] M. Berg, F. van der Duyn Schouten, and J. Jansen: *Optimal Batch Provisioning to Customers Subject to a Delay-Limit*. In *Management Science*, volume 44 (5), 1998, pages 684–697. URL: http://www.jstor.org/stable/2634473.

[BCR06] A. Blum, T. Chan, and M. Rwebangira: *A Random-Surfer Web-Graph Model*. In *Proceedings of the Third Workshop on Analytic Algorithmics and Combinatorics, ANALCO 2006*, 2006, pages 238–246. URL: http://dx.doi.org/10.1137/1.9781611972962.8.

[BGPS06] S. P. Boyd, A. Ghosh, B. Prabhakar, and D. Shah: *Randomized gossip algorithms*. In *IEEE Transactions on Information Theory*, volume 52 (6), 2006, pages 2508–2530.

[BMPS04] S. Brahma, S. Macharla, S. P. Pal, and S. K. Singh: *Fair Leader Election by Randomized Voting*. In *Proc. ICDCIT '04*, 2004, pages 22–31.

[CC12] L. Cardelli and A. Csikász-Nagy: *The cell cycle switch computes approximate majority*. In *Scientific reports*, volume 2, 2012.

[CFL08] A. Chaintreau, P. Fraigniaud, and E. Lebhar: *Networks Become Navigable as Nodes Move and Forget*. In *Automata, Languages and Programming, 35th International Colloquium, ICALP*, 2008, pages 133–144.

[CM08] P. Chebolu and P. Melsted: *PageRank and the Random Surfer Model*. In *Proceedings of the Nineteenth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA*. Society for Industrial and Applied Mathematics, 2008, pages 1010–1018. URL: http://dl.acm.org/citation.cfm?id=1347082.1347193.

[CDS+13] Y.-J. Chen, N. Dalchau, N. Srinivas, A. Phillips, L. Cardelli, D. Soloveichik, and G. Seelig: *Programmable chemical controllers made from DNA*. In *Nature nanotechnology*, volume 8 (10), 2013, pages 755–762.

[CL03] F. Chung and L. Lu: *The Average Distance in a Random Graph with Given Expected Degrees*. In *Internet Mathematics*, volume 1 (1), 2003, pages 91–113. URL: http://dx.doi.org/10.1080/15427951.2004.10129081.

[CIG+15] A. E. F. Clementi, M. D. Ianni, G. Gambosi, E. Natale, and R. Silvestri: *Distributed community detection in dynamic graphs*. In *Theor. Comput. Sci.* volume 584, 2015, pages 19–41.

[CKM17] V. Cohen-Addad, V. Kanade, and F. Mallmann-Trenn: *Hierarchical Clustering Beyond the Worst-Case*. In, 2017. To appear in NIPS 2017. URL: http://arxiv.org/abs/1702.03959.

[CKMM17] V. Cohen-Addad, V. Kanade, F. Mallmann-Trenn, and C. Mathieu: *Hierarchical Clustering: Objective Functions and Algorithms*. In *CoRR*, volume abs/1704.02147, 2017. URL: http://arxiv.org/abs/1704.02147.

[CMM+98] R. Cole, B. M. Maggs, F. Meyer auf der Heide, M. Mitzenmacher, A. W. Richa, K. Schröder, R. K. Sitaraman, and B. Vöcking: *Randomized Protocols for Low-congestion Circuit Routing in Multistage Interconnection Networks*. In *Proceedings of the Thirtieth Annual ACM Symposium on Theory of Computing*. STOC '98. ACM, 1998, pages 378–388. URL: http://doi.acm.org/10.1145/276698.276790.

[CF03] C. Cooper and A. Frieze: *A general model of web graphs*. In *Random Struct. Algorithms*, volume 22 (3), 2003, pages 311–335. URL: http://dx.doi.org/10.1002/rsa.10084.

[CEOR13] C. Cooper, R. Elsässer, H. Ono, and T. Radzik: *Coalescing Random Walks and Voting on Connected Graphs*. In *SIAM Journal on Discrete Mathematics*, volume 27 (4), 2013, pages 1748–1758.

[CER14] C. Cooper, R. Elsässer, and T. Radzik: *The Power of Two Choices in Distributed Voting*. In *Proc. ICALP*, 2014, pages 435–446.

[CER+15] C. Cooper, R. Elsässer, T. Radzik, N. Rivera, and T. Shiraga: *Fast Consensus for Voting on General Expander Graphs*. In *DISC*, 2015, pages 248–262.

[CF05] C. Cooper and A. M. Frieze: *The Cover Time of Random Regular Graphs*. In *SIAM J. Discrete Math.* volume 18 (4), 2005, pages 728–740.

[CRRS16] C. Cooper, T. Radzik, N. Rivera, and T. Shiraga: *Fast plurality consensus in regular expanders*. In *CoRR*, volume abs/1605.08403, 2016. URL: http://arxiv.org/abs/1605.08403.

[CR16] C. Cooper and N. Rivera: *The Linear Voting Model*. In *ICALP*. Volume 55, 2016, pages 144:1–144:12.

[Cox89] J. Cox: *Coalescing Random Walks and Voter Model Consensus Times on the Torus in $\mathbb{Z}^d$*. In *The Annals of Probability*, volume 17 (4), 1989, pages 1333–1366.

[CG14] J. Cruise and A. Ganesh: *Probabilistic consensus via polling and majority rules*. In *Queueing Systems*, volume 78 (2), 2014, pages 99–120.

[CS97] A. Czumaj and V. Stemann: *Randomized allocation processes*. In *Foundations of Computer Science, 1997. Proceedings., 38th Annual Symposium on*, 1997, pages 194–203.

[Czu98] A. Czumaj: *Recovery Time of Dynamic Allocation Processes*. In *Proceedings of the Tenth Annual ACM Symposium on Parallel Algorithms and Architectures*. SPAA. ACM, 1998, pages 202–211. URL: http://doi.acm.org/10.1145/277651.277686.

[CS10] A. Czumaj and C. Sohler: *Testing Expansion in Bounded-Degree Graphs*. In *Combinatorics, Probability & Computing*, volume 19 (5-6), 2010, pages 693–709.

[DGH+87] A. Demers, D. Greene, C. Hauser, W. Irish, J. Larson, S. Shenker, H. Sturgis, D. Swinehart, and D. Terry: *Epidemic algorithms for replicated database maintenance*. In *Proc. of the 6th Ann. ACM Symposium on Principles of Distributed Computing (PODC)*. ACM, 1987, pages 1–12.

[Dev87] L. Devroye: *Branching processes in the analysis of the heights of trees*. English. In *Acta Informatica*, volume 24 (3), 1987, pages 277–298. URL: http://dx.doi.org/10.1007/BF00265991.

[DG13] B. Doerr and L. A. Goldberg: *Adaptive Drift Analysis*. In *Algorithmica*, volume 65 (1), 2013, pages 224–250. URL: https://doi.org/10.1007/s00453-011-9585-3.

[DGM+11] B. Doerr, L. A. Goldberg, L. Minder, T. Sauerwald, and C. Scheideler: *Stabilizing consensus with the power of two choices*. In *Proceedings of the 23rd Annual ACM Symposium on Parallelism in Algorithms and Architectures, (SPAA)*, 2011, pages 149–158.

[Dow55] F. Downton: *Waiting Time in Bulk Service Queues*. In *Journal of the Royal Statistical Society. Series B (Methodological)*, volume 17 (2), 1955, pages 256–261. URL: http://www.jstor.org/stable/2983959.

[DM10] M. Draief and L. Massoulié: *Epidemics and Rumours in Complex Networks*, 1st. Cambridge University Press, 2010.

[DV12] M. Draief and M. Vojnovic: *Convergence Speed of Binary Interval Consensus*. In *SIAM J. Control and Optimization*, volume 50 (3), 2012, pages 1087–1109.

[DR98] D. P. Dubhashi and D. Ranjan: *Balls and bins: A study in negative dependence*. In *Random Struct. Algorithms*, volume 13 (2), 1998, pages 99–124.

[DP09] D. Dubhashi and A. Panconesi: *Concentration of Measure for the Analysis of Randomized Algorithms*, 1st. Cambridge University Press, 2009.

[ER09] K. Efremenko and O. Reingold: *How Well Do Random Walks Parallelize?* In *RANDOM*, 2009, pages 476–489.

[EFK+16] R. Elsässer, T. Friedetzky, D. Kaaser, F. Mallmann-Trenn, and H. Trinker: *Efficient k-Party Voting with Two Choices*. In *CoRR*, volume abs/1602.04667, 2016. URL: http://arxiv.org/abs/1602.04667.

[ES11] R. Elsässer and T. Sauerwald: *Tight bounds for the cover time of multiple random walks*. In *Theor. Comput. Sci.* volume 412 (24), 2011, pages 2623–2641.

[EG16] P. Eschenfeldt and D. Gamarnik: *Supermarket Queueing System in the Heavy Traffic Regime. Short Queue Dynamics*. In *ArXiv e-prints*, 2016. arXiv: 1610.03522 [math.PR].

[EGA+04] S. Eubank, H. Guclu, V. S. Anil Kumar, M. V. Marathe, A. Srinivasan, Z. Toroczkai, and N. Wang: *Modelling disease outbreaks in realistic urban social networks*. In *Nature*, volume 429 (6988), 2004, pages 180–184. URL: http://dx.doi.org/10.1038/nature02541.

[FMM95] G. Fayolle, V. Malyshev, and M. Menshikov: *Topics in the Constructive Theory of Countable Markov Chains*. Cambridge University Press, 1995. URL: https://books.google.ca/books?id=lTJltFEnnHcC.

[FHK15] O. Feinerman, B. Haeupler, and A. Korman: *Breathe before speaking: efficient information dissemination despite noisy, limited and anonymous communication*. In *Distributed Computing*, 2015.

[Fel68] W. Feller: *An Introduction to Probability Theory and Its Applications*. Vol. 1. Wiley, 1968. URL: http://www.amazon.ca/exec/obidos/redirect?tag=citeulike04-20%7B%5C&%7Dpath=ASIN/0471257087.

[FN16] P. Fraigniaud and E. Natale: *Noisy Rumor Spreading and Plurality Consensus*. In *Proceedings of the 2016 ACM Symposium on Principles of Distributed Computing*. PODC '16. ACM, 2016, pages 127–136. URL: http://doi.acm.org/10.1145/2933057.2933089.

[FPM+02] N. R. Franks, S. C. Pratt, E. B. Mallon, N. F. Britton, and D. J. Sumpter: *Information flow, opinion polling and collective intelligence in house–hunting social insects*. In *Philosophical Transactions of the Royal Society of London B: Biological Sciences*, volume 357 (1427), 2002, pages 1567–1583.

[GS17] L. Gasieniec and G. Stachowiak: *Fast Space Optimal Leader Election in Population Protocols*. In *ArXiv e-prints*, 2017. arXiv: 1704.07649 [cs.DC].

[GL17] M. Ghaffari and J. Lengler: *Tight Analysis for the 3-Majority Consensus Dynamics*. In *CoRR*, volume abs/1705.05583, 2017. URL: http://arxiv.org/abs/1705.05583.

[GP16] M. Ghaffari and M. Parter: *A polylogarithmic gossip algorithm for plurality consensus*. In *Proceedings of the 2016 ACM Symposium on Principles of Distributed Computing*. ACM, 2016, pages 117–126.

[GM96] B. Ghosh and S. Muthukrishnan: *Dynamic Load Balancing by Random Matchings*. In *J. Comput. Syst. Sci.* volume 53 (3), 1996, pages 357–370.

[Gon81] G. H. Gonnet: *Expected Length of the Longest Probe Sequence in Hash Code Searching*. In *J. ACM*, volume 28 (2), 1981, pages 289–304. URL: http://doi.acm.org/10.1145/322248.322254.

[HR90] T. Hagerup and C. Rüb: *A Guided Tour of Chernoff Bounds*. In *Information Processing Letters*, volume 33 (6), 1990, pages 305–308.

[Haj82] B. Hajek: *Hitting-Time and Occupation-Time Bounds Implied by Drift Analysis with Applications*. In *Advances in Applied Probability*, volume 14 (3), 1982, pages 502–525.

[HP01] Y. Hassin and D. Peleg: *Distributed Probabilistic Polling and Applications to Proportionate Agreement*. In *Information and Computation*, volume 171 (2), 2001, pages 248–268.

[Jac05] M. O. Jackson: *The economics of social networks*. In, 2005.

[Jac+08] M. O. Jackson et al.: *Social and economic networks*. Vol. 3. Princeton University Press Princeton, 2008.

[Joh13] J. P. Johnson: *Targeted advertising and advertising avoidance*. In *The RAND Journal of Economics*, volume 44 (1), 2013, pages 128–144. URL: http://dx.doi.org/10.1111/1756-2171.12014.

[JK77] N. L. Johnson and S. Kotz: *Urn Models and Their Application: An Approach to Modern Discrete Probability Theory*. Wiley, 1977.

[KMN16] D. Kaaser, F. Mallmann-Trenn, and E. Natale: *On the Voting Time of the Deterministic Majority Process*. In *41st International Symposium on Mathematical Foundations of Computer Science, MFCS'16*, 2016, pages 55:1–55:15. URL: http://dx.adoi.org/10.4230/LIPIcs.MFCS.2016.55.

[Kam96] A. Kamal: *Efficient solution of multiple server queues with application to the modeling of ATM concentrators*. In *INFOCOM. Fifteenth Annual Joint Conference of the IEEE Computer Societies. Networking the Next Generation. Proceedings IEEE*. Volume 1, 1996, pages 248–254 vol.1.

[KLL+16] V. Kanade, R. Levi, Z. Lotker, F. Mallmann-Trenn, and C. Mathieu: *Distance in the Forest Fire Model How far are you from Eve?* In *Proceedings of the Twenty-Seventh Annual ACM-SIAM Symposium on Discrete Algorithms, SODA'16*, 2016, pages 1602–1620. URL: http://dx.adoi.org/10.1137/1.9781611974331.ch109.

[KMS16] V. Kanade, F. Mallmann-Trenn, and T. Sauerwald: *On coalescence time in graphs-When is coalescing as fast as meeting?* In *CoRR*, volume abs/1611.02460, 2016. URL: http://arxiv.org/abs/1611.02460.

[KMV17] V. Kanade, F. Mallmann-Trenn, and V. Verdugo: *How large is your graph?* In *CoRR*, volume abs/1702.03959, 2017. To appear in DISC'17. URL: http://arxiv.org/abs/1702.03959.

[KSSV00] R. Karp, C. Schindelhauer, S. Shenker, and B. Vocking: *Randomized rumor spreading*. In *Proc. of the 41th Ann. IEEE Symp. on Foundations of Computer Science (FOCS)*. IEEE, 2000, pages 565–574.

[KDG03] D. Kempe, A. Dobra, and J. Gehrke: *Gossip-Based Computation of Aggregate Information*. In *Proceedings 44th Symposium on Foundations of Computer Science (FOCS)*, 2003, pages 482–491.

[KCYK12] N. K. Kim, M. L. Chaudhry, B. K. Yoon, and K. Kim: *A Complete and Simple Solution to a Discrete-Time Finite-Capacity BMAP/D/c Queue*. In, 2012.

[Kle01] J. Kleinberg: *Small-World Phenomena and the Dynamics of Information*. In *Advances in Neural Information Processing Systems 14, NIPS*, 2001, pages 431–438.

[KKR+99] J. Kleinberg, R. Kumar, P. Raghavan, S. Rajagopalan, and A. Tomkins: *The Web as a Graph: Measurements, Models, and Methods*. English. In *Computing and Combinatorics*, volume 1627, Lecture Notes in Computer Science. Springer Berlin Heidelberg, 1999, pages 1–17. URL: http://dx.doi.org/10.1007/3-540-48686-0_1.

[KM10] A. Klenke and L. Mattner: *Stochastic ordering of classical discrete distributions.* In *Adv. in Appl. Probab.* volume 2 (42), 2010, pages 392–410.

[KCM12] P. Kling, A. Cord-Landwehr, and F. Mallmann-Trenn: *Slow Down and Sleep for Profit in Online Deadline Scheduling*. In *Design and Analysis of Algorithms - First Mediterranean Conference on Algorithms, MedAlg'12*, 2012, pages 234–247. URL: http://dx.adoi.org/10.1007/978-3-642-34862-4_17.

[KR01] P. Krapivsky and S. Redner: *Organization of growing random networks*. In *Phys. Rev. E*, volume 63, 6 2001, pages 066123. URL: http://link.aps.org/doi/10.1103/PhysRevE.63.066123.

[KLS08] F. Kuhn, T. Locher, and S. Schmid: *Distributed computation of the mode*. In *Proceedings of the 26th Annual ACM Symposium on Principles of Distributed Computing (PODC)*, 2008, pages 15–24.

[KRR+00] R. Kumar, P. Raghavan, S. Rajagopalan, D. Sivakumar, A. Tomkins, and E. Upfal: *Stochastic models for the Web graph*. In *Proceedings. 41st Annual Symposium on Foundations of Computer Science. 2000*, 2000, pages 57–65.

[LS09] S. Lattanzi and D. Sivakumar: *Affiliation networks*. In *Proceedings of the 41st Annual ACM Symposium on Theory of Computing, STOC*, 2009, pages 427–434.

[LW14] P. K. Lehre and C. Witt: *Concentrated Hitting Times of Randomized Search Heuristics with Variable Drift*. In *Proceedings of the 25th International Symposium on Algorithms and Computation (ISAAC)*, 2014, pages 686–697.

[LS16] J. Lengler and A. Steger: *Drift Analysis and Evolutionary Algorithms Revisited*. In *ArXiv e-prints*, 2016. arXiv: 1608.03226 [math.CO].

[LCKF05] J. Leskovec, D. Chakrabarti, J. Kleinberg, and C. Faloutsos: *Realistic, Mathematically Tractable Graph Generation and Evolution, Using Kronecker Multiplication*. In *Knowledge Discovery in Databases: PKDD 2005*, 2005, pages 133–145. URL: http://dx.doi.org/10.1007/11564126_17.

[LKF07] J. Leskovec, J. Kleinberg, and C. Faloutsos: *Graph evolution: Densification and shrinking diameters*. In *ACM Transactions on Knowledge Discovery from Data (TKDD)*, volume 1 (1), 2007, pages 2.

[LPW06] D. Levin, Y. Peres, and E. Wilmer: *Markov chains and mixing times*. American Mathematical Society, 2006.

[LP08] D. A. Levin and Y. Perres: *Markov Chains and Mixing Times*. American Mathematical Society, 2008.

[Lez89] P. Lezaud: *Chernoff-type bound for finite Markov chains*. In *Ann. Appl. Probab.* volume 8 (3), 1989, pages 849–867.

[Lyn89] N. Lynch: *A Hundred Impossibility Proofs for Distributed Computing*. In *Proceedings of the Eighth Annual ACM Symposium on Principles of Distributed Computing*. PODC '89. ACM, 1989, pages 1–28. URL: http://doi.acm.org/10.1145/72981.72982.

[LP15] R. Lyons and Y. Peres: *Probability on Trees and Networks*. Available at http://pages.iu.edu/~rdlyons/. Cambridge University Press, 2015.

[MMV] F. Mallmann-Trenn, C. Mathieu, and V. Verdugo: *Skyline Computation with Noisy Comparisons*. Unpublished manuscript.

[MSS15] A. W. Marcus, D. A. Spielman, and N. Srivastava: *Interlacing Families IV: Bipartite Ramanujan Graphs of All Sizes*. In *FOCS*, 2015, pages 1358–1377.

[MOA11] A. W. Marshall, I. Olkin, and B. C. Arnold: *Inequalities: Theory of Majorization and Its Applications*. Springer Series in Statistics. Springer, 2011.

[MW14] A. Mehrabian and N. Wormald: *It's a Small World for Random Surfers*. In *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques, APPROX/RANDOM*, 2014, pages 857–871. URL: http://dx.doi.org/10.4230/LIPIcs.APPROX-RANDOM.2014.857.

[MNRS14] G. B. Mertzios, S. E. Nikoletseas, C. Raptopoulos, and P. G. Spirakis: *Determining Majority in Networks with Local Interactions and Very Small Local Memory*. In *Automata, Languages, and Programming - 41st International Colloquium, (ICALP)*, 2014, pages 871–882.

[MU05] M. Mitzenmacher and E. Upfal: *Probability and Computing: Randomized Algorithms and Probabilistic Analysis*. Cambridge University Press, 2005.

[Mit01] M. Mitzenmacher: *The Power of Two Choices in Randomized Load Balancing*. In *IEEE Trans. Parallel Distrib. Syst.* volume 12 (10), 2001, pages 1094–1104. URL: http://doi.ieeecomputersociety.org/10.1109/71.963420.

[MRS00] M. Mitzenmacher, A. W. Richa, and R. Sitaraman: *The Power of Two Random Choices: A Survey of Techniques and Results*. In *in Handbook of Randomized Computing*, 2000.

[MS08] D. Mosk-Aoyama and D. Shah: *Fast Distributed Algorithms for Computing Seperable Functions*. In *IEEE Trans. Information Theory*, volume 54 (7), 2008, pages 2997–3007.

[MNT14] E. Mossel, J. Neeman, and O. Tamuz: *Majority dynamics and aggregation of information in social networks*. In *Autonomous Agents and Multi-Agent Systems*, volume 28 (3), 2014, pages 408–429.

[MS10] E. Mossel and G. Schoenebeck: *Reaching Consensus on Social Networks*. In *Innovations in Computer Science - (ICS)*, 2010, pages 214–229.

[Oli12] R. I. Oliveira: *On the coalescence time of reversible random walks*. English. In *Trans. Am. Math. Soc.* volume 364 (4), 2012, pages 2109–2128.

[Oli13] R. I. Oliveira: *Mean field conditions for coalescing random walks*. In *Ann. Probab.* volume 41 (5), 2013, pages 3420–3461.

[PRU06] G. Pandurangan, P. Raghavan, and E. Upfal: *Using PageRank to Characterize Web Structure*. In *Internet Mathematics*, volume 3 (1), 2006, pages 1–20. URL: http://dx.doi.org/10.1080/15427951.2006.10129114.

[PSL80] M. Pease, R. Shostak, and L. Lamport: *Reaching agreement in the presence of faults*. In *Journal of the ACM*, volume 27 (2), 1980, pages 228–234.

[Pel02] D. Peleg: *Local majorities, coalitions and monopolies in graphs: a review*. In *Theor. Comput. Sci.* volume 282 (2), 2002, pages 231–257.

[PR99] R. Pemantle and J. S. Rosenthal: *Moment conditions for a sequence with negative drift to be uniformly bounded in Lr*. In *Stochastic Processes and their Applications*, volume 82 (1), 1999, pages 143–155. URL: http://www.sciencedirect.com/science/article/pii/S0304414999000125.

[PS15] Y. Peres and P. Sousi: *Mixing Times are Hitting Times of Large Sets*. In *Journal of Theoretical Probability*, volume 28 (2), 2015, pages 488–519.

[PTW10] Y. Peres, K. Talwar, and U. Wieder: *The $(1+\beta)$-choice Process and Weighted Balls-into-Bins*. In *Proceedings of the 21st Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*. SODA. Society for Industrial and Applied Mathematics, 2010, pages 1613–1619.

[PVV09] E. Perron, D. Vasudevan, and M. Vojnović: *Using Three States for Binary Consensus on Complete Graphs*. In *28th IEEE International Conference on Computer Communications, (INFOCOM)*, 2009, pages 2527–2535.

[Pik13] T. Piketty: *Le capital au 21e siècle*. Editions du Seuil, 2013.

[PKC13] A. Pinar, T. G. Kolda, and S. Comandur: *The Block Two-level Erdos Renyi (BTER) Graph Model*. In, 2013. URL: http://www.osti.gov/scitech/servlets/purl/1106519.

[RS98] M. Raab and A. Steger: *"Balls into Bins" - A Simple and Tight Analysis*. In *Randomization and Approximation Techniques in Computer Science, Second International Workshop, RANDOM, Barcelona, Spain, October 8-10, 1998, Proceedings*. Lecture Notes in Computer Science, volume 1518, Springer, 1998, pages 159–170. URL: http://dx.doi.org/10.1007/3-540-49543-6_13.

[Rab83] M. O. Rabin: *Randomized byzantine generals*. In *Proc. of the 24th Ann. Symp. on Foundations of Computer Science (SFCS)*. IEEE, 1983, pages 403–409.

[RS+77] Y. Rinott, T. J. Santner, et al.: *An Inequality for Multivariate Normal Probabilities with Application to a Design Problem*. In *The Annals of Statistics*, volume 5 (6), 1977, pages 1228–1234.

[SS12] T. Sauerwald and H. Sun: *Tight Bounds for Randomized Load Balancing on Arbitrary Network Topologies*. In *CoRR*, volume abs/1201.2715, 2012. full version of FOCS. arXiv: 1201.2715 [cs.DM].

[SPK12] C. Seshadhri, A. Pinar, and T. Kolda: *The Similarity Between Stochastic Kronecker and Chung-Lu Graph Models*. In *Proceedings of the Twelfth SIAM International Conference on Data Mining*, 2012, pages 1071–1082. URL: http://dx.doi.org/10.1137/1.9781611972825.92.

[SZ92] K. Sohraby and J. Zhang: *Spectral decomposition approach for transient analysis of multi-server discrete-time queues*. In *INFOCOM. Eleventh Annual Joint Conference of the IEEE Computer and Communications Societies, IEEE*. IEEE, 1992, pages 395–404.

[Ste96] V. Stemann: *Parallel Balanced Allocations*. In *Proceedings of the Eighth Annual ACM Symposium on Parallel Algorithms and Architectures*. SPAA. ACM, 1996, pages 261–269. URL: http://doi.acm.org/10.1145/237502.237565.

[SR03] M. R. Subramani and B. Rajagopalan: *Knowledge-sharing and influence in online social networks via viral marketing*. In *Communications of the ACM*, volume 46 (12), 2003, pages 300–307.

[TW14] K. Talwar and U. Wieder: *Balanced Allocations: A Simple Proof for the Heavily Loaded Case*. English. In *Automata, Languages, and Programming*, volume 8572, Lecture Notes in Computer Science. Springer Berlin Heidelberg, 2014, pages 979–990. URL: http://dx.doi.org/10.1007/978-3-662-43948-7_81.

[TM67] J. Travers and S. Milgram: *The small world problem*. In *Phychology Today*, volume 1, 1967, pages 61–67.

[Vaz01] A. Vazquez: *Disordered networks generated by recursive searches*. In *EPL (Europhysics Letters)*, volume 54 (4), 2001, pages 430.

Appendix A

Probabilistic Preliminaries

A.1 Elementary Bounds

Proposition A.1 (Markov's Inequality)**.** *Let X be a random variable that assumes only nonnegative values. Then, for all $a > 0$*

$$\mathbb{P}[\, X \geq a \,] \leq \frac{\mathbb{E}[\, x \,]}{a}$$

Proposition A.2 (Union Bound)**.** *For any countable set of events $\mathcal{E}_1, \mathcal{E}_2, \ldots, \mathcal{E}_n$ we have*

$$\mathbb{P}\left[\bigcup_{i \in [n]} \mathcal{E}_i \right] \leq \sum_{i \in [n]} \mathbb{P}[\, \mathcal{E}_i \,].$$

A.2 Concentration Inequalities and Drift Analysis

A.2.1 Binomial Distribution

Proposition A.3 (Chernoff bound [MU05, Theorem 4.4 and 4.5])**.** *Let $X = \sum_i X_i$ be the sum of $0/1$ independent random variables. Then,*

1. *for any $\delta > 0$,*
$$\mathbb{P}[\, X \geq (1+\delta)\mathbb{E}[\, X \,]\,] < \left(\frac{e^{\delta}}{(1+\delta)^{1+\delta)}} \right)^{\mathbb{E}[\, X \,]}.$$
2. *for any $0 < \delta \leq 1$,*
$$\mathbb{P}[\, X \geq (1+\delta)\mathbb{E}[\, X \,]\,] \leq e^{-\mathbb{E}[\, X \,]\delta^2/3}.$$
3. *for $R \geq 6E[X]$,*
$$\mathbb{P}[\, X \geq R] \leq 2^{-R}.$$
4. *for $0 < \delta < 1$,*
$$\mathbb{P}[\, X \leq (1-\delta)\mathbb{E}[\, X \,]\,] \leq \left(\frac{e^{-\delta}}{(1-\delta)^{1-\delta)}} \right)^{\mathbb{E}[\, X \,]}.$$
5. *for $0 < \delta < 1$,*
$$\mathbb{P}[\, X \leq (1-\delta)\mathbb{E}[\, X \,]\,] \leq e^{-\mathbb{E}[\, X \,]\delta^2/2}.$$

Theorem A.4 ([HR90, Equation 10])**.** *Let $Y = \sum_{i=1}^{m} Y_i$ be the sum of m i.i.d. random variables with $\mathbb{P}[\, Y_i = 1 \,] = p$ and $\mathbb{P}[\, Y_i = 0 \,] = 1 - p$. We have for any $\alpha \in (0, 1)$ that*

$$\mathbb{P}[\, Y \geq \alpha \cdot m \,] \leq \left(\left(\frac{p}{\alpha} \right)^{\alpha} \left(\frac{1-p}{1-\alpha} \right)^{1-\alpha} \right)^{m}.$$

Theorem A.5 ([ABKU99, Lemma 3.1])**.** *Let $X_1, X_2, \ldots, X_n$ be a sequence of random variables with values in an arbitrary domain and let $Y_1, Y_2, \ldots, Y_n$ be a sequence of binary random variables with the property that $Y_i = Y_i(X_1, \ldots, X_i)$. If $\mathbb{P}[\, Y_i = 1 | X_1, \ldots, X_{i-1} \,] \leq p$, then*

$$\mathbb{P}\Big[\sum Y_i \geq \ell \Big] \leq \mathbb{P}[\, \mathsf{Bin}(n, p) \geq \ell \,] \tag{A.1}$$

and, similarly, if $\mathbb{P}[\,Y_i = 1 | X_1, \ldots, X_{i-1}\,] \geq p$, *then*

$$\mathbb{P}\Big[\sum Y_i \leq \ell\Big] \leq \mathbb{P}[\,\mathsf{Bin}(n,p) \leq \ell\,]. \tag{A.2}$$

Here, $\mathsf{Bin}(n,p)$ *denotes the binomial distribution with parameters* n *and* p.

We adapt Theorem 2 and equation (6.7) from [Fel68] as follows.

Theorem A.6 (DeMoivre-Laplace limit theorem [Fel68])**.** *Let* X *be a random variable with binomial distribution* $X \sim \mathsf{Bin}(N,p)$. *It holds for any* $x > 0$ *with* $x = \mathrm{o}\Big(N^{1/6}\Big)$ *that*

$$\mathbb{P}\Big[\,X \geq \mathbb{E}[\,X\,] + x \cdot \sqrt{\mathrm{Var}[\,X\,]}\,\Big] = \frac{1}{\sqrt{2\pi}\cdot x} \cdot \exp\Big(-x^2 2\Big) \pm \mathrm{o}(1)\ .$$

A.2.2 Negative Regression

Negative regression is defined as follows.

Definition A.7 (Neg. Regression [DR98, Def. 21])**.** *A vector* $(X_1, X_2, \ldots, X_n)$ *of random variables is said -to satisfy the* negative regression *condition if*

$$\mathbb{E}[\,f(X_l, l \in \mathscr{L}) | X_r = x_r, r \in \mathscr{R}\,]$$

is non-increasing in each x_r *for any disjoint* $\mathscr{L}, \mathscr{R} \subseteq [n]$ *and for any non-decreasing function* f.

Proposition A.8 ([DR98, Lemma 26])**.** *Let* $(X_1, X_2, \ldots, X_n)$ *satisfy the negative regression condition and consider an arbitrary index set* $I \subseteq [n]$ *as well as any family of non-decreasing functions* f_i *(*$i \in \{\, I \,\}$*). Then, we have*

$$\mathbb{E}\Big[\prod_{i\in I} f_i(X_i)\Big] \leq \prod_{i\in I} \mathbb{E}[\,f_i(X_i)\,] \tag{A.3}$$

A.2.3 Poisson Distribution

Lemma A.9. *Let* $X_n \sim \mathsf{Bin}(n, \alpha/n)$ *for all* n. *Let* $Y \sim Poi(e\alpha)$. *Let* $\alpha \leq 1/e$. *For all* n *and* $k \geq 1$ *we have* $Pr(X_n \geq k) \leq Pr(Y \geq k)$.

Proof. Fix an arbitrary n and $k \geq 1$. $Pr(X_n = k) \leq \binom{n}{k}(\alpha/n)^k \leq \alpha^k/k! \leq \frac{\alpha^k}{k!}\frac{e}{e^{e\alpha}} \leq \frac{(e\alpha)^k}{k!}\frac{1}{e^{e\alpha}} = Pr(Y = k)$. □

Lemma A.10 (Poisson tail bound)**.** *Let* $\alpha \geq 1$. *Let* $\alpha' = 1 + \lceil e\alpha \rceil + e\alpha$. *Let* N_i *be sum of* N_{i-1} *independent* $(1 + \lceil e\alpha \rceil + \mathsf{Poisson}(e\alpha))$*-distributed random variables. Let* $b_i = (6\alpha')^i(k+1)$. *Then,* $\mathbb{P}[\,N_i \geq b_i | N_{i-1} < b_{i-1}\,] \leq 2^{-6e\alpha b_{i-1}}$.

Proof. Due to the independence of the random variables, we have $N_{i-1} \cdot \mathsf{Poisson}(e\alpha) = \mathsf{Poisson}(e\alpha N_{i-1})$. In the following we fix $N_{i-1} = n$ for $n < b_{i-1}$. We derive, by using the

definition b_i and by applying the Poisson tail bound given in [MU05],

$$\begin{aligned}\mathbb{P}[\,N_i \geq b_i | n < b_{i-1}\,] &\leq \mathbb{P}[\,\mathsf{Poisson}(e\alpha n) + n(1 + \lceil e\alpha \rceil) \geq b_i | n < b_{i-1}\,] \\ &\leq \mathbb{P}[\,\mathsf{Poisson}(e\alpha n) \geq (6\alpha')b_{i-1} - b_{i-1}(1 + \lceil e\alpha \rceil) | n < b_{i-1}\,] \\ &= \mathbb{P}[\,\mathsf{Poisson}(e\alpha n) \geq 6e\alpha b_{i-1} | n < b_{i-1}\,] \\ &\leq \frac{e^{-e\alpha n} \cdot (e^2\alpha n)^{6e\alpha b_{i-1}}}{(6e\alpha b_{i-1})^{6e\alpha b_{i-1}}} \leq \frac{1 \cdot (e^2\alpha b_{i-1})^{6e\alpha b_{i-1}}}{(6e\alpha b_{i-1})^{6e\alpha b_{i-1}}} \leq 2^{-6e\alpha b_{i-1}}.\end{aligned}$$

□

A.2.4 Hajek's Theorem

Theorem A.11 (Hajek's Theorem – Simplified version of [Haj82, Theorem 2.3])**.** *Let $(Y(t))_{t\geq 0}$ be a sequence of random variables on a probability space $(\Omega, \mathcal{F}, P)$ with respect to the filtration $(\mathcal{F}(t))_{t\geq 0}$. Assume the following two conditions hold:*

(i) (Majorization) There exists a random variable Z and a constant $\lambda' > 0$, such that $\mathbb{E}\left[\, e^{\lambda' Z}\,\right] \leq D$ for some finite D, and $(|Y(t+1) - Y(t)||\mathcal{F}(t)) \leq^{st} Z$ for all $t \geq 0$; and

(ii) (Negative Bias) There exist $a, \varepsilon_0 > 0$, such for all t we have

$$\mathbb{E}[\,Y(t+1) - Y(t) | \mathcal{F}(t), Y(t) > a\,] \leq -\varepsilon_0.$$

Let $\eta = \min\{\lambda', \varepsilon_0 \cdot \lambda'^2/(2D), 1/(2\varepsilon_0)\}$. Then, for all b and t we have

$$\mathbb{P}[\,Y(t) \geq b | \mathcal{F}(0)\,] \leq e^{\eta(Y(0)-b)} + \frac{2D}{\varepsilon_0 \cdot \eta} \cdot e^{\eta(a-b)}.$$

Proof. The statement of the theorem provided in [Haj82] requires besides (i) and (ii) to choose constants η, and ρ such that $0 < \rho \leq \lambda'$, $\eta < \varepsilon_0/c$ and $\rho = 1 - \varepsilon_0 \cdot \eta + c\eta^2$ where

$$c = \frac{\mathbb{E}\left[\, e^{\lambda' Z}\,\right] - (1 + \lambda'\mathbb{E}[\,Z\,])}{\lambda'^2} = \sum_{k=2}^{\infty} \frac{\lambda'^{k-2}}{k!} \mathbb{E}\left[\, Z^k\,\right].$$

With these requirements it then holds that for all b and t

$$\mathbb{P}[\,Y(t) \geq b | \mathcal{F}(0)\,] \leq \rho^t e^{\eta(Y(0)-b)} + \frac{1-\rho^t}{1-\rho} \cdot D \cdot e^{\eta(a-b)}. \tag{A.4}$$

In the following we bound (A.4) by setting $\eta = \min\{\,\lambda', \varepsilon_0 \cdot \lambda'^2/(2D), 1/(2\varepsilon_0)\,\}$. The following upper and lower bound on ρ follow.

- $\rho = 1 - \varepsilon_0 \cdot \eta + c\eta^2 \leq 1 - \varepsilon_0 \cdot \eta + \varepsilon_0 \cdot \eta \cdot c \cdot \lambda'^2/(2D) \leq 1 - \varepsilon_0 \cdot \eta + \varepsilon_0 \cdot \eta/2 = 1 - \varepsilon_0 \cdot \eta/2$, where we used $c \leq D/\lambda'^2$.
- $\rho = 1 - \varepsilon_0 \cdot \eta + c\eta^2 \geq 1 - \varepsilon_0/(2\varepsilon_0) \geq 0$.

We derive, from (A.4) using that for any $t \geq 0$ we have $0 \leq \rho^t < 1$

$$\mathbb{P}[\,Y(t) \geq b|\mathcal{F}(0)\,] \leq \rho^t e^{\eta(Y(0)-b)} + \frac{1-\rho^t}{1-\rho} \cdot D \cdot e^{\eta(a-b)} \leq e^{\eta(Y(0)-b)} + \frac{1}{1-\rho} \cdot D \cdot e^{\eta(a-b)}$$
$$\leq e^{\eta(Y(0)-b)} + \frac{2D}{\varepsilon_0 \cdot \eta} \cdot e^{\eta(a-b)}, \tag{A.5}$$

since $\frac{1}{(1-\rho)} \leq \frac{2}{\varepsilon_0 \cdot \eta}$. This yields the claim. □

A.2.5 Drift Theorem

The following is a version of the multiplicative drift theorem which we will use in Lemma 11.12 to derive a bound on the number of required periods until all nodes agree on one opinion.

Theorem A.12 ([LS16, Theorem 5])**.** *Let $(X_t)_{t\in\mathbb{N}_0}$ be a Markov chain with state space $\mathcal{S} \subseteq \{0\} \cup [1,\infty)$ and with $X_0 = n$. Let T be the random variable that denotes the earliest point in time $t \geq 0$ such that $X_t = 0$. Assume that there is $\delta > 0$ such that for all $x \in \mathcal{S}$*

$$\mathbb{E}[\,X_{t+1} \mid X_t = x\,] \leq (1-\delta)x \ .$$

Then

$$\mathbb{P}\left[\,T > \left\lceil \frac{\log n + k}{|\log(1-\delta)|} \right\rceil\right] \leq e^{-k} \ .$$

Theorem A.13 (Variable Drift Theorem [LW14, Corollary 1.(i)])**.** *Let $(X_t)_{t\geq 0}$, be a stochastic process over some state space $S \subseteq \{0\} \cup [x_{\min}, x_{\max}]$, where $x_{\min} \geq 0$. Let $h\colon [x_{\min}, x_{\max}] \to \mathbb{R}^+$ be a differentiable function. Then the following statements hold for the first hitting time $T \coloneqq \min\{t \mid X_t = 0\}$. If $\mathbb{E}[\,X_{t+1} - X_t \mid \mathcal{F}_t; X_t \geq x_{\min}\,] \leq -h(X_t)$ and $\frac{d}{dx}h(x) \geq 0$, then*

$$\mathbb{E}[\,T \mid X_0\,] \leq \frac{x_{\min}}{h(x_{\min})} + \int_{x_{\min}}^{X_0} \frac{1}{h(y)}\,\mathrm{d}y.$$

Lemma A.14. *Let $(X_t)_{t\geq 0}$ be a stochastic process satisfying (i) $\mathbb{E}[\,X_t|\mathcal{F}_{t-1}\,] \leq \beta \cdot X_{t-1}$, for some $\beta < 1$, and (ii) $X_t \geq 0$ for all $t \geq 0$. Let $\tau(g) = \min\{t \geq 0|X_t \leq g\}$ for $g \in (0, |X_0|)$, then*

$$\mathbb{E}[\,\tau(g)\,] \leq 2 \cdot \left\lceil \log_\beta(g/(2X_0)) \right\rceil.$$

Proof. By the iterative law of expectation, we have

$$\mathbb{E}[\,X_t\,] \leq \beta^t \cdot X_0.$$

Furthermore, by Markov's inequality, for any $\lambda \geq 1$

$$\mathbb{P}\left[\,\tau(g) > \lambda \cdot \left\lceil \log_\beta(g/(2X_0)) \right\rceil\,\right] \leq \mathbb{P}\left[\,X_{\lambda \cdot \lceil \log_\beta(g/(2X_0)) \rceil} > g\,\right] \leq \frac{\mathbb{E}\left[\,X_{\lambda \cdot \lceil \log_\beta(g/(2X_0)) \rceil}\,\right]}{g} \leq 2^{-\lambda}.$$

Therefore,

$$\begin{aligned}\mathbb{E}[\,\tau(g)\,] &= \sum_{i=1}^{\infty} \mathbb{P}[\,\tau(g) \geq i\,] \\ &\leq \left\lceil \log_\beta(g/(2X_0)) \right\rceil + \sum_{\lambda=1}^{\infty} \left\lceil \log_\beta(g/(2X_0)) \right\rceil \cdot \mathbb{P}\Big[\,\tau(g) > \lambda \cdot \left\lceil \log_\beta(g/(2X_0) \right\rceil\,\Big] \\ &\leq 2 \cdot \left\lceil \log_\beta(g/(2X_0)) \right\rceil.\end{aligned}$$

□

A.2.6 Doob-Martingale

We thus consider the process of uncovering $Y_v(\tau)$ one node after the other in order to obtain the Doob martingale of $Y(\tau)$ as follows. We define the sequence $Z_j(\tau)$ as $Z_j(\tau) = \mathbb{E}[\,Y(\tau)|T_j(\tau), \ldots, T_1(\tau)\,]$ with $Z_0(\tau) = \mathbb{E}[\,Y(\tau)\,]$. We have

$$\mathbb{E}[\,Z_j(\tau)|T_{j-1}(\tau), \ldots, T_1(\tau)\,] = \mathbb{E}[\,\mathbb{E}[\,Y(\tau)|T_j(\tau), \ldots, T_1(\tau)\,]|T_{j-1}(\tau), \ldots, T_1(\tau)\,]$$

which, applying the tower property, gives us that

$$\mathbb{E}[\,Z_j(\tau)|T_{j-1}(\tau), \ldots, T_1(\tau)\,] = \mathbb{E}[\,Y(\tau)|T_{j-1}(\tau), \ldots, T_1(\tau)\,] = Z_{j-1}(\tau).$$

Therefore $Z_j(\tau)$ is indeed the Doob martingale of $Y(\tau)$.

A.2.7 Azuma-Hoeffding Inequality

The following definitions and propositions were introduced in [DP09].

Definition A.15 (Martingale)**.** *A* martingale *is a sequence of random variables $X_0, X_1, \ldots$ such that for all $i \geq 1$*

$$\mathbb{E}[\,X_i \mid X_0, X_1, \ldots, X_{i-1}\,] = X_{i-1}.$$

Definition A.16. *The* Doob sequence *of a function f w.r.t. a sequence of random variables $X_1, \ldots, X_n$ is defined by*

$$Y_i := \mathbb{E}[\,f|X_i\,], i \in \{0, 1, \ldots, n\}.$$

Proposition A.17. *The Doob sequence of a function defines a martingale, i. e.,*

$$\mathbb{E}[\,Y_i \mid X_{i-1}\,] = Y_{i-1}, i \in \{0, 1, \ldots, n\}.$$

Proposition A.18 (Azuma-Hoeffding Inequality)**.** *Let $Y_0, Y_1, \ldots$ be a martingale with respect to the sequence $X_0, X_1, \ldots$. Suppose also that for $a_i \leq Y_i - Y_{i-1} \leq b_i$ for $i \geq 1$. Then,*

$$\mathbb{P}[\,|Y_n - Y_0| + t\,] \leq \exp\left(-\frac{2t^2}{\sum_{i\in[n]}(b_i - a_i)^2}\right).$$

A.3 Markov Chains

A.3.1 Couplings

The following lemma will be helpful to define a coupling between distributions that are close to the stationary distribution and the exact stationary distribution. (A very similar lemma has been derived in [ES11, Lemma 2.8])

Lemma A.19. *Let $\varepsilon \in (0,1]$ be an arbitrary value. Let Z_1 and Z_2 be two probability distributions over $\{1,\dots,n\}$ so that $\mathbb{P}[\,Z_1 = i\,] \geq \epsilon \cdot \mathbb{P}[\,Z_2 = i\,]$ for every $1 \leq i \leq n$. Then, there is a coupling $(\tilde{Z}_1, \tilde{Z}_2)$ of (Z_1, Z_2) and an event $\mathcal{E}$ with $\mathbb{P}[\,\mathcal{E}\,] \geq \epsilon$ so that*

$$\mathbb{P}\Big[\,\tilde{Z}_1 = i \mid \mathcal{E}\,\Big] = \mathbb{P}\Big[\,\tilde{Z}_2 = i\,\Big] \qquad \text{for every } 1 \leq i \leq n.$$

Proof. Let $U \in [0,1]$ be a uniform random variable. We next define our coupling $(\tilde{Z}_1, \tilde{Z}_2)$ of Z_1 and Z_2 that will depend on the outcome of U. First, if $U \in [0, \varepsilon)$, then we set

$$\tilde{Z}_1 = \tilde{Z}_2 = i, \qquad \text{if } i \text{ satisfies } \varepsilon \cdot \textstyle\sum_{k=1}^{i-1} \mathbb{P}[\,Z_2 = k\,] \leq U < \varepsilon \cdot \sum_{k=1}^{i} \mathbb{P}[\,Z_2 = k\,].$$

For the case where $U \in (\varepsilon, 1)$, it is clear that the definition of U can be extended in a way so that $\tilde{Z}_1$ has the same distribution as Z_1, and $\tilde{Z}_2$ has the same distribution as Z_2. Furthermore, notice that if $U \in [0,\varepsilon)$ happens, then $\tilde{Z}_1$ has the same distribution as Z_2, and $\tilde{Z}_1 = \tilde{Z}_2$. Observing that $\mathbb{P}[\,U \in [0,\varepsilon)\,] = \varepsilon$ completes the proof. □

A.3.2 Strassen's Theorem

Proposition A.20 ([MOA11, Proposition 11.E.11],[RS+77])**.** *For $N \in \mathbb{N}$ and a probability vector $\Theta \in [0,1]^l$, consider a random vector X having the multinomial distribution $\mathsf{Mult}(N, \Theta)$. Let*

$$\phi\colon \left\{ x \in \mathbb{N}_0^l \;\middle|\; \sum_{i \in [l]} x_i = N \right\} \to \mathbb{R} \tag{A.6}$$

be such that $\mathbb{E}[\,\phi(X)\,]$ exists. Note that this expected value depends on Θ. Define the function ψ on probability vectors as $\psi(\Theta) \coloneqq \mathbb{E}[\,\phi(X)\,]$. If ϕ is Schur-convex, then so is ψ.

Theorem A.21 (Strassen's Theorem [MOA11, 17.B.6])**.** *Suppose that $\mathcal{A} \subseteq \mathbb{R}^n$ is closed and that $\leq_{\mathcal{C}}$ is the preorder of $\mathcal{A}$ generated by the convex cone $\mathcal{C}$ of real-valued functions defined on $\mathcal{A}$. Suppose further that $\{\,(x,y) \mid x \leq_{\mathcal{C}} y\,\}$ is a closed set. Then the conditions*

(i) $X \leq_{\mathcal{C}}^{st} Y$ and

(ii) there exists a pair $\tilde{X}, \tilde{Y}$ of random variables such that

(a) X and $\tilde{X}$ are identically distributed, Y and $\tilde{Y}$ are identically distributed and

(b) $\mathbb{P}\Big[\,\tilde{X} \leq_{\mathcal{C}} \tilde{Y}\,\Big] = 1$

are equivalent if and only if $\mathcal{C}^+ = \mathcal{C}^*$; i.e., the stochastic completion $\mathcal{C}^+$ of $\mathcal{C}$ is complete.

A.3.3 Positive Recurrence

A Markov chain $\boldsymbol{X}$ is time-homogeneous (transition probabilities are time-independent), irreducible (every state is reachable from every other state[1]), and aperiodic (path lengths have no period). Recall that such a Markov chain is positive recurrent (or ergodic) if the probability to return to the start state is 1 and the expected return time is finite. In particular, this implies the existence of a unique stationary distribution. Positive recurrence is a standard formalization of the intuitive concept of stability. See [LP08] for an excellent introduction into Markov chains and the involved terminology.

Theorem A.22 (Fayolle et al. [FMM95, Theorem 2.2.4])**.** *A time-homogeneous irreducible aperiodic Markov chain ζ with a countable state space Ω is positive recurrent if and only if there exists a positive function $\phi(x), x \in \Omega$, a number $\eta > 0$, a positive integer-valued function $\beta(x), x \in \Omega$, and a finite set $C \subseteq \Omega$ such that the following inequalities hold:*

1. $E[\phi(\zeta(t+\beta(x))) - \phi(x) | \zeta(t) = x] \leq -\eta\beta(x)$, $x \notin C$
2. $E[\phi(\zeta(t+\beta(x)))|\zeta(t) = x] < \infty$, $x \in C$

A.4 Markov Chains: Random Walks

Proposition A.23 ([LPW06])**.** *Consider two irreducible Markov chains $(X_t)_{t\geq 0}$, $(Y_t)_{t\geq 0}$ with transition matrix P, $X_0 = x$, and $Y_0 = y$. Let $\{(X_t, Y_t)\}$ be a coupling satisfying that if $X_s = Y_s$, then $X_t = Y_t$ for $t \geq s$. Let $T = \min\{t \geq 0 \mid X_t = Y_t\}$. Then $\|p^t(x,\cdot) - p^t(y,\cdot)\|_{TV} \leq \mathbb{P}[\, T > t \,]$.*

Proposition A.24. *Biased Random Walk [Fel68, Chapter XIV.2]] Let $p \in (0, 1/2)$ and $b, s \in \mathbb{N}$. Consider a discrete time Markov chain $(Z_t)_{t\geq 0}$ with state space $\Omega = [0, b]$ where*

- $Z_0 = s \in [0, b]$
- $\mathbb{P}[\, Z_t = i \mid Z_{t-1} = i - 1 \,] = p$ *for* $i \in [1, b-1], t \geq 1$
- $\mathbb{P}[\, Z_t = i \mid Z_{t-1} = i + 1 \,] = 1 - p$ *for* $i \in [1, b-1], t \geq 1$
- $\mathbb{P}[\, Z_t = i \mid Z_{t-1} = i \,] = 1$ *for* $i \in \{0, b\}, t \geq 1$

Let $T = \min\{t \geq 0 \mid Z_t \in \{0, b\}\}$. Then

$$\mathbb{P}[\, Z_T = b \,] = \frac{\left(\frac{1-p}{p}\right)^s - 1}{\left(\frac{1-p}{p}\right)^b - 1}.$$

Observation A.25. *Consider two random walks $(X_t)_{t\geq 0}$ and $(Y_t)_{t\geq 0}$ starting on nodes drawn from the stationary distribution. Fix an arbitrary $t \in \mathbb{N}$. Define the collision-counting random variables $Z_1 = \sum_{i=0}^{\lceil t/2 \rceil} \mathbf{1}_{X_t = Y_t}$, $Z_2 = \sum_{i=\lceil t/2 \rceil + 1}^{t} \mathbf{1}_{X_t = Y_t}$, and $Z = Z_1 + Z_2$. Then $\mathbb{P}[\, Z_1 \geq 1 \mid Z \geq 1 \,] \geq \frac{1}{2}$.*

Proof. Since both nodes start from the stationary distribution, $\mathbb{P}[\, Z_1 \geq 1 \,] \geq \mathbb{P}[\, Z_2 \geq 1 \,]$. By the Union bound, $\mathbb{P}[\, Z \geq 1 \,] \leq \mathbb{P}[\, Z_1 \geq 1 \,] + \mathbb{P}[\, Z_2 \geq 1 \,] \leq 2 \cdot \mathbb{P}[\, Z_1 \geq 1 \,]$. By law of total probability, $\mathbb{P}[\, Z_1 \,] = \mathbb{P}[\, Z_1 \geq 1 \mid Z \geq 1 \,] \cdot \mathbb{P}[\, Z \geq 1 \,]$. Putting everything together yields $\mathbb{P}[\, Z_1 \geq 1 \mid Z \geq 1 \,] \geq \frac{1}{2}$. □

[1] The state space includes all vectors with non-increasing entries over $\mathbb{N}^n$.

A.4.1 Spectral Gap

We will frequently use the following basic fact about lazy random walks, which in fact also holds for arbitrary reversible Markov chains:

Proposition A.26 (cf. [LPW06, Chapter 12])**.** *Let P be the transition matrix of a reversible Markov chain with state space Ω. Then the following statements hold:*

(i) If P is irreducible, then for any two states $x, y \in \Omega$,

$$p^t_{x,y} \leq \pi(y) + \sqrt{\frac{\pi(y)}{\pi(x)}} \cdot \lambda^t,$$

where $\lambda := \max\{\lambda_2, |\lambda_n|\}$ and $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_n$ are the n real eigenvalues of the matrix P.

(ii) If the Markov chain is a non-lazy random walk on a bipartite regular graph with two partitions V_1 and V_2, then for any pair of states x, y in the same partition

$$p^t_{x,y} \leq \frac{2}{n} \cdot \left(1 + (-1)^{t-1}\right) + 2(\max\{\lambda_2, |\lambda_{n-1}|\})^t.$$

Similarly, if x and y are in opposite partitions,

$$p^t_{x,y} \leq \frac{2}{n} \cdot \left(1 + (-1)^{t}\right) + 2(\max\{\lambda_2, |\lambda_{n-1}|\})^t.$$

(iii) If the Markov chain is lazy, then for any state $x \in \Omega$, $p^t_{x,x}$ is non-increasing in t. In particular, $p^t_{x,x} \geq \pi(u)$.

Proof. The first statement can be found in [LPW06, Equation 12.11].

For the second statement, recall the spectral representation [LPW06, Lemma 12.2 (iii)]

$$p^t_{x,y} = \pi(y) + \pi(y) \cdot \sum_{k=2}^{n} u_k(x) \cdot u_k(y) \cdot \lambda_k^t, \tag{A.7}$$

where u_k is the corresponding eigenvector to λ_k. Since all eigenvalues are non-negative, we conclude from (A.7) that $p^t_{x,x}$ is non-increasing in t as needed. Since G is bipartite and regular, it is not difficult to verify that $\lambda_n = -1$ and $u_n(x) = \sqrt{1/n}$ if $x \in V_1$ and $u_n(x) = -\sqrt{1/n}$ if $x \in V_2$ is the corresponding eigenvector. Hence,

$$\begin{aligned}
\left|p^t_{x,y} - \frac{2}{n} \cdot \left(1 + (-1)^{t-1}\right)\right| &\leq \pi(y) \cdot \left|\sum_{k=2}^{n-1} u_k(x) \cdot u_k(y) \cdot \lambda_k^t\right| \\
&\leq \frac{2}{n} \cdot \max_{2 \leq k \leq n-1} \left|\lambda_k^t\right| \cdot \frac{1}{n} \cdot \sum_{k=2}^{n-1} |u_k(x) \cdot u_k(y)| \\
&\leq \frac{2}{n} \cdot \max_{2 \leq k \leq n-1} \left|\lambda_k^t\right| \cdot \sqrt{\sum_{k=2}^{n-1} u_k(x)^2 \cdot \sum_{k=2}^{n-1} u_k(y)^2}
\end{aligned}$$

As in [LPW06, Proof of Theorem 12.3], using the orthonormality of the eigenvectors, we have

$$\sum_{k=2}^{n-1} u_k(x)^2 \leq \sum_{k=2}^{n} u_k(x)^2 \leq n,$$

and the second statement follows if u and v are in the same partition. The case where u and v are in different partitions follows analogously.

For the third statement, first note that by [LPW06, Exercise 12.3], all eigenvalues of the transition matrix M are non-negative. Since all eigenvalues are non-negative, we conclude from (A.7) that $p^t_{x,x}$ is non-increasing in t as needed. Due to this, we get that $p^t_{x,x} \geq \pi(u)$. This can be verified by means of a simple proof by contradiction. The existence of a t with $p^t_{x,x} < \pi(u)$ and the fact that $p^t_{x,x}$ is non-increasing in t implies that the expected number of visits to u during a sufficiently long period of length τ starting from the stationary distribution would be strictly less than $\tau \cdot \pi(u)$. A contradiction. □

A.4.2 Separation Time (t_{sep})

The following lemma is an immediate consequence of Lemma A.19.

Lemma A.27. *Consider a random walk $(X_t)_{t\geq 0}$, starting from an arbitrary but fixed vertex x_0. Then with probability at least $1 - 1/e$, we can couple $X_{4t_{\text{mix}}}$ with the stationary distribution.*

Proof. Consider the random walk $(X_t)_{t\geq 0}$ after step $s := t_{\text{sep}} \leq 4t_{\text{mix}}$. By definition of t_{sep}, $p^s_{u,v} \geq (1-1/e)\pi(v)$. Applying Lemma A.19, where Z_1 is the distribution given by $p^t_{u,v}$ and Z_2 is the stationary distribution shows that with probability at least $1 - 1/e$, X_s has the same distribution as $\boldsymbol{\pi}$. If this is the case, then the same holds for $X_{4t_{\text{mix}}}$ as well. □

Lemma A.28. *For any graph G,*

$$\max\{(1/e)t_{\text{mix}}, t^{\pi}_{\text{meet}}\} \leq t_{\text{meet}} \leq \frac{2}{(1-1/e)^2} \cdot (4t_{\text{mix}} + 2t^{\pi}_{\text{meet}}),$$

and similarly, $t_{\text{coal}} \leq 4 \cdot (4t_{\text{mix}} + 2t^{\pi}_{\text{coal}})$.

Proof. We begin by proving the lower bound on t_{meet}. First, consider two independent random walks $(X_t)_{t\geq 0}$ and $(Y_t)_{t\geq 0}$ that are run for $t = e \cdot t_{\text{meet}}$ time steps. Then, we have

$$\bar{d}(t) = \max_{u,v} \|p^t_{u,\cdot} - p^t_{v,\cdot}\|_{\text{TV}} \leq \mathbb{P}\Big[\cup^t_{s=0} X_s = Y_s\Big] \leq \frac{1}{e},$$

where the first inequality is due to the coupling method [LPW06, Theorem 5.3] and the second inequality follows by Markov's inequality. The above inequality implies $t_{\text{mix}} \leq e \cdot t_{\text{meet}}$. Furthermore, $t^{\pi}_{\text{meet}} \leq t_{\text{meet}}$ holds by definition, and the lower bound follows.

For the upper bound, we divide the two random walks into consecutive epochs of length $\ell := 4t_{\text{mix}} + 2t^{\pi}_{\text{meet}}$. For the statement it suffices to prove that in each such epoch, regardless of the start vertices of the two random walks, a meeting occurs with probability at least $(1-1/e)^2 \cdot 1/2$.

Consider the first random walk $(X_t)_{t\geq 0}$ starting from an arbitrary vertex after $s := 4t_{\text{mix}}$ steps. By Lemma A.27, we obtain that with probability at least $1 - 1/e$, the distribution of X_s is equal to that of a stationary random walk. Similarly, we obtain that with probability

at least $1-1/e$, the distribution of Y_s is equal to that of a stationary distribution. Hence with probability $(1-1/e)^2$, X_s and Y_s are drawn independently from the stationary distribution. In this case, it follows by Markov's inequality that the two random walks meet before step $s + 2t^{\pi}_{\text{meet}}$ with probability at least $1/2$. Overall, we have shown that with probability at least $(1 - 1/e)^2 \cdot 1/2$, a meeting occurs in a single epoch. Since this lower bound holds for every epoch, independent of the outcomes in previous epochs, the upper bound on the expected time t_{meet} follows. The upper bound on t_{coal} in terms of t^{π}_{coal} is shown in exactly the same way. □

A.4.3 Ramanujan graphs

The following is a simple corollary from a recent work by Marcus et al. [MSS15] on the existence of Ramanujan graphs.

Proposition A.29 (cf. [MSS15])**.** *For any integer $d \geq 3$, there are d-regular graphs $H = (V, E)$ with $t_{\text{mix}} = O(\log n / \log(d))$.*

Proof. Marcus et al. [MSS15] show that the existence of a d-regular bipartite Ramanujan graph H such that $\max\{\lambda_2(\widehat{Q}), |\lambda_{n-1}(\widehat{Q})|\} = O(1/\sqrt{d})$, where $\widehat{Q} = \frac{1}{d}A$ is the transition matrix of a non-lazy random walk where A is the adjacency matrix. By the second statement of Proposition A.26, for any pair of states x, y in the same partition

$$\hat{q}^t_{x,y} \leq \frac{2}{n} \cdot \left(1 + (-1)^{t-1}\right) + 2(\max\{\lambda_2, |\lambda_{n-1}|\})^t.$$

Similarly, x and y are in opposite partitions,

$$\hat{q}^t_{x,y} \leq \frac{2}{n} \cdot \left(1 + (-1)^{t}\right) + 2(\max\{\lambda_2, |\lambda_{n-1}|\})^t.$$

Furthermore note that $q^t_{x,y} \geq 2/n$ due to Proposition A.26.(iii) for even (or odd) t depending on whether x and y are in the same partitions.

Fix $t = O(\log n / \log d)$ such that $2\Big(\max\{\lambda_2(\widehat{Q}), |\lambda_{n-1}(\widehat{Q})|\}\Big)^t \leq \frac{1}{20n}$, where we note that such a t exists due to $\max\{\lambda_2(\widehat{Q}), |\lambda_{n-1}(\widehat{Q})|\} = O(1/\sqrt{d})$. We choose s to be the smallest odd integer being greater than $20t$. To translate from the non-lazy random walk $\widehat{Q}$ to a lazy-random walk P, let Z denote the number of non-loops performed by a lazy random walk of length s. Since, the probability for a self-loop is $1/2$ and the number of self-loops is binomially distributed, we have

$$\mathbb{P}[\, Z \geq t \,] \geq 19/20.$$

By symmetry and the fact that s is odd, $\mathbb{P}[\, Z \text{ is even} \,] = \frac{1}{2}$. Hence, by the Union bound,

$$\mathbb{P}[\, Z \text{ is even} \mid Z \geq t \,] \geq \mathbb{P}[\, Z \text{ is even} \cap Z \geq t \,] \geq \mathbb{P}[\, Z \text{ is even} \,] - \mathbb{P}[\, Z < t \,] \geq \frac{9}{20},$$

and similarly, $\mathbb{P}[\,Z \text{ is odd} \mid Z \geq t\,] \geq \frac{9}{20}$. Let V_1 and V_2 be the bipartite partition of V.

$$\begin{aligned}
\|p^s_{u,\cdot} - \boldsymbol{\pi}\|_{\mathrm{TV}} &\leq \mathbb{P}[\,Z < t\,] \cdot 1 + \mathbb{P}[\,Z \geq t\,] \cdot \left(\sum_{v \in V_1} \left| \frac{11}{20} \hat{q}^t_{u,v} - \frac{1}{n} \right| + \sum_{v \in V_2} \left| \frac{11}{20} \hat{q}^t_{u,v} - \frac{1}{n} \right| \right) \\
&\leq \mathbb{P}[\,Z < t\,] \cdot 1 + \mathbb{P}[\,Z \geq t\,] \cdot \left(\sum_{v \in V} \left| \frac{11}{20} \left(\frac{2}{n} + \frac{1}{20n} \right) - \frac{1}{n} \right| \right) \\
&\leq \mathbb{P}[\,Z < t\,] \cdot 1 + \mathbb{P}[\,Z \geq t\,] \cdot \left(\sum_{v \in V} \left| \frac{22}{20n} - \frac{1}{n} \right| + \frac{11}{400} \right) \\
&\leq \frac{1}{20} \cdot 1 + \frac{19}{20} \left(\frac{2}{20} + \frac{11}{400} \right) < 1/e,
\end{aligned}$$

where the first inequality follows from the equations for $p^t_{x,y}$ above. □

Corollary A.30. *Let n_0 be a sufficiently large constant. Let H_n be the graph of Proposition A.29 with n nodes and $d = \lceil \sqrt{n} \rceil$ for $n \geq n_0$. There exists a universal constant C such that $\max_{n \geq n_0} \{t_{\mathrm{sep}}(H_n)\} \leq C$.*

The corollary follows directly from Proposition A.29 and $t_{\mathrm{sep}} \leq 4 t_{\mathrm{mix}}$.

A.4.4 Mixing time bounds via Hitting time

The following result of Peres and Sousi is useful to establish a bound on the mixing time.

Theorem A.31 ([PS15])**.** *For any $\beta < 1/2$, let $t_{\mathrm{hit}}(\beta) = \max_{u, A : \pi(A) \geq \beta} t_{\mathrm{hit}}(u, A)$. Then there exist positive constants c_β and c'_β such that*

$$c'_\beta \cdot t_{\mathrm{hit}}(\beta) \leq t_{\mathrm{mix}}(1/4) \leq c_\beta \cdot t_{\mathrm{hit}}(\beta).$$

A.5 Random Processes

A.5.1 Galton-Watson Trees

The analysis of the Forest Fire Process uses reductions to Galton-Watson branching processes.

Definition A.32 (Galton-Watson Tree)**.** *A Galton-Watson process is a stochastic process $\{X_n\}$ which evolves according to the recurrence formula $X_0 = 1$ and $X_{n+1} = \sum_{j=1}^{X_n} \xi_j^{(n)}$, where $\{\xi_j^{(n)} : n, j \in \mathbb{N}\}$ is a set of i.i.d. natural number-valued random variables.*

The interpretation is as follows: the process builds a random tree. X_n can be thought of as the number of descendants of the root in the n^{th} generation, and $\xi_j^{(n)}$ can be thought of as the number of children (in generation $n+1$) of the j^{th} of these (n^{th} generation) descendants. The recurrence relation states that the number of descendants in the $(n+1)^{st}$ generation is the sum, over all n^{th} generation descendants, of the number of children of that descendant. For more information, see [LP15].

Proposition A.33 ([DM10, Lemma 1.9]). *Consider the Galton-Watson process with* $\xi_j^{(n)} = \xi$ *such that* $\mathbb{E}[\,\xi\,] < 1$. *Then, we have for the total population size* X

$$\mathbb{P}[\,X > k\,] \le e^{-k \cdot h(1)},$$

where $h(x) = \sup_{\theta \ge 0}\{\theta x - \log \mathbb{E}\Big[\, e^{\theta \xi} \,\Big]\}$. *Furthermore, if* $\xi \sim \mathsf{Poisson}(\lambda)$, *then* $h(1) = \log(1/\lambda) - 1 + \lambda$.

A.5.2 Balls-Into-Bins

Theorem A.34 (Raab and Steger [RS98, Theorem 1]). *Let* M *be the random variable that counts the maximum number of balls in any bin, if we throw* m *balls independently and uniformly at random into* n *bins. Then* $\mathbb{P}[\,M > k_\alpha\,] = o(1)$ *if* $\alpha > 1$ *and* $\mathbb{P}[\,M > k_\alpha\,] = 1 - o(1)$ *if* $0 < \alpha < 1$, *where*

$$k_\alpha = \begin{cases} \frac{\log n}{\log \frac{n \log n}{m}} \left(1 + \alpha \frac{\log \log \frac{n \log n}{m}}{\log \frac{n \log n}{m}}\right) & \text{if } \frac{n}{\operatorname{polylog}(n)} \le m \ll n \log n \\ (d_c - 1 + \alpha) \log n & \text{if } m = c \cdot n \log n \text{ for some constant } c \\ \frac{m}{n} + \alpha \sqrt{2 \frac{m}{n} \log n} & \text{if } n \log n \ll m \le n \operatorname{polylog}(n) \\ \frac{m}{n} + \sqrt{2 \frac{m}{n} \log n \Big(1 - \frac{1}{\alpha} \frac{\log \log n}{2 \log n}\Big)} & \text{if } m \gg n (\log n)^3, \end{cases}$$

where d_c *is largest solution of* $1 + x(\log c - \log x + 1) - c = 0$.

A.5.3 Póly-Urn

The Póly urns [JK77]. In this model, we are given an urn containing marbles of two colors, black and white. In every step, one marble is drawn uniformly at random from the urn. Its color is observed, the marble is returned to the urn and one more marble of the same color is added. For any color, the ratio of marbles with that given color over the total number of marbles is a martingale. Formally, the Póly urn process is defined as follows.

Definition A.35 (Pólya Urn Process). *Let* Pólya(α_1, α_2) *with* $\alpha_1, \alpha_2 \in \mathbb{Z}_0^+$ *be the following urn process. At the beginning there are* α_1 *black marbles and* α_2 *white marbles in the urn. The process runs in multiple steps where* $\alpha_1(i)$ *and* $\alpha_2(i)$ *denote the number of black and white marbles in the urn, respectively, for every time step* i. *In every time step* i, *a black marble is added with probability* $\alpha_1(i)/(\alpha_1(i) + \alpha_2(i))$, *and with remaining probability* $\alpha_2(i)/(\alpha_1(i) + \alpha_2(i))$ *a white marble is added.*

Proposition A.36. *Consider a Pólya urn starting with a fraction* $X_1 > 0$ *of white balls. Then, the sequence* $X_1, X_2, \ldots$ *forms a martingale.*

The statement and the proposition are folklore.

Proof. Let ℓ_i denote the number of balls at time step i. Let $\mathcal{E}_i$ be the event that the ball drawn at step i is white. We have

$$\begin{aligned}
\mathbb{E}[\, X_i \mid X_{i-1}, \ldots, X_1 \,] &= \mathbb{E}[\, X_i \mid X_{i-1} \,] \\
&= \frac{\ell_{i-1} X_{i-1} + 1}{\ell_{i-1} + 1} \cdot \mathbb{P}[\, \mathcal{E}_i \mid X_{i-1} \,] + \frac{\ell_{i-1} X_{i-1}}{\ell_{i-1} + 1} \cdot (1 - \mathbb{P}[\, \mathcal{E}_i \mid X_{i-1} \,]) \\
&= \frac{\ell_{i-1} X_{i-1} + 1}{\ell_{i-1} + 1} \cdot X_{i-1} + \frac{\ell_{i-1} X_{i-1}}{\ell_{i-1} + 1} \cdot (1 - X_{i-1}) \\
&= \frac{\ell_{i-1} X_{i-1}^2 + X_{i-1} + \ell_{i-1} X_{i-1} - \ell_{i-1} X_{i-1}^2}{\ell_{i-1} + 1} = X_{i-1}.
\end{aligned}$$

□