

**CONVERGENCE TO NASH EQUILIBRIA IN DISTRIBUTED AND  
SELFISH REALLOCATION PROCESSES**

by

Iman Hajirasouliha

B.Sc., Sharif University of Technology, Tehran, Iran, 2005

A THESIS SUBMITTED IN PARTIAL FULFILLMENT  
OF THE REQUIREMENTS FOR THE DEGREE OF  
MASTER OF SCIENCE  
in the School  
of  
Computing Science

© Iman Hajirasouliha 2007  
SIMON FRASER UNIVERSITY  
Fall 2007

All rights reserved. This work may not be  
reproduced in whole or in part, by photocopy  
or other means, without the permission of the author.

## APPROVAL

**Name:** Iman Hajirasouliha  
**Degree:** MASTER OF SCIENCE  
**Title of thesis:** Convergence to Nash Equilibria in Distributed and Selfish Reallocation Processes

**Examining Committee:** Dr. S. Cenk Sahinalp  
Associate Professor, Computing Science,  
Simon Fraser University,

Chair

---

Dr. Petra Berenbrink,  
Assistant Professor, Computing Science,  
Simon Fraser University,  
Senior Supervisor

---

Dr. Funda Ergun,  
Associate Professor, Computing Science,  
Simon Fraser University,  
Supervisor

---

Dr. Tom Friedetzky,  
Assistant Professor, Computer Science,  
Durham University,  
External Examiner

**Date Approved:**

Aug 31, 2007



SIMON FRASER UNIVERSITY  
LIBRARY

## Declaration of Partial Copyright Licence

The author, whose copyright is declared on the title page of this work, has granted to Simon Fraser University the right to lend this thesis, project or extended essay to users of the Simon Fraser University Library, and to make partial or single copies only for such users or in response to a request from the library of any other university, or other educational institution, on its own behalf or for one of its users.

The author has further granted permission to Simon Fraser University to keep or make a digital copy for use in its circulating collection (currently available to the public at the "Institutional Repository" link of the SFU Library website <[www.lib.sfu.ca](http://www.lib.sfu.ca)> at: <<http://ir.lib.sfu.ca/handle/1892/112>>) and, without changing the content, to translate the thesis/project or extended essays, if technically possible, to any medium or format for the purpose of preservation of the digital work.

The author has further agreed that permission for multiple copying of this work for scholarly purposes may be granted by either the author or the Dean of Graduate Studies.

It is understood that copying or publication of this work for financial gain shall not be allowed without the author's written permission.

Permission for public performance, or limited permission for private scholarly use, of any multimedia materials forming part of this work, may have been granted by the author. This information may be found on the separately catalogued multimedia material and in the signed Partial Copyright Licence.

While licensing SFU to permit the above uses, the author retains copyright in the thesis, project or extended essays, including the right to change the work for subsequent purposes, including editing and publishing the work in whole or in part, and licensing other parties, as the author may desire.

The original Partial Copyright Licence attesting to these terms, and signed by this author, may be found in the original bound copy of this work, retained in the Simon Fraser University Archive.

Simon Fraser University Library  
Burnaby, BC, Canada

# Abstract

We introduce several congestion games and study the speed of convergence to Nash Equilibria under reasonable reallocation protocols. We focus on a particular atomic congestion game, distributed selfish load balancing, in which a set of resources are to be allocated to tasks with selfish agents willing to minimize their own latency. We revisit and improve the previous results for the uniform case where tasks share identical resources, and the latency function of a resource is the number of tasks utilizing it. Moreover we introduce two variations of this setting. In the first variation we consider the case where tasks have different weights and the latency of each resource is the total weights of the tasks utilizing it. Another variation is the case where tasks are identical, but resources have arbitrary latency functions. We give upper bounds for the convergence time of these models, and some examples to justify our protocols.

*I dedicate my work to my beloved parents for all their support*

*“There are 10 kinds of people in this world:  
Those that understand binary, and those that don’t.”*

# Acknowledgments

First and foremost, I am mostly grateful to my senior supervisor Dr. Petra Berenbrink. Petra has supervised all aspects of my professional development during past two years. This work could not be done without her constant support and invaluable remarks. I would like to greatly thank Dr. Funda Ergun and S. Cenk Sahinalp for their strong support during my studies in Simon Fraser and also taking part in the exam committee.

Many thanks to my co-authors and other collaborators in particular Tom Friedetzky, Zengjian Hu, Hossein Jowhari, and Payam Valadkhan, for providing nice ideas and motivating my research.

The support and encouragement of my friends in Simon Fraser University and in particular colleagues in the Network Modelling and Computational Biology Labs, and Algorithms Group has always been a big factor in my achievements. I would like to thank all of them.

I thank my beloved parents for their unconditional love and continuous support throughout the years. This thesis is dedicated to them.

# Contents

<b>Approval</b>	<b>ii</b>
<b>Abstract</b>	<b>iii</b>
<b>Dedication</b>	<b>iv</b>
<b>Quotation</b>	<b>v</b>
<b>Acknowledgments</b>	<b>vi</b>
<b>Contents</b>	<b>vii</b>
<b>List of Tables</b>	<b>ix</b>
<b>List of Figures</b>	<b>x</b>
<b>1 Introduction</b>	<b>1</b>
1.1 Congestion game model . . . . .	3
1.2 Related work . . . . .	5
1.2.1 The complexity of Nash equilibria . . . . .	5
1.2.2 Previous works on convergence to Nash equilibria . . . . .	7
1.3 Natural variations and results . . . . .	9
1.4 The structure of the thesis . . . . .	11



<b>2</b>	<b>Unit Weighted Tasks and Uniform Resources</b>	<b>12</b>
2.1	The $O(\log \log m + n^4)$ convergence time upper bound . . . . .	13
2.2	Reallocation protocol with a slowdown factor . . . . .	14
2.3	Convergence to Nash Equilibrium . . . . .	15
2.4	Lower bounds . . . . .	21
<b>3</b>	<b>Weighted Tasks and Uniform Resources</b>	<b>23</b>
3.1	The reallocation model with weighted tasks . . . . .	24
3.2	Convergence to Nash equilibrium . . . . .	26
3.3	Lower bounds . . . . .	35
<b>4</b>	<b>Resources with Different Latency Functions</b>	<b>37</b>
4.1	The reallocation process with arbitrary latency functions . . . . .	38
4.1.1	Potential function . . . . .	38
4.1.2	Summary of the convergence result . . . . .	38
4.2	Notation and preliminary results . . . . .	39
4.3	Convergence to Nash equilibria . . . . .	43
4.4	Justification . . . . .	47
4.4.1	The maximum slope is the relevant parameter . . . . .	47
<b>5</b>	<b>Future work</b>	<b>49</b>
<b>A</b>	<b>Omitted proofs</b>	<b>50</b>
A.1	Proof of Part (2) of Observation 3.1.2 . . . . .	50
A.2	Proof of Lemma 4.3.5 . . . . .	51
	<b>Bibliography</b>	<b>52</b>

# List of Tables

1.1 Complexity of computing equilibria for various congestion games . . . . .	8
---	---

# List of Figures

4.1 $\beta$ is the relevant parameter for the analysis . . . . .	48
--	----

# Chapter 1

## Introduction

Most of us, as rational drivers, would probably choose the shortest route in order to get to school. Not surprisingly, to select one among all possible routes, we choose our route in a *selfish* manner, taking only the delay experienced by ourselves in to account without caring about the effects of our choice on other commuters. Regarding social welfare, the selfish behavior of the drivers can be quite bad. Apparently the time it takes to travel along a route depends on the congestion of the route and thus if all the drivers stick to a route which was initially the shortest one, and lose their incentive to migrate to other possible routes, the average delay could be very far from the optimal situation. In fact, the empirical and theoretical studies indicate that with the lack of central control and coordination, the average delay of the drivers can be substantially large.

Not surprisingly, in the traffic model, an optimal solution which minimizes the average delay of all the drivers might not be stable. The drivers, which can be modeled as strategic and rational players in a congestion game, might change their actions and reroute to other permissible routes whenever an improvement to their own cost(delay) is possible. The concept of Nash equilibrium [32] seems to be the only reasonable solution for the study of these sorts of systems. A strategy profile for the players is at a Nash equilibrium if the players can not unilaterally improve their cost and thus have no incentive to change their action.

Related to the selfish behavior of the players and the concept of Nash equilibrium, several different models and questions have been addressed so far. In this thesis, our main focus is on the problem of *convergence to Nash equilibria*. We want to find out, what the dynamic behavior of a particular system looks like, and how much time it takes in order to reach or get close to a Nash equilibrium. For the classic load balancing model, where many individual selfish agents assign their tasks to chosen resources, we introduce natural reallocation protocols for the selfish agents, and study the time it takes for the system to reach a Nash equilibrium.

For all three variations of the load balancing model which we study in this thesis, the nature of our reallocation protocols is as following: In each step, every agent samples one resource and compares the cost of its current utilized resource with the cost of the sampled resource. The agent will migrate to the sampled resource with a certain probability, where is it more likely to migrate if the ratio of the current cost to the new cost is large. The migrations are in parallel.

It is worth mentioning that since the migrations are in parallel, naive reallocation protocols may lead to *oscillatory* effects. Consider for example a reallocation protocol which forces the agents to migrate to a lower-cost resource unconditionally (i.e., if an agent observes a resource with a cost lower than her current cost, then she certainly migrates to the new resource.) In a system with only two resources, it is easy to observe that if initially most tasks are assigned to one of the resources, the overload would oscillate indefinitely between two resources. Our protocols avoid such oscillatory effects.

Another important feature of our protocols is that in order to guarantee the fast convergence toward a Nash equilibrium, agents do not need any global information. Each agent only needs to query the load of the *sampled* resource which can be done efficiently, and the sampling phase itself is quite simple.

In the next section we review the formal definition of a congestion game and fundamental properties of it.

## 1.1 Congestion game model

Congestion games, introduced by Rosenthal [34], is a natural class of games in which participating players aim to allocate sets of resources simultaneously. Some of the resources might be shared among several players, and the cost of a resource is a function of the congestion (e.g. the number of players allocating the resource.) Two well studied special cases of congestion games are the parallel link model of [29] or equivalently the distributed selfish load balancing model of [16, 4, 5] and the Wardrop's selfish routing model which has been widely studied in [20, 21, 41, 38]. In the parallel link model, each agents use only one resource and the resource can be chosen from the set of all resources. The Wardrop's routing model is basically a continuous model in which infinite population of selfish agents carries an infinitesimal amount of flow each. In this model each agent may choose from a set of paths and strives to minimize its sustained latency selfishly. We will get back to these two cases of congestion games again.

Formally a congestion game is a tuple  $(N, E, (\Sigma_i)_{i \in N}, (f_e)_{e \in E})$  where  $N$  is the set of  $n$  players,  $E$  is the set of resources,  $\Sigma_i \subseteq 2^E$  is the strategy space, a collection of strategies, of player  $i$ , and  $f_e$  is a cost function associated with resource  $e$ . A strategy profile(state)  $S = (S_1, \dots, S_n)$  is a vector of strategies where  $S_i$  denotes the strategy of player  $i$ . The cost of player  $i$ , say  $c_i$ , is a function of strategy profile given by  $c_i(S) = \sum_{e \in S_i} f_e(n_e(S))$ , where  $n_e(S)$  is the number of players allocating the resource  $e$  in  $S$ (e.g. the congestion on resource  $e$ .)

A congestion game is called *symmetric* if all the players have the same strategy set:  $\Sigma_i = \Sigma$ . Normally the term *asymmetric* refers to all games(including the symmetric ones) [12].

**Definition.** [*Nash equilibrium*] A strategy profile  $S = (S_1, \dots, S_n)$  is a Nash equilibrium if  $\forall i \in N, \forall S' \in (\Sigma_i), c_i(S) \leq c_i(S_1, \dots, S_{i-1}, S'_i, S_{i+1}, \dots, S_n)$ .

It is clear from the definition that a single player can not improve her cost by changing her strategy, when the system is at a Nash equilibrium. Rosenthal shows that a sequence of such improvements by individual players ends up at a Nash equilibrium state in a finite number of steps [34]. A direct result of his potential function argument is that every congestion game admits a pure

Nash equilibrium.

**Remark.** The term "pure" here, means that all the players choose a strategy deterministically, and no randomization is involved (e.g. players do not have mixed strategies). Curious readers may refer to game theory books such as [33], for more information on this topic.

In order to prove the existence of a Nash equilibrium, a simple method can be shown to actually find one. Consider the Rosenthal's potential function,  $\Phi(S)$ , [34] as following,

$$\Phi(S) = \sum_{e \in E} \sum_{j=1}^{n_e(S)} f_e(j)$$

This potential function has the property that if player  $i$  improves her cost by migrating from strategy  $S_i$  to  $S'_i$ , then the change in  $\Phi$  exactly mirrors the player's gain. So after one improvement step by any of the players the potential function decreases. Since the number of all possible strategy profiles is finite, there should be some lower-bound for the potential. Therefore the number of such improvement steps before reaching a Nash equilibrium is finite; and hence we can find a Nash equilibrium.

**Discrete selfish load balancing model** Throughout this thesis, we mostly talk about the *discrete selfish load balancing model*, which is a special case of general congestion games. In this model, we have  $m$  tasks  $b_1, \dots, b_m$  and  $n$  resources. Each task can be assigned to any resources, and the cost (latency) of resource  $i$  depends only on the total weight of the tasks using resource  $i$  and increases with the total weight of the tasks using the resource. The assignment of tasks to resources is represented by a vector  $x(t) = (x_1(t), \dots, x_n(t))$  where  $x_i(t)$  denotes the number of tasks using resource  $i$  in time  $t$ . Note that in this model, tasks are not splittable.

We restrict ourselves to three different models of the distributed load balancing model that we just defined.

- In the *uniform model*, all tasks have unit weights, and all the resources are identical. The cost of a resource is simply the number of users using it.

- In the *weighted tasks model*, tasks have weights, say less than a parameter  $\Delta$ , and resources are identical. The cost (latency) of resource  $R$  is the total weight of the tasks using  $R$ .
- The *arbitrary latency functions model* is much the same as the previous two models, except that we now assume uniform tasks, but each resource  $i$ , has its own latency function,  $f_i$ . In this model  $f_i(x_i(t))$  would be the latency function of resource  $i$  and each task using the resource experiences this cost (latency).

## 1.2 Related work

Congestion games have recently drawn attention from computer scientists. In this section we try to address some of the fundamental results in the literature.

### 1.2.1 The complexity of Nash equilibria

Although the potential function approach we mentioned in the previous section eventually finds a Nash equilibrium, it does not guarantee a fast convergence toward a Nash equilibrium. By applying this approach it might take an exponential number of steps to reach a Nash Equilibrium, even for symmetric congestion games, [18] or the *best-response sequences* algorithms [1]. In a best-response sequence algorithm, when a player changes her strategy, she always switches to an alternative strategy of minimal cost.

In this section we shall briefly address the following question: *Given a congestion game, how hard is to find a Nash equilibrium?* The reader might have already observed the connection of this problem to a local search problem. Given a state  $S$ , we can look at the states which are deviating from  $S$  only in a single player's strategy as a *neighborhood* for  $S$ , and a Nash equilibrium as a *local optimum*.

**General congestion game.** Fabrikant et al. [18] show that the problem of computing a Nash equilibrium is PLS-complete for general congestion games by a reduction from a version of the



MAX 3-SAT problem, so called POSNAE3FLIP.

**Symmetric network congestion game.** In [18] the authors also show that for the special case of symmetric network congestion games, where sets of strategies are simple paths in a given network, and players choose their path from a common set of paths, a Nash equilibrium can be found in polynomial time through a simple reduction to a min-cost flow computation.

**Theorem** There is a polynomial algorithm for finding a Nash equilibrium in symmetric network congestion games. [18]

*Proof.* Given the network  $N$  and the cost functions  $d_e$ , replace each edge in the network with  $n$  parallel edges with costs  $d_e(1), \dots, d_e(n)$ . It is easy to see that any min-cost flow in the new network is a state of game which minimize the Rosenthal's potential function, and thus is a Nash equilibrium.  $\square$

In this thesis, our focus is on the convergence to Nash equilibrium. Having said that computing a Nash equilibrium in a congestion game might be PLS-complete, we would really like to see for which class of the congestion games, there is a *polynomial* time algorithm to compute a Nash equilibrium.

**Singleton congestion games** Jeong et al. [28] show that in *singleton* congestion games, all improvement sequences have length  $O(m \cdot n^2)$ , where  $n$  is the number of players, and  $m$  is the number of resources. Singleton games are special cases of congestion games in which the strategy profile of the players only contains single resources. Note that our discrete load balancing model is a special case of singleton congestion games. On the other hand, reallocation algorithms differ with the improvement sequences algorithms in the way that in ours randomization is involved, players change their strategies in parallel, and players do not have huge amount of global knowledge. We will get back to these issues later.

**Matroid congestion game** Ackermann et al. [1] study the impact of combinatorial structure on congestion games. They generalize the result of Jeong et al. [28] to a *matroid congestion game* and show that for matroid games, either symmetric or asymmetric, the complexity of computing equilibria is  $O(n^2 \cdot m^2)$ , where  $n$  is the number of players and  $m$  is the number of resources.

A matroid itself can be defined as a combinatorial object. Suppose there is a ground set  $\mathcal{R} = \{1, 2, \dots, m\}$ , and let  $\mathcal{I}$  be a family of subsets of  $\mathcal{R}$ . (e.g.  $\mathcal{I} \subseteq 2^{\mathcal{R}}$ ). A pair  $(\mathcal{R}, \mathcal{I})$  is a matroid such that, if  $I \in \mathcal{I}$  and  $J \subseteq I$ , then  $J \in \mathcal{I}$ , and, if  $I, J \in \mathcal{I}$  and  $|J| \leq |I|$ , then there exists an  $i \in I \setminus J$  with  $J \cup i \in \mathcal{I}$ . Given a matroid  $\mathcal{M} = (\mathcal{R}, \mathcal{I})$ , a *basis* of  $\mathcal{M}$  is a maximal subset of  $\mathcal{R}$  which is a member of  $\mathcal{I}$ . It is easy to observe that all basis of a matroid have the same cardinality. The cardinality is called *rank* of the matroid. A game is called a matroid congestion game, if for every player  $i$ , the strategy space  $i$ , is a bases of a matroid over the set of resources.

Given a matroid game  $\Gamma$ , Ackermann et al. [1] show that, all best response improvements sequences have length  $O(n^2 m \cdot \text{rank}(\Gamma))$ , where rank of a matroid congestion game is defined as the maximum matroid rank over all the players. They also show that their result is tight and the matroid property is a necessity.

We summarize these complexity results in table 1.1.

Recently Skopalik and Vocking [39] present inapproximability results on congestion games. They show that for any poly-time computable  $\alpha > 1$ , finding and  $\alpha$ -approximate Nash equilibrium in general congestion games with positive and increasing delay functions is PLS-hard. They also show that, for every  $n \in \mathbb{N}$ , there is a congestion game with  $n$  players having a state with the property that every sequence of improvement steps leading from this state to an approximate equilibrium has exponential length in  $n$ .

### 1.2.2 Previous works on convergence to Nash equilibria

Goldberg [25] considers a reallocation protocol in which tasks select alternative resources at random. In his protocol, tasks migrate sequentially in continuous time. The continuous time implies that only

Table 1.1: Complexity of computing equilibria for various congestion games

Class of congestion game	Symmetry	Result
General	symmetric and asymmetric	PLS-complete [18]
Singleton	symmetric and asymmetric	$O(n^2 \cdot m)$ [28]
Matroid	symmetric and asymmetric	$O(n^2 \cdot m^2)$ [1]
Network	symmetric	Polynomial [18]
Network	asymmetric	PLS-complete[18]

one user tries to reroute at each specific time. In that model, rerouting succeeds only if the user migrates to an alternative resource with lower load. The work shows a simple randomized algorithm in which the expected number of rerouting attempts, until convergence to a Nash equilibrium, is polynomial in the number of the number of link users.

Even-Dar and Mansour [17] allow concurrent, independent reallocation decisions where tasks are allowed to migrate from resources with load above the average to resources with load below the average. They show that the system reaches a Nash equilibrium after expected  $O(\log \log m + \log n)$  rounds, where  $m$  is the number of tasks and  $n$  is the number of resources. However, their protocol requires tasks to have a certain amount of global knowledge in order to make their decisions.

Berenbrink et al.[4] consider the distributed load balancing model and restrict themselves to uniform tasks. They show an upper bound of  $O(\log \log m + n^4)$  on the convergence time in their model as well as a lower bound of  $\Omega(\log \log m + n)$ . Without loss of generality, assume that  $n$  is divisible by  $m$ , the only Nash equilibrium in the model studied in [4] is the *perfectly balanced states*, which means the load of each resource is exactly  $m/n$ . It can be observed that their convergence result also holds for a model that has one Nash equilibrium which is the perfectly balanced states. In particular it can be shown that, if there are identical resources with an arbitrary increasing latency function,  $(\log \log m + n^4)$  is the upper bound for the protocol. They furthermore derive bounds on the convergence time to an approximate Nash equilibrium as well as an exponential lower bound for a slight modification of their protocol, which is possibly even more natural than the one considered for the  $O(\log \log m + n^4)$  bound in that it results in a perfectly even distribution in expectation after

only one step whereas the “quicker” protocol does not have this property. Since their model is one of the models we consider in this thesis, and we actually borrow some of their techniques in our results, we explain this model in more details later.

Chien and Sinclair [11] study the convergence to approximate Nash equilibria for symmetric congestion games in which the edges delay satisfy a *bounded jump* condition. By using the properties of the Rosenthal’s potential function, they show that convergence to an  $\epsilon$ -Nash equilibrium occurs within a number of steps that is polynomial in the number of players and  $\epsilon^{-1}$ .

In [20], Fischer, Räcke, and Vöcking investigate the convergence to Wardrop equilibria for both asymmetric and symmetric routing games. Wardrop’s model [41] is one of the most important and well studied continuous network routing models. In this model an infinite population of agents carries an infinitesimal amount of flow each. Several commodities with specified flow demands are indicated in the network and the flow must satisfy the demands. Each agent may choose from a set of paths and strives to minimize its sustained latency selfishly. Similar to the congestion game model, the latency of a path  $p$  only depends on the volume of the agents who have chosen  $p$ . Population states which are stable in that no agent can improve its latency by migrating to another path are referred to as Wardrop equilibria. [19]

Fischer et al. [20] present a replicative rerouting protocol in which agents adopt strategies of more successful agents. For the symmetric case, where there is only one commodity in the network, and agents choose their routes from a common set of strategies, they show that their protocol converge fast toward a Wardrop equilibrium.

### 1.3 Natural variations and results

We consider the problem of dynamically reallocating  $m$  tasks among a set of  $n$  resources. We assume an arbitrary initial placement of tasks, and we study the performance of distributed, natural reallocation algorithms. We are interested in the time it takes the system to converge to a Nash equilibrium (or get close to an equilibrium).

Our result improves the previous upper bound for the uniform case where tasks share identical resources, and the latency function of a resource is the number of tasks utilizing it. In a first step, in Theorem 2.3.1 we show that for uniform tasks our protocol reaches (the unique) Nash equilibrium in expected time  $O(\log m + n \log n)$ . This already is better than [4] for small values of  $m$  (roughly when  $m/\log m < 2^{\Theta(n^4)}$ ), but still worse for  $m \gg n$ . The reason for this is mainly that we have smaller migration probabilities than [4] (the slowdown factor  $\rho$  in our protocol). These smaller probabilities have the effect that they speed up the end game (that is, once we are close to an equilibrium) at the expense of the early game, where the protocol in [4] is quicker. We first show a lower bound for the expected convergence time of our protocol of  $\Omega(\log m + n)$ , and then mention (Remark 2.3.8) how to *combine* our protocol with that of [4] in order to obtain overall  $O(\log \log m + n \log n)$ . The idea here is mainly to run the protocol in [4] during the early game, and then later switch to our new protocol (which is faster for almost balanced systems), thus getting the best from both approaches.

We also consider weighted tasks. Theorem 3.1.1 shows that our protocol yields an expected time to converge to an  $\epsilon$ -Nash equilibrium of  $O(nm\Delta^3\epsilon^{-2})$ . Notice that this would appear to be much worse than the  $O(\log \log m + \text{poly}(n))$  bound in [4] when only considering uniform tasks (i.e., assuming  $\Delta = 1$ ). We do, however, provide a lower bound of  $\Omega(\epsilon^{-1}m\Delta)$  in Observation 3.3.1 for the case  $\Delta \geq 2$ . In Corollary 3.2.7 we also show convergence to a Nash equilibrium in expected time  $O(mn\Delta^5)$  in the case of integer weights. To justify our logarithmic bound on  $m$  instead of something smaller like  $\log \log m$  in [4] we refer the reader to Section 3.3. The ideas used in our proofs are different from those in [4], the main reason being that equilibria are no longer unique and can, in fact, have very different potentials. It is therefore not possible (as it was in [4]) to “simply” analyze in terms of distance from equilibrium potential, namely zero, as there is no such thing. Instead, we show that tasks improve their induced cost when and if they can as long as the possible improvement is not too small.

The main contribution of this thesis is as follows. We study the variation where tasks are identical, but resources have arbitrary latency functions. Similar to the concept of *virtual potential* in

[21, 20], we define the virtual potential function  $\Psi$ . Then we show that under our reallocation protocol, the Rosenthal's potential function drops at least half of the absolute value of  $\Psi$ . We then state our main theorem which shows the system converges to Nash equilibrium in expected time  $O\left(\frac{4\beta \cdot n^2}{\alpha \cdot R^2} \cdot \log \frac{\Phi(x_0)}{\Phi^*}\right)$ , where  $\Phi(x_0)$  and  $\Phi^*$  are the initial and minimum possible potential, respectively.  $R, \alpha, \beta$  are some parameters related to the cost (latency) functions. Moreover we give some examples to justify that  $\beta$  is the relevant parameter in our analysis.

## 1.4 The structure of the thesis

The rest of the thesis is structured as follows. In the next chapter, we revisit the convergence results of Berenbrink et al. [4] for the uniform case. We express our new protocol and the analysis of our results which beats the previous  $O(\log \log m + n^4)$  upper bound in [4]. In Chapter 3, we define the setting for weighted tasks and show the upper bound for the convergence time of our protocol to an  $\epsilon$ -Nash equilibrium. We also show a lower bound to justify the protocol for the weighted case. In Chapter 4, we study the case where resources have different latency functions. There we give the upper bound of  $O\left(\frac{4\beta \cdot n^2}{\alpha \cdot R^2} \cdot \log \frac{\Phi(x_0)}{\Phi^*}\right)$ . We conclude in Chapter 5 with a summary of our results and future directions.

## Unit Weighted Tasks and Uniform Resources

In this chapter, we consider the *uniform case* where  $m$  identical unit weighted tasks are to be assigned to a set of  $n$  uniform resources. The cost of a resource is simply the number of users using it. We assume an arbitrary initial placement of tasks, and we study the performance of distributed, natural reallocation algorithms. We are interested in the time it takes the system to converge to an equilibrium.

The assignment of tasks to resources is represented by a vector  $x(t) = (x_1(t), \dots, x_n(t))$  where  $x_i(t)$  denotes the number of tasks using resource  $i$  in time  $t$ .

### Potential function

For any assignment  $x = (x_1, \dots, x_n)$ , the potential function  $\Phi(x)$  is defined as following,

$$\Phi(x) = \sum_{i=1}^n (x_i - \bar{x})^2$$

where  $\bar{x} = \frac{1}{n} \sum_{i=1}^n x_i$ .

This potential function has the property that if only one task migrates to a new resource, the change in  $\Phi$  is exactly twice the gain of the task. Note that,  $\Phi$  is equivalent to the Rosenthal's potential function[34] in the sense that after a single migration, the change in  $\Phi$  is proportional to the improvement of the cost

## 2.1 The $O(\log \log m + n^4)$ convergence time upper bound

As we mentioned before, Berenbrink et. al. [4] studied the uniform case, and showed that the greedy distributed protocol in Algorithm 1, guarantees a fast convergence toward the Nash equilibrium.

---

### Algorithm 1 Reallocation Protocol for Uniform Case [4]

---

- 1: **for** each task  $b$  in parallel **do**
  - 2:   let  $i$  be the current resource of task  $b$
  - 3:   choose resource  $j$  uniformly at random
  - 4:   **if**  $X_i(t) > X_j(t) + 1$  **then**
  - 5:     move task  $b$  from resource  $i$  to  $j$  with probability  $1 - \frac{X_j(t)}{X_i(t)}$
- 

In the above algorithm, for every resource  $i$  and time step  $t$ ,  $X_i(t)$  is a random variable defined as the load of resource  $i$  at time  $t$ . We can consider  $X(t) = (X_1(t), \dots, X_n(t))$  as the state of the system at time  $t$ , and the transition from state  $X(t)$  to  $X(t+1)$  is given by the protocol in Algorithm 1.

Note that if  $X(t)$  is a Nash equilibrium, none of the tasks have incentive to migrate to another resource, and thus  $X(t+1) = X(t)$ .

The authors of [4] use the concept of multinomial distribution in order to describe the transition from a state  $X(t) = x$ . Independently, for every  $i \in [n]$ , let  $(Y_{i,1}(x), \dots, Y_{i,n}(x))$  be a random variable drawn from a multinomial distribution with the constraint  $\sum_{j=1}^n Y_{i,j}(x) = x_i$ . (Given the state  $x$ ,  $Y_{i,j}(x)$  represents the number of migrations from  $i$  to  $j$  in one round.) The corresponding probabilities  $p_{ij}(x)$  are given by



$$p_{ij}(x) = \begin{cases} \frac{1}{n} \left(1 - \frac{x_j}{x_i}\right) & \text{if } x_i > x_j + 1, \\ 0 & \text{if } i \neq j \text{ and } x_i \leq x_j + 1, \\ 1 - \sum_{r \in [n]: r \neq i} p_{ir}(x) & \text{if } i = j. \end{cases}$$

Given the state  $X(t) = x$ ,  $X_i(t+1) = \sum_{l=1}^n Y_{l,i}(x)$ .

**Theorem 2.1.1.** [4] *The expected number of steps taken by protocol of Algorithm 1 to reach a Nash equilibrium for the first time is  $O(\log \log m + n^4)$ .*

**Sketch of the proof [4].** The proof of this theorem proceeds as follows. First they give an upper bound on  $E[\Phi(X(t))]$  which implies that there is a  $\tau = O(\log \log m)$  such that, with high probability,  $\Phi(X(\tau)) = O(n)$ . They also show that  $\Phi(X(t))$  is a super-martingale and it has enough variance. Using these facts, they obtain the upper bound on the convergence time.

## 2.2 Reallocation protocol with a slowdown factor

We modify the protocol of the previous section, and introduce another reallocation protocol for the uniform case in Algorithm 2. We set  $\rho = 1/8$  as a slowdown factor, and let the migration probability (Line 5) be  $\rho$  times the original migration probability.

---

### Algorithm 2 Reallocation Protocol with a Slowdown Factor

---

- 1: **for** each task  $b$  in parallel **do**
  - 2:   **let**  $i$  **be** the current resource of task  $b$
  - 3:   choose resource  $j$  uniformly at random
  - 4:   **if**  $X_i(t) > X_j(t) + 1$  **then**
  - 5:     move task  $b$  from resource  $i$  to  $j$  with probability  $\rho \left(1 - \frac{X_j(t)}{X_i(t)}\right)$
- 

Note that when Algorithm 2 terminates, we have  $\forall i, j \in [n]$ ,  $x_i < x_j + 2$ . Hence the system is in (the) Nash equilibrium.

Although experimental results show that the new protocol is actually slower than the one in [4], by slowing down the protocol, we would be able to show that given the state  $X(t) = x$ , the expected

potential function at the next step,  $E[\Phi(X(t+1))]$ , drops by a constant multiplicative factor. More formally in Lemma 2.3.7, we show that for any  $t > 0$ ,  $E[\Phi(X(t+1))] \leq (1 - \frac{1}{8n}) E[\Phi(X(t))]$ .

The result of Lemma 2.3.7 is quite strong, because it holds for any time step  $t > 0$ . In particular when we are close to a Nash equilibrium, without applying the slowdown factor, it would be difficult to get such a result. The authors of [4] use a martingale technique to get the upper bound for this case. We think that it would be an interesting open question to give a potential function argument, similar to the one in Lemma 2.3.7 for the original potential.

## 2.3 Convergence to Nash Equilibrium

Our main result in this section is as follows.

**Theorem 2.3.1.** *Given any initial load configuration  $X(0) = x$ . Let  $T$  be the number of rounds taken by the protocol in Algorithm 2 to reach the unique Nash equilibrium for the first time. Then,*

$$E[T] = O(\log m + n \log n).$$

*Furthermore,  $T = O(\log m + n \log n)$  with a probability of at least  $1 - 1/n$ .*

In the following, we show after  $O(\log m + n \log n)$  steps, Algorithm 2 terminates with high probability. This improves the previous upper bound of  $O(\log \log m + n^4)$  in [4] for small values of  $m$ . In fact, we can actually combine these two protocols to obtain a tight convergence time of  $O(\log \log m + n \log n)$  with high probability<sup>1</sup>. The tightness of this result can be shown by Theorem 4.2 in [4] and Observation 2.4.2.

For simplicity we assume that  $m$  is a multiple of  $n$ , the proof can easily be extended to  $n \nmid m$ . We first bound the expected potential drop in one round. Then we show that in each round the potential drops at least by a factor of  $1/32$  if the current system potential is larger than  $n$  (Lemma

---

<sup>1</sup>The probability is at least  $1 - 1/n^\alpha$  for some constant  $\alpha > 0$

2.3.7(1)), and at least by a factor of  $1/8n$  otherwise (Lemma 2.3.7(2)). With these two lemmas, we are ready to show Theorem 2.3.1.

We will use the same potential function  $\Phi(x)$  as the one in the previous section.

**Observation 2.3.2.** *In the following we present two useful tools,*

1.

$$\Phi(x) = \sum_{i=1}^n (x_i - \bar{x})^2 = \frac{1}{n} \cdot \sum_{i=1}^n \sum_{k=i+1}^n (x_i - x_k)^2.$$

2.

$$\begin{aligned} E[\Phi(X(t+1))|X(t) = x] &= \sum_{i=1}^n \left( E[X_i(t+1)|X(t) = x] - \bar{x} \right)^2 \\ &+ \sum_{i=1}^n \text{var}[X_i(t+1)|X(t) = x] \end{aligned}$$

*Proof.* Part (1) is similar to Lemma 10 in [7]. Part (2) is in Appendix A.1.  $\square$

For resource  $i, k \in [n]$ , let  $E[W_{i,k}]$  denote the expected number of tasks being transferred from resource  $i$  to  $k$ . Note that by Algorithm 2, if  $x_i - x_k \geq 2$ ,  $E[W_{i,k}] = x_i \cdot \rho(1 - x_k/x_i)/n = \rho(x_i - x_k)/n$ , otherwise  $E[W_{i,k}] = 0$ . Let  $S_i(x) = \{k : x_i \geq x_k + 2\}$  and  $E_i(x) = \{k : x_i = x_k + 1\}$ .

Let

$$\Gamma(x) = \frac{1}{n} \sum_{i=1}^n \sum_{k \in S_i(x)} (x_i - x_k)^2.$$

Note that the bigger  $\Gamma(x)$  is, the more tasks are expected to be transferred by Algorithm 2. We first show some relations between  $\Gamma(x)$  and  $\Phi(x)$ .

**Observation 2.3.3.** *For any load configuration  $x$ , we have*

1. *If  $\Phi(x) \geq n$ , then  $\Gamma(x) > \Phi(x)/2$ .*

2. *If  $\Gamma(x) < 2$ , then  $\Gamma(x) = \Phi(x)^2/n$  and  $\Phi(x) < \sqrt{2n}$ .*

3. If  $\Phi(x) < 2$ , then  $x$  is Nash equilibrium.

*Proof.* For Part (1), by definition,

$$\Gamma(x) = \Phi(x) - \frac{1}{n} \sum_{i=1}^n \sum_{k \in E_i(x)} 1 \geq \Phi(x) - \frac{1}{n} \cdot \frac{n(n-1)}{2} = \Phi(x) - \frac{n-1}{2}.$$

Hence if  $\Phi(x) \geq n$ , we get  $\Gamma(x) \geq \Phi(x) - (n-1)/2 > \Phi(x)/2$ .

For Part (2), we first show that if  $\Gamma(x) < 2$ , then  $x_1 - x_n \leq 2$  (notice that  $x_1 \geq x_2 \geq \dots \geq x_n$ ).

For a contradiction assume that  $\Gamma(x) < 2$  and  $x_1 - x_n \geq 3$ . Hence  $\forall 1 \leq i \leq n$ , either  $|x_i - x_1| \geq 2$  or  $|x_i - x_n| \geq 2$ . Also notice that by symmetry we have

$$\Gamma(x) = \frac{1}{n} \sum_{i=1}^n \sum_{k \in S_i(x)} (x_i - x_k)^2 = \frac{1}{2n} \sum_{i=1}^n \sum_{k \in S_i(x)} (x_i - x_k)^2 \geq \frac{1}{2n} \cdot n \cdot (2^2) = 2,$$

yielding a contradiction.

Next we show  $\Gamma(x) = \Phi(x)^2/n$ . Since  $\bar{x} = m/n$  is an integer and  $x_1 - x_n \leq 2$ , each resource can only have  $\bar{x} - 1, \bar{x}, \bar{x} + 1$  tasks. Let  $A(B)$  be the set of resources with  $\bar{x} - 1$  tasks and  $(\bar{x} + 1$  tasks), respectively. Of course  $|A| = |B| = r$ . Thus  $\Phi(x) = \sum_{i=1}^n (x_i - \bar{x})^2 = 2r$ . Hence,

$$\Gamma(x) = \frac{1}{n} \sum_{i=1}^n \sum_{k \in S_i(x)} (x_i - x_k)^2 = \frac{1}{n} \cdot (2r)^2 = 4r^2/n = \Phi(x)^2/n.$$

Consequently, given  $\Gamma(x) < 2$ , we have  $\Phi(x) \leq \sqrt{2n}$ .

For Part (3), for a contradiction assume that  $x$  is not Nash equilibrium. Then there must be two resources  $u, v$ , such that  $x_u \geq \bar{x} + 1$  and  $x_v \leq \bar{x} - 1$ . Thus  $\Phi(x) = \sum_{i=1}^n (x_i - \bar{x})^2 \geq 2$ . We get a contradiction.

□

**Lemma 2.3.4.**

$$E[\Phi(X(t+1))|X(t) = x] < \Phi(x) - \frac{1}{2} \sum_{i=1}^n \sum_{k=i+1}^n E[W_{i,k}](x_i - x_k).$$

*Proof.* Combining Observation 2.3.2(2), Lemma 3.2.1(1) and 3.2.2, we get

$$\begin{aligned} & E[\Phi(X(t+1))|X(t) = x] \\ &= \sum_{i=1}^n \left( E[X_i(t+1)|X(t) = x] - \bar{x} \right)^2 + \sum_{i=1}^n \text{var}[X_i(t+1)|X(t) = x] \\ &< \Phi(x) - (2 - 4\rho) \sum_{i=1}^n \sum_{k=i+1}^n E[W_{i,k}](x_i - x_k) + (2 - \epsilon) \sum_{i=1}^n \sum_{k=i+1}^n E[W_{i,k}](x_i - x_k) \\ &= \Phi(x) - \frac{\epsilon}{2} \sum_{i=1}^n \sum_{k=i+1}^n E[W_{i,k}](x_i - x_k), \end{aligned}$$

□

We then show the following bound for the expected potential drop in one step.

**Lemma 2.3.5.**  $E[\Phi(X(t+1))|X(t) = x] \leq \Phi(x) - \Gamma(x)/16$ .

*Proof.* Recall that if  $x_i - x_k \geq 2$ ,  $E[W_{i,k}] = \rho(x_i - x_k)/n$ , otherwise  $E[W_{i,k}] = 0$ . Hence,

$$\Gamma(x) = \frac{1}{n} \sum_{i=1}^n \sum_{k \in S_i(x)} (x_i - x_k)^2 = \rho^{-1} \cdot \sum_{i=1}^n \sum_{k=i+1}^n E[W_{i,k}](x_i - x_k).$$

Due to Lemma 2.3.4 we get

$$E[\Phi(X(t+1))|X(t) = x] = \Phi(x) - \frac{1}{2} \sum_{i=1}^n \sum_{k=i+1}^n E[W_{i,k}](x_i - x_k) = \Phi(x) - \frac{\rho\Gamma(x)}{2} = \Phi(x) - \frac{\Gamma(x)}{16}$$

□

The following corollaries follow from Lemma 2.3.5.

**Corollary 2.3.6.**

1. If  $\Phi(x) \geq n$ ,  $E[\Phi(X(t+1))|X(t) = x] < (1 - 1/32)\Phi(x)$ .
2. If  $n > \Phi(x) \geq \sqrt{2n}$ ,  $E[\Phi(X(t+1))|X(t) = x] \leq \Phi(x) - 1/8$ .
3. If  $\sqrt{2n} > \Phi(x)$ ,  $E[\Phi(X(t+1))|X(t) = x] \leq \Phi(x) - \Phi(x)^2/(16n)$ .

*Proof.* Part (1) follows directly from Lemma 2.3.5 and Observation 2.3.3(1).

To prove Part (2), if  $\Phi(x) \geq \sqrt{2n}$ , by Observation 2.3.3(2)  $\Gamma(x) \geq 2$ . Then using Lemma 2.3.5 we get  $E[\Phi(X(t+1))|X(t) = x] \leq \Phi(x) - 1/8$ .

For Part (3), note that  $E[\Phi(X(t+1))|X(t) = x] \leq \Phi(x) - \Gamma(x)/16$  by Lemma 2.3.5. Thus it is sufficient to show that  $\Gamma(x) \geq \Phi(x)^2/n$ . We consider two cases for different values of  $\Gamma(x)$ . If  $\Gamma(x) \geq 2$ ,  $\Gamma(x) > \Phi(x)^2/n$  since  $\Phi(x) < \sqrt{2n}$ . If  $\Gamma(x) < 2$ , by Observation 2.3.3(2),  $\Gamma(x) = \Phi(x)^2/n$ .  $\square$

Next we prove two results that bound the expected potential drop.

**Lemma 2.3.7.** *For any  $t > 0$ ,*

1.  $E[\Phi(X(t+1))] \leq \max\{n, (1 - 1/32)E[\Phi(X(t))]\}$ .
2.  $E[\Phi(X(t+1))] \leq (1 - \frac{1}{8n})E[\Phi(X(t))]$ .

*Proof.* For Part (1), by Corollary 2.3.6(1),

$$\begin{aligned} E[\Phi(X(t+1))] &= \sum_x E[\Phi(X(t+1))|X(t) = x] \cdot \Pr[X(t) = x] \\ &\leq \sum_x \max\{n, (1 - 1/32)\Phi(x)\} \cdot \Pr[X(t) = x] \\ &= \max\{n, (1 - 1/32)E[\Phi(X(t))]\}. \end{aligned}$$

To prove Part (2), we first show that for any load configuration  $x$ ,  $E[\Phi(X(t+1))|X(t) = x] \leq (1 - 1/(8n))\Phi(x)$ . There are four cases for different values of  $\Phi(x)$ .

1. If  $\Phi(x) \geq n$ , by Corollary 2.3.6(1),  $E[\Phi(X(t+1))|X(t) = x] < (1 - 1/32)\Phi(x) < (1 - 1/(8n))\Phi(x)$  as long as  $n > 4$ .

2. If  $n > \Phi(x) \geq \sqrt{2n}$ , by Corollary 2.3.6(2),  $E[\Phi(X(t+1))|X(t) = x] \leq \Phi(x) - 1/8 \leq (1 - 1/(8n))\Phi(x)$  since  $\Phi(x)/(8n) < 1/8$  due to  $\Phi(x) < n$ .
3. If  $\sqrt{2n} > \Phi(x) \geq 2$ , by Corollary 2.3.6(3),  $E[\Phi(X(t+1))|X(t) = x] \leq \frac{\Phi(x) - \Phi(x)^2}{16n} \leq (1 - 1/(8n))\Phi(x)$  since  $\Phi(x) \geq 2$ .
4. Finally, if  $\Phi(x) < 2$ , by Observation 2.3.3(3),  $x$  must be Nash equilibrium so that  $\Phi(x) = 0$ . In this case the system potential will not change. Hence  $E[\Phi(X(t+1))|X(t) = x] = 0 \leq (1 - 1/(8n))\Phi(x)$ .

Consequently,

$$\begin{aligned} E[\Phi(X(t+1))] &= \sum_x E[\Phi(X(t+1))|X(t) = x] \cdot \Pr[X(t) = x] \\ &\leq \sum_x \left(1 - \frac{1}{8n}\right) \Phi(x) \cdot \Pr[X(t) = x] = \left(1 - \frac{1}{8n}\right) E[\Phi(x)]. \end{aligned}$$

□

We are now ready to prove the main result in this section.

*Proof of Theorem 2.3.1.* We first show that after  $\tau = 64 \ln m$  steps,  $E[\Phi(X(\tau))] \leq n$ . By Observation 3.1.2(3),  $\Phi(X(0)) \leq m^2 \Delta^2 = m^2$ . Using Lemma 2.3.7(1) iteratively for  $\tau$  times, we get

$$E[\Phi(X(\tau))] \leq \max\{n, (1 - 1/32)^\tau \cdot \Phi(X(0))\} \leq \max\{n, (1 - 1/32)^\tau \cdot m^2\} = n.$$

We then show that after  $T = 16n \ln n$  additional steps, the system reaches Nash equilibrium w.h.p. Using Lemma 2.3.7(2) iteratively for  $T$  times, we get

$$E[\Phi(X(\tau+T))] \leq E[\Phi(X(\tau))] \cdot (1 - 1/(8n))^T \leq n \cdot (1 - 1/(8n))^{16n \ln n} < n \cdot e^{-2 \ln n} = n^{-1}.$$

By Markov's inequality,  $\Pr[\Phi(X(\tau+T)) \geq 2] < 1/n$ . Observation 2.3.3(3) tells us that if  $\Phi(X(\tau+T)) \geq 2$ , then  $\Phi(X(\tau+T)) \geq 2$  and  $\Phi(X(\tau+T)) \leq 2$  with probability at least  $1/n$ .

$T)) < 2$ ,  $X(\tau + T)$  is Nash equilibrium. Hence, after  $\tau + T = 64 \ln m + 16n \ln n$  steps, the probability that the system does not reach the Nash equilibrium is at most  $1/n$ .  $\square$

**Remark 2.3.8.** *Note that we can combine Algorithm 1 and Algorithm 2 to obtain an algorithm that converges in  $O(\log \log m + n \log n)$  steps. To see this, first note that by Corollary 3.9 in [4], after  $T_1 = 2 \log \log m$  steps,  $E[\Phi(X(T_1))] \leq 18n$ . Then using a similar argument as above, we can show that after  $O(\log \log m + n \log n)$ , the system state is at some Nash equilibrium w.h.p.*

## 2.4 Lower bounds

We prove the following two lower bound results which show the tightness of Theorem 2.3.1. As discussed earlier, the “slowdown” ( $\log m$  as opposed to the  $\log \log m$  in [4]) is the result of the introduction of the factor  $\rho$  to the migration probabilities in our protocol.

**Observation 2.4.1.** *Let  $T$  be the first time at which  $E[X(t)] \leq c$  for constant  $c > 0$ . There is an initial load configuration  $X(0)$  that requires  $T = \Omega(\log m)$ .*

*Proof.* Consider a system with  $n = 2$  resources and  $m$  uniform tasks. Let  $X(0) = (m, 0)$ . We first show that  $E[\Phi(X(t+1))] \geq \frac{7}{8}E[\Phi(X(t))]$ . By definition,

$$\sum_{i=1}^n \sum_{k=i+1}^n E[W_{i,k}](x_i - x_k) = \frac{\rho}{n} \left( \sum_{i=1}^n \sum_{k=i+1}^n I_{i,k}(1 - x_k/x_i)(x_i - x_k) \right) \leq \frac{\Phi(x)}{8}.$$

Hence, setting  $\epsilon = 1$  in Lemma 2.3.4(2) we obtain

$$E[\Phi(X(t+1))|X(t) = x] \geq \Phi(x) - \epsilon \sum_{i=1}^n \sum_{k=i+1}^n E[W_{i,k}](x_i - x_k) \geq \frac{7\Phi(x)}{8}.$$

Now similar to Lemma 2.3.7 (1) we can show that  $E[\Phi(X(t+1))] \geq 7E[\Phi(X(t))]/8$ . Note that  $\Phi(X(0)) = m^2/2$ . In order to make  $E[X(T)] \leq c$ , we need  $T = \Omega(\log m)$ .  $\square$



**Observation 2.4.2.** *Let  $T$  be the first time at which  $X(t)$  is a Nash equilibrium and  $T^*$  be the upper bound for  $T$ . There is an initial load configuration  $X(0)$  that in order to make  $\Pr[T \leq T^*] > 1 - 1/n$ , we need  $T^* = \Omega(n \log n)$ .*

*Proof.* Consider a system with  $n$  resources and  $m = n$  uniform tasks. Let  $X(0)$  be the assignment given by  $X(0) = (2, 1, \dots, 1, 0)$ . Denote  $q$  be the probability for the tasks in resource 1 to move to resource  $n$  (if exactly one of the two tasks in resource 1 moves, the system reaches the Nash equilibrium). By Algorithm 2 (with  $\rho = 1/8$ ),  $q = 2/(2\rho n) = 1/(8n)$ . Note that  $T$  is geometrically distributed with probability  $2q(1 - q) < 1/(4n)$ . Consequently,  $\Pr[T > T^*] \leq (1/(4n))^{T^*}$  (since steps  $1, \dots, T^*$  all fail). Thus, to have  $\Pr[T \leq T^*] > 1 - 1/n$ , we need  $T^* = \Omega(n \log n)$ .  $\square$

**Remark 2.4.3.** *Note that this lower bound also holds for the protocol in [4] (with  $\rho = 1$ ).*

## Weighted Tasks and Uniform Resources

In this model we have  $m$  weighted tasks  $b_1, \dots, b_m$  and  $n$  uniform resources. Assume that  $m \geq n$ .

Each task  $b_i \in [m]$  is associated with a weight  $w_i \geq 1$ . Let  $\Delta = \max\{w_i\}$  denote the maximum weight of any task. Let  $M = \sum_{i=1}^m w_i$  be the total task weight.

The assignment of tasks to resources is represented by a vector  $(x_1(t), \dots, x_n(t))$  in which  $x_i(t)$  denotes the *load* of resource  $i$  at the end of step  $t$ , i.e., the sum of weights of tasks allocated to resource  $i$ . Let  $\bar{x} = M/n$  be the average load. For any task  $b \in [m]$ , let  $r_b$  denote the current resource of task  $b$ .

**Definition.** [*Nash equilibrium*] An assignment is a *Nash equilibrium* for task  $b$  if

$$x_{r_b} \leq x_j + w_b \text{ for all } j \in [n], \quad (3.1)$$

i.e., if task  $b$  cannot improve its situation by migrating to any other resource.

**Definition.** [ $\epsilon$ -*Nash equilibrium*] For  $0 \leq \epsilon \leq 1$ , we say a state is an  $\epsilon$ -*Nash equilibrium* for task  $b$  if

$$x_{r_b} \leq x_j + (1 + \epsilon)w_b. \quad (3.2)$$

Notice that this definition is somewhat different from (and stronger than), say, Chien and Sinclair's

in [11] where they say that (translated into our model) a state is an  $\epsilon'$ -Nash equilibrium for  $\epsilon' \in (0, 1)$  if  $(1 - \epsilon')x_{r_b} \leq x_j + w_b$  for all  $j \in [n]$ . However, our definition captures theirs: for  $\epsilon' \in (0, 1)$  let  $\epsilon = \frac{1}{1-\epsilon'} - 1 (> 0)$  and observe that  $x_{r_b} \leq x_j + (1 + \epsilon)w_b \leq (1 + \epsilon)(x_j + w_b) = (1 + (\frac{1}{1-\epsilon'} - 1))(x_j + w_b) = \frac{x_j + w_b}{1-\epsilon'}$ .

### 3.1 The reallocation model with weighted tasks

We define our allocation process for weighted tasks and uniform resources. Let  $X_1(0), \dots, X_n(0)$  be the initial assignment. The transition from state  $X(t) = (X_1(t), \dots, X_n(t))$  to state  $X(t+1)$  is given by the protocol below. Let  $0 \leq \epsilon \leq 1$  and  $\rho = \epsilon/8$ .

---

#### Algorithm 3 Greedy Reallocation Protocol for Weighted Tasks

---

- 1: **for** each task  $b$  in parallel **do**
  - 2:   let  $r_b$  be the current resource of task  $b$
  - 3:   choose resource  $j$  uniformly at random
  - 4:   **if**  $X_{r_b}(t) \geq X_j(t) + (1 + \epsilon)w_b$  //violation of Eq.3.2// **then**
  - 5:     move task  $b$  from resource  $r_b$  to  $j$  with probability  $\rho \left(1 - \frac{X_j(t)}{X_{r_b}(t)}\right)$
- 

If the process converges, i.e. if  $X(t) = X(t+1)$  for all  $t \geq \tau$  for some  $\tau \in \mathbb{N}$ , then the system has reached some  $\epsilon$ -Nash equilibrium (“some” because  $\epsilon$ -Nash equilibria are, in general, not unique). Our goal is to bound the number of steps it takes for the algorithm to converge, that is, to find the smallest  $\tau$  with the property from above. We prove the following convergence result.

**Theorem 3.1.1.** *Let  $\epsilon > 0$  and  $\rho = \epsilon/8$ . Let  $\Delta \geq 1$  denote the maximum weight of any task. Let  $T$  be the number of rounds taken by the protocol in Algorithm 3 to reach an  $\epsilon$ -Nash equilibrium for the first time. Then,*

$$E[T] = O(mn\Delta^3(\rho\epsilon)^{-1}).$$

### Potential Function

For the analysis we use a standard potential function:

$$\Phi(x) = \sum_{i=1}^n (x_i - \bar{x})^2 = \sum_{i=1}^n x_i^2 - n\bar{x} \quad (3.3)$$

(see also [4]). In the following we assume, without loss of generality, that the assignment is “normalized”, meaning  $x_1 \geq \dots \geq x_n$ . If it is clear from the context we will omit the time parameter  $t$  in  $X(t) = (X_1(t), \dots, X_n(t))$  and write  $X = (X_1, \dots, X_n)$  instead. We say task  $b$  has an *incentive* to move to resource  $i$  if  $x_{r_b} \geq x_i + (1 + \epsilon)w_b$  (notice that this is the condition used in Line 4 of Algorithm 3).

Let  $Y^b = (Y^b(r_b, 1), \dots, Y^b(r_b, n))$  be a random variable with  $\sum_{i=1}^n Y^b(r_b, i) = 1$ .  $Y^b$  is an  $n$ -dimensional unit vector with precisely one coordinate equal to 1 and all others equal to 0.  $Y^b(r_b, i) = 1$  corresponds to the event of task  $b$  moving from resource  $r_b$  to resource  $i$  (or staying at resource  $i$  if  $i = r_b$ ). Let the corresponding probabilities  $(P^b(r_b, 1), \dots, P^b(r_b, n))$  be given by

$$P^b(r_b, i) = \begin{cases} \frac{\rho(1-x_i/x_{r_b})}{n} & \text{if } r_b \neq i \text{ and } x_{r_b} > x_i + (1 + \epsilon)w_b \\ 0 & \text{if } r_b \neq i \text{ and } x_{r_b} \leq x_i + (1 + \epsilon)w_b \\ 1 - \sum_{k \in [n]: x_i > x_k} P^b(i, k) & \text{if } r_b = i. \end{cases}$$

The first (second) case corresponds to randomly choosing resource  $i$  and finding (not finding) an incentive to migrate, and the third case corresponds to randomly choosing the current resource.

For  $i \in [n]$ , let  $S_i(t)$  denote the set of tasks currently on resource  $i$  at step  $t$ . In the following we will omit  $t$  in  $S_i$  and write  $S_i$  if it is clear from the content. For  $i, j \in [n]$  with  $i \neq j$ , let  $I_{j,i}$  be the total weight of tasks on resource  $j$  that have an incentive to move to resource  $i$ , i.e.,

$$I_{j,i} = \sum_{b \in S_j: x_j \geq x_i + (1 + \epsilon)w_b} w_b.$$

Let

$$E[W_{j,i}] = \sum_{b \in S_j} w_b \cdot P^b(j, i) \leq x_j \cdot \frac{\rho(1 - x_i/x_j)}{n} = \frac{\rho(x_j - x_i)}{n}$$

denote the expected total weight of tasks migrating from resource  $j$  to resource  $i$  (the last inequality is true because  $I_{j,i} \leq x_j$ ; at most all the tasks currently on  $j$  migrate to  $i$ ). Next, we show three simple observations.

**Observation 3.1.2.**

1.  $\Phi(x) = \frac{1}{n} \cdot \sum_{i=1}^n \sum_{k=i+1}^n (x_i - x_k)^2$ .
2.  $\Phi(x) \leq m^2 \Delta^2$ .

*Proof.* Part (1) is similar to Lemma 10 in [7]. For Part (2), simply consider the worst case in which all the  $m$  tasks are in one particular resource.  $\square$

## 3.2 Convergence to Nash equilibrium

In this section we bound the number of time steps for the system to reach some Nash equilibrium. We first bound the expected potential change during a fixed time step  $t$  (Lemma 3.2.3). For this we shall first prove two technical lemmas: bounds for  $\sum_{i=1}^n (E[X_i(t+1)|X(t) = x] - \bar{x})^2$  and  $\sum_{i=1}^n \text{var}[X_i(t+1)|X(t) = x]$ , respectively (Lemma 3.2.1 and Lemma 3.2.2).

**Lemma 3.2.1.**

1.  $\sum_{i=1}^n (E[X_i(t+1)|X(t) = x] - \bar{x})^2 < \Phi(x) - (2 - 4\rho) \sum_{i=1}^n \sum_{k=i+1}^n E[W_{i,k}](x_i - x_k)$ .
2.  $\sum_{i=1}^n (E[X_i(t+1)|X(t) = x] - \bar{x})^2 > \Phi(x) - 2 \sum_{i=1}^n \sum_{k=i+1}^n E[W_{i,k}](x_i - x_k)$ .

*Proof.* Since  $E[W_{i,j}]$  is the expected total weight migrating from resource  $i$  to  $j$ , we have  $E[X_i(t+1)|X(t) = x] = x_i + \sum_{j=1}^{i-1} E[W_{j,i}] - \sum_{k=i+1}^n E[W_{i,k}]$ ; recall that we assume  $x_1 \geq \dots \geq x_n$ .

To estimate  $\sum_{i=1}^n (E[X_i(t+1)|X(t) = x] - \bar{x})^2$ , we use an indirect approach by first analyzing a (deterministic) load balancing process. We then use the load balancing process to show our desired result (see [7]).

We consider the following load balancing scenario: Assume that there are  $n$  resources and every pair of resources is connected so that we have a complete network. Initially, every resource  $1 \leq i \leq n$  has  $z_i = x_i$  load items on it. Assume that  $z_1 \geq \dots \geq z_n$ . Then every pair of resources  $(i, k), i < k$  concurrently exchanges  $\ell_{i,k} = E[W_{i,k}] \leq \rho(x_i - x_k)/n = \rho(z_i - z_k)/n$  load items. If  $i \geq k$  we assume  $\ell_{i,k} = 0$ . Note that the above system is similar to one step of the diffusion load balancing algorithm on a complete graph  $K_n$ . In both cases the exact potential change is hard to calculate due to the concurrent load transfers. The idea we use now is to first “sequentialize” the load transfers, measure the potential difference after each of these sub-steps, and then to use these results to get a bound on the total potential drop for the whole step.

In the following we assume that every edge  $e_s = (i, k), i, k \in [n], k > i$  is labelled with weight  $\ell_{i,k} \geq 0$ . Note that  $\ell_{i,k} = 0$  if  $x_i \leq x_k$ . Let  $N = n(n-1)/2$  and  $E = \{e_1, e_2, \dots, e_N\}$  be the set of edges sorted in increasing order of their labels. We assume the edges are sequentially activated, starting with the edge  $e_1$  with the smallest weight. Let  $z^s = (z_1^s, \dots, z_n^s)$  be the load vector resulting after the activation of the first  $s$  edges. Note that  $z^0 = (z_1^0, \dots, z_n^0)$  is the load vector before load balancing and  $z^N = (z_1^N, \dots, z_n^N)$  is the load vector the activation of all edges. Note that  $\Phi(z^0) = \Phi(x)$  since  $i \in [n], z_i^0 = z_i = x_i$ . Moreover, by the definition of our load balancing process and since  $\ell_{i,k} = E[W_{i,k}]$  we have

$$z_i^N = z_i + \sum_{j=1}^{i-1} \ell_{j,i} - \sum_{k=i+1}^n \ell_{i,k} = x_i + \sum_{j=1}^{i-1} E[W_{i,j}] - \sum_{k=i+1}^n E[W_{i,k}] = E[X_i(t+1)|X(t) = x]$$

Hence

$$\Phi(z^N) = \sum_{i=1}^n (z_i^N - \bar{x})^2 = \sum_{i=1}^n \left( E[X_i(t+1)|X(t) = x] - \bar{x} \right)^2.$$

Next we bound  $\Phi(z^N)$ . For any  $s \in [N]$ , let  $\Delta_s(\Phi) = \Phi(z^{s-1}) - \Phi(z^s)$  be the potential drop

due to the activation of edge  $e_s = (i, k)$ . Note that

$$\Phi(z^0) - \Phi(z^N) = \sum_{s=1}^N (\Phi(z^{s-1}) - \Phi(z^s)) = \sum_{e_s \in E} \Delta_s(\Phi).$$

Now we bound  $\Delta_s(\Phi)$ . Since all edges are activated in increasing order of their weights we get  $l_{i,j} \leq l_{i,k} = \rho(z_i - z_k)/n$  for any node  $j$  that is considered before the activation of  $e_s$ . Node  $i$  has  $n - 2$  additional neighbors, hence the expected load that it can send to these neighbors before the activation of edge  $e_s = (i, k)$  is at most  $(n - 2)l_{i,k} < \rho(z_i - z_k) - l_{i,k}$ . This gives us

$$z_i^{s-1} \geq z_i - (n - 2)l_{i,k} > z_i - \rho(z_i - z_k) + l_{i,k}.$$

Similarly, the expected load that  $k$  receives before the activation of edge  $e_s = (i, k)$  is at most  $\rho(z_i - z_k) - l_{i,k}$ . Hence,

$$z_k^{s-1} < z_k + \rho(z_i - z_k) - l_{i,k}.$$

Thus,

$$\begin{aligned} \Delta_s(\Phi) &= (z_i^{s-1})^2 + (z_k^{s-1})^2 - (z_i^{s-1} - l_{i,k})^2 - (z_k^{s-1} + l_{i,k})^2 \\ &= 2l_{i,k} \cdot (z_i^{s-1} - z_k^{s-1} - l_{i,k}) > (2 - 4\rho)l_{i,k}(z_i - z_k). \end{aligned}$$

Similarly, since  $z_i^{s-1} < z_i$  and  $z_k^{s-1} > z_k$ , we get  $\Delta_s(\Phi) = 2l_{i,k} \cdot (z_i^{s-1} - z_k^{s-1} - l_{i,k}) < 2l_{i,k}(z_i - z_k)$ .

Next we bound  $\Phi(z^N)$ .

$$\begin{aligned} \Phi(z^0) - \sum_{i=1}^n \sum_{k=i+1}^n 2l_{i,k}(z_i - z_k) &< \Phi(z^N) \\ &= \Phi(z^0) - \sum_{e_s \in E} \Delta_s(\Phi) < \Phi(z^0) - \sum_{i=1}^n \sum_{k=i+1}^n (2 - 4\rho)l_{i,k}(z_i - z_k). \end{aligned}$$

Consequently, we get the following two bounds,

$$\Phi(z^N) = \sum_{i=1}^n \left( E[X_i(t+1)|X_i = x] - \bar{x} \right)^2 < \Phi(x) - (2 - 4\rho) \sum_{i=1}^n \sum_{k=i+1}^n E[W_{i,k}](x_i - x_k),$$

$$\Phi(z^N) = \sum_{i=1}^n \left( E[X_i(t+1)|X_i = x] - \bar{x} \right)^2 > \Phi(x) - 2 \sum_{i=1}^n \sum_{k=i+1}^n E[W_{i,k}](x_i - x_k).$$

□

Next we show an upper bound for the sum of variance.

**Lemma 3.2.2.**  $\sum_{i=1}^n \text{var}[X_i(t+1)|X(t) = x] < (2 - \epsilon) \sum_{i=1}^n \sum_{k=i+1}^n E[W_{i,k}](x_i - x_k).$

*Proof.* First of all, note that  $\{Y^b(r_b, i)\}$  and  $\{Y^{b'}(r_{b'}, i)\}$  are independent for  $b \neq b'$ . Let  $S_i(t)$  be the set of tasks that is assigned to resource  $i$  in step  $t$ .

$$\begin{aligned} & \text{var}[X_i(t+1)|X(t) = x] \\ &= \text{var} \left[ \sum_b w_b \cdot Y^b(r_b, i) \right] = \sum_b w_b^2 \cdot \text{var}[Y^b(r_b, i)] \\ &= \sum_{j=1}^n \sum_{b \in S_j(t)} w_b^2 \cdot \text{var}[Y^b(r_b, i)] \\ &= \sum_{j \neq i} \sum_{b \in S_j(t)} w_b^2 \cdot P^b(r_b, i)(1 - P^b(r_b, i)) + \sum_{b \in S_i(t)} w_b^2 \cdot P^b(r_b, i)(1 - P^b(r_b, i)) \\ &< \sum_{j \neq i} \sum_{b \in S_j(t)} w_b^2 \cdot P^b(r_b, i) + \sum_{b \in S_i(t)} w_b^2 \cdot (1 - P^b(r_b, i)) \\ &= \sum_{j \neq i} \sum_{b \in S_j(t)} w_b^2 \cdot P^b(r_b, i) + \sum_{b \in S_i(t)} w_b^2 \cdot \sum_{j \neq i} P^b(r_b, j) \\ &= \sum_{j \neq i} \sum_{b \in S_j(t)} w_b^2 \cdot P^b(r_b, i) + \sum_{j \neq i} \sum_{b \in S_i(t)} w_b^2 \cdot P^b(r_b, j) \\ &\leq \sum_{j \neq i} \sum_{b \in S_j(t)} w_b \cdot \frac{x_j - x_i}{1 + \epsilon} \cdot P^b(r_b, i) + \sum_{j \neq i} \sum_{b \in S_i(t)} w_b \cdot \frac{x_i - x_j}{1 + \epsilon} \cdot P^b(r_b, j) \\ &= \sum_{j \neq i} E[W_{j,i}] \cdot \frac{x_j - x_i}{1 + \epsilon} + \sum_{j \neq i} E[W_{i,j}] \cdot \frac{x_i - x_j}{1 + \epsilon} \end{aligned}$$



The second inequality holds since  $(x_j - x_i) \geq (1 + \epsilon) \cdot w_b$  whenever a task  $b$  in resource  $j$  have an incentive to move to resource  $i$  (see Algorithm 3). Now note that  $E[W_{i,j}] = 0$  whenever  $x_j > x_i$ .

Hence,

$$\begin{aligned} \sum_{i=1}^n \text{var}[X_i(t+1)|X(t) = x] &= \sum_{i=1}^n \sum_{j:x_j > x_i} E[W_{j,i}] \cdot \frac{(x_j - x_i)}{1 + \epsilon} + \sum_{i=1}^n \sum_{j:x_i > x_j} E[W_{i,j}] \cdot \frac{(x_i - x_j)}{1 + \epsilon} \\ &= \sum_{i=1}^n \sum_{j=i+1}^n 2E[W_{i,i}] \cdot \frac{(x_i - x_j)}{1 + \epsilon} \\ &< (2 - \epsilon) \sum_{i=1}^n \sum_{k=i+1}^n E[W_{i,k}](x_i - x_k). \end{aligned}$$

□

Now we are ready to show the following lemma bounding the potential change during step  $t$ .

**Lemma 3.2.3.**

1.  $E[\Phi(X(t+1))|X(t) = x] < \Phi(x) - \frac{\epsilon}{2} \sum_{i=1}^n \sum_{k=i+1}^n E[W_{i,k}](x_i - x_k)$ .
2.  $E[\Phi(X(t+1))|X(t) = x] > \Phi(x) - \epsilon \sum_{i=1}^n \sum_{k=i+1}^n E[W_{i,k}](x_i - x_k)$ .

*Proof.* To prove part (1), combining Observation 2.3.2(2), Lemma 3.2.1(1) and 3.2.2, we get

$$\begin{aligned} &E[\Phi(X(t+1))|X(t) = x] \\ &= \sum_{i=1}^n \left( E[X_i(t+1)|X(t) = x] - \bar{x} \right)^2 + \sum_{i=1}^n \text{var}[X_i(t+1)|X(t) = x] \\ &< \Phi(x) - (2 - 4\rho) \sum_{i=1}^n \sum_{k=i+1}^n E[W_{i,k}](x_i - x_k) + (2 - \epsilon) \sum_{i=1}^n \sum_{k=i+1}^n E[W_{i,k}](x_i - x_k) \\ &= \Phi(x) - \frac{\epsilon}{2} \sum_{i=1}^n \sum_{k=i+1}^n E[W_{i,k}](x_i - x_k), \end{aligned}$$

since  $\rho = \epsilon/8$ . The proof of part (2) is similar. □

Next we first show that if  $\Phi(x) \geq 4n\Delta^2$ , then the expected system potential decreases by a

multiplicative factor of at least  $\rho\epsilon/4$  per round (Lemma 3.2.4). We then show that whenever  $x$  is not  $\epsilon$ -Nash equilibrium, in every round the system potential decreases at least by an additive factor of  $\rho\epsilon/(6m\Delta)$  in expectation (Lemma 3.2.6). With these two lemmas, we are ready to show our main result (Theorem 3.1.1).

**Lemma 3.2.4.** *If  $\Phi(x) \geq 4n\Delta^2$ ,  $\Delta$  is the maximum task weight. We have  $E[\Phi(X(t+1))|X(t) = x] < (1 - \rho\epsilon/4)\Phi(x)$ .*

*Proof.* We first bound  $\sum_{i=1}^n \sum_{k=i+1}^n E[W_{i,k}](x_i - x_k)$ . Recall that  $E[W_{i,k}] = I_{i,k} \cdot \rho \cdot \frac{1-(x_k/x_i)}{n}$ , where  $0 \leq I_{i,k} \leq x_i$  is the total weight of tasks in  $x_i$  which have an incentive to migrate to  $x_k$ . To prove our bound we only add up the cases when  $I_{i,k} = x_i$ . Note that if  $I_{i,k} < x_i$ , we have  $x_i - x_k < (1 + \epsilon)\Delta$ , since otherwise every task in resource  $i$  would have an incentive to move to resource  $k$  resulting in  $I_{i,k} = x_i$ .

$$\begin{aligned}
\sum_{i=1}^n \sum_{k=i+1}^n E[W_{i,k}](x_i - x_k) &= \sum_{i=1}^n \sum_{k=i+1}^n I_{i,k} \cdot \frac{\rho(1-x_k/x_i)}{n} \cdot (x_i - x_k) \\
&\geq \frac{\rho}{n} \cdot \sum_{i=1}^n \sum_{k:I_{i,k}=x_i} I_{i,k}(1-x_k/x_i)(x_i - x_k) \\
&= \frac{\rho}{n} \cdot \sum_{i=1}^n \sum_{k:I_{i,k}=x_i} (x_i - x_k)^2 \\
&= \frac{\rho}{n} \cdot \left( \sum_{i=1}^n \sum_{k=i+1}^n (x_i - x_k)^2 - \sum_{i=1}^n \sum_{k:I_{i,k}<x_i} (x_i - x_k)^2 \right) \\
&\geq \frac{\rho}{n} \cdot \left( n\Phi(x) - \binom{n}{2} \cdot (1+\epsilon)^2 \Delta^2 \right) \geq \frac{\rho\Phi(x)}{2},
\end{aligned}$$

since  $\Phi(x) \geq 4n\Delta^2$  and  $\epsilon \leq 1$ . Now, using Lemma 3.2.3(1) we obtain,

$$E[\Phi(X(t+1))|X(t) = x] \leq \Phi(x) - \frac{\epsilon}{2} \sum_{i=1}^n \sum_{k=i+1}^n E[W_{i,k}](x_i - x_k) \leq (1 - \rho\epsilon/4) \cdot \Phi(x).$$

□

It is easy to derive the following corollary from Lemma 3.2.4.

**Corollary 3.2.5.** *For any  $t > 0$ ,  $E[\Phi(X(t+1))] \leq \max\{4n\Delta^2, (1 - \rho\epsilon/4) \cdot E[\Phi(X(t))]\}$ .*

*Proof.* We first show that  $E[\Phi(X(t+1))|X(t) = x] < \max\{4n\Delta^2, (1 - \rho\epsilon/4)\Phi(x)\}$ . We consider two cases for different values of  $\Phi(x)$ . If  $\Phi(x) \leq 4n\Delta^2$ , by Lemma 3.2.3(1)  $E[\Phi(X(t+1))|X(t) = x] < \Phi(x) < 4n\Delta^2$ . If  $\Phi(x) > 4n\Delta^2$ , by Lemma 3.2.4  $E[\Phi(X(t+1))|X(t) = x] < (1 - \rho\epsilon/4)\Phi(x)$ . Consequently,

$$\begin{aligned} E[\Phi(X(t+1))] &= \sum_x \{E[\Phi(X(t+1))|X(t) = x] \cdot \Pr[X(t) = x]\} \\ &\leq \sum_x \{\max\{4n\Delta^2, (1 - \rho\epsilon/4)\Phi(x)\} \cdot \Pr[X(t) = x]\} \\ &= \max\{4n\Delta^2, (1 - \rho\epsilon/4) \cdot E[\Phi(X(t))]\}. \end{aligned}$$

□

Next we show that whenever the system is not at some  $\epsilon$ -Nash equilibrium, the system potential decreases by an amount of  $\rho\epsilon/(6m\Delta)$  in expectation.

**Lemma 3.2.6.** *Assume that at step  $t$  the system is not at some  $\epsilon$ -Nash equilibrium. We have  $E[\Phi(X(t+1))|X(t) = x] \leq \Phi(x) - \frac{\rho\epsilon}{6m\Delta}$ .*

*Proof.* We consider two cases for different values of  $x_1$ , the maximum load of a resource.

1.  $x_1 > \bar{x} + 2\Delta$ . In this case we have  $x_1 > x_n + 2\Delta > x_n + (1 + \epsilon)\Delta$  since  $x_n \leq \bar{x}$  and  $0 < \epsilon < 1$ . Thus, every task in resource 1 has an incentive to move to resource  $n$ . Using

Lemma 3.2.3(1), we get

$$\begin{aligned}
E[\Phi(X(t+1))|X(t) = x] &\leq \Phi(x) - \frac{\epsilon}{2} \cdot \sum_{i=1}^n \sum_{k=i+1}^n E[W_{i,k}](x_i - x_k) \\
&\leq \Phi(x) - \frac{\epsilon}{2} \cdot E[W_{1,n}](x_1 - x_n) \\
&= \Phi(x) - \frac{\epsilon}{2} \cdot \frac{x_1 \cdot \rho(1 - x_n/x_1)}{n} \cdot (x_1 - x_n) \\
&= \Phi(x) - \frac{\rho\epsilon(x_1 - x_n)^2}{2n} \\
&< \Phi(x) - \frac{2\rho\epsilon\Delta^2}{n} < \Phi(x) - \frac{\rho\epsilon}{6m\Delta}.
\end{aligned}$$

2.  $x_1 \leq \bar{x} + 2\Delta$ . Since  $x$  is not  $\epsilon$ -Nash equilibrium, there must be at least one task  $b$  that has an incentive to migrate to some resource  $v \neq r_b$ . Note that  $x_{r_b} - x_v \geq (1 + \epsilon)w_b > 1$  and  $x_{r_b} \leq x_1 \leq \bar{x} + 2\Delta$ . Similar to Case 1,

$$\begin{aligned}
E[\Phi(X(t+1))|X(t) = x] &\leq \Phi(x) - \frac{\epsilon}{2} \cdot E[W_{r_b,v}](x_{r_b} - x_v) \\
&\leq \Phi(x) - \frac{\epsilon}{2} \cdot w_b \cdot \frac{\rho(1 - x_v/x_{r_b})}{n} \cdot (x_{r_b} - x_v) \\
&= \Phi(x) - \frac{\rho \cdot \epsilon \cdot w_b \cdot (x_{r_b} - x_v)^2}{2x_{r_b} \cdot n} \\
&\leq \Phi(x) - \frac{\rho\epsilon}{2(\bar{x} + 2\Delta)n} \leq \Phi(x) - \frac{\rho\epsilon}{6m\Delta}.
\end{aligned}$$

For the last inequality, we use  $\bar{x} \cdot n = M \leq m \cdot \Delta$  and  $m \geq n$ .

□

Now we are ready to prove Theorem 3.1.1.

*Proof of Theorem 3.1.1.* We first show that after  $\tau = 8(\epsilon\rho)^{-1} \log m$  steps,  $E[\Phi(X(\tau))] \leq 4n\Delta^2$ .

By Observation 2.3.2(2),  $\Phi(X(0)) \leq m^2\Delta^2$ .

Repeatedly using Corollary 3.2.5, we get  $E[\Phi(X(\tau))] \leq \max\{4n\Delta^2, (1 - \rho\epsilon/4)^\tau \cdot \Phi(X(0))\} = 4n\Delta^2$ . By Markov inequality,  $\Pr[\Phi(X(\tau)) > 40n\Delta^2] \leq 0.1$ .

The following proof is done by a standard Martingale argument similar to [4]. Let us assume that  $\Phi(X(\tau)) \leq 40n\Delta^2$ . Let  $T$  be the number of additional time steps for the system to reach some  $\epsilon$ -Nash equilibrium after step  $\tau$  and let  $t \wedge T$  be the minimum of  $t$  and  $T$ . Let  $V = \rho\epsilon/(6m\Delta)$  and let  $Z_t = \Phi(X(t + \tau)) + Vt$ . Observe that  $\{Z_t\}_{t \wedge T}$  is a supermartingale since by Lemma 3.2.6 with  $X(t + \tau) = x$ ,

$$E[Z_{t+1}|Z_t = z] = E[\Phi(X(\tau + t + 1)) | \Phi(x) = z - Vt] + V \cdot (t+1) \leq (z - Vt - V) + V \cdot (t+1) = z.$$

Hence  $E[Z_{t+1}] = \sum_z E[Z_{t+1}|Z_t = z] \cdot \Pr[Z_t = z] \leq \sum_z z \cdot \Pr[Z_t = z] = E[Z_t]$ . We obtain

$$V \cdot E[T] \leq E[\Phi(X(\tau + T))] + V \cdot E[T] = E[Z_T] \leq \dots \leq E[Z_0] \leq 40n\Delta^2.$$

Therefore  $E[T] \leq 40n\Delta^2/V = 240mn\Delta^3(\rho\epsilon)^{-1}$ , and  $\Pr[T > 2400mn\Delta^3(\rho\epsilon)^{-1}] < 0.1$  by Markov's inequality. Hence, after  $\tau + T = 8(\rho\epsilon)^{-1} \log m + 2400mn\Delta^3(\rho\epsilon)^{-1}$  rounds, the probability that the system is not at some  $\epsilon$ -Nash equilibrium is at most  $0.1 + 0.1 = 0.2$ .

Subdivide time into intervals of  $\tau + T$  steps each. The probability that the process has *not* reached an  $\epsilon$ -Nash equilibrium after  $s$  intervals is at most  $(1/5)^s$ . This finishes the proof.  $\square$

The following corollary bounds the convergence time to a (real, non- $\epsilon$ ) Nash equilibrium in the case of integral weights.

**Corollary 3.2.7.** *Assume that every task has integer weight of at least 1, and let  $\epsilon = 1/\Delta$ . Let  $T$  be the number of rounds taken by the protocol in Algorithm 3 to reach a Nash equilibrium for the first time. Then,*

$$E[T] = O(mn\Delta^5).$$

*Proof.* When Algorithm 3 terminates, for any task  $b$  and resource  $i \in [n]$ , we have  $x_{r_b} < x_i + (1 + \epsilon)w_b < x_i + w_b + 1 \leq x_i + w_b$  since  $w_b \leq \Delta$  and  $w_b$  is an integer. This implies that the system is at one of the Nash equilibria. Now, setting  $\epsilon = 1/\Delta$  in Theorem 3.1.1 and using  $\rho = \epsilon/8 = (8\Delta)^{-1}$ ,

we obtain the result.  $\square$

### 3.3 Lower bounds

The main reason for the slow convergence is that migration probabilities (must) depend on the quotients of the involved resource loads. Intuitively, problematic cases are those where we have two resources with (large) loads that differ only slightly. With uniform tasks we would have that *all* tasks on the higher-loaded of the two would have a (small) probability to migrate. Here, bigger tasks may be perfectly happy and there may be only very few (small) tasks on the higher-loaded resource that would attempt the migration, each also with only small probability (recall that uniform tasks implies a direct correspondence between load and number of tasks).

The authors feel that it is an interesting open problem to design a protocol that requires no or only a very small amount of (global) knowledge with regards to weight distribution, average loads, and number of tasks on each resource which circumvents this problem.

**Observation 3.3.1.** *Let  $T$  be the first time at which  $X(t)$  is the Nash equilibrium. There is a load configuration  $X(0)$  that requires  $E[T] = \Omega(m\Delta/\epsilon)$ .*

*Proof.* Consider a system with  $n$  resources,  $n$  tasks of weight 1 each, and  $m - n$  tasks of weight  $\Delta \geq 2$  each. Let  $\ell = m/n$  where  $m$  is a multiple of  $n$ . Let  $X(0) = ((\ell - 1)\Delta + 2, (\ell - 1)\Delta + 1, \dots, (\ell - 1)\Delta + 1, (\ell - 1)\Delta)$ . The perfectly balanced state is the only Nash equilibrium. Let  $q$  be the probability for the unit-size tasks in resource 1 to move to resource  $n$  (if exactly one of the two unit-sized tasks moves, the system reaches the Nash equilibrium). By Algorithm 3, we have  $q = \rho \cdot 2/(n((\ell - 1)\Delta + 2)) = O(\epsilon/m\Delta)$  since  $\ell = m/n$  and  $\rho = \epsilon/8$ . Note that  $T$  is geometric distributed with probability  $2q(1 - q)$ . Thus  $E[T] = 1/(2q(1 - q)) = \Omega(m\Delta/\epsilon)$ .  $\square$

**Remark 3.3.2.** *We believe that since there is lack of global knowledge and also tasks query the load of only one other server, even with significant change to the protocol we can not omit the term  $\Delta$ . For an evidence consider two different games as follow, both with two servers.*

- *There are 4 tasks with weights  $(1, 1, \Delta, \Delta)$  and the initial configuration is  $(\Delta + 2, \Delta)$ .*
- *There are  $2\Delta + 2$  tasks all of unit weight, and the initial configuration is  $(\Delta + 2, \Delta)$ .*

*Considering the lack of global knowledge, a task with unit weight can not distinguish between the above games. But in order to have a fast convergence to Nash Equilibrium(see [17] for the definition), in the first game it needs to migrate with a probability significantly higher than the corresponding probability in the second game.*

## Resources with Different Latency

### Functions

In this chapter we study the variation where tasks are identical, but resources have arbitrary latency functions. In this model we have  $m$  uniform tasks  $b_1, \dots, b_m$  and  $n$  resources with different latency functions  $f_i$ . The latency function of resource  $i$  depends only on the number of tasks using resource  $i$  and increases with the number of tasks using the resource. The assignment of tasks to resources is represented by a vector  $x(t) = (x_1(t), \dots, x_n(t))$  where  $x_i(t)$  denotes the number of tasks using resource  $i$  in time  $t$ . In this model  $f_i(x_i(t))$  would be the latency function of resource  $i$  and each task using the resource experiences this delay. Let  $\beta_i = \max_{a \in (0, m)} f'_i(a)$  which is nothing else than the maximum slope of  $f_i$  on the interval  $(0, m)$ . We define  $\beta = \max_i \beta_i$ . For each task  $b$ , let  $r_b$  denote the utilized resource of task  $b$ .

**Nash equilibrium** An assignment is a Nash equilibrium if for every task  $b$ ,

$$f(x_{r_b}) \leq f(x_j + 1) \text{ for all } j \in [n], \tag{4.1}$$

Intuitively the above definition means no task has an incentive to migrate to another resource.



## 4.1 The reallocation process with arbitrary latency functions

We define our reallocation process for  $m$  uniform tasks and  $n$  resources with different latency functions. Suppose  $X_1(0), \dots, X_n(0)$  is the initial assignment and the transition from state  $X(t) = X_1(t), \dots, X_n(t)$  to state  $X(t+1)$  is given by Algorithm 4. Suppose the vector  $x = (x_1, \dots, x_n)$

---

**Algorithm 4** Reallocation protocol for tasks utilizing resources with different latency functions

---

- 1: **for** each task  $b$  in parallel **do**
  - 2:   let  $i$  be the current resource of task  $b$
  - 3:   choose resource  $j$  uniformly at random
  - 4:   **if**  $f_i(X_i(t)) > f_j(X_j(t)+1)$  **then**
  - 5:     move task  $b$  from resource  $i$  to  $j$  with probability  $\frac{f_i(X_i(t)) - f_j(X_j(t)+1)}{X_i(t) \cdot (\beta_i + \beta_j)}$
- 

is the load vector of the resources at (fixed) time  $t$ . According to the reallocation protocol in Algorithm 4, after one step, each task either switches to a new resource or remains at its utilized resource. Therefore at time  $t+1$ , we expect a new random load vector  $X(t+1)$ . We are interested in bounding the number of steps it takes for the protocol to converge, which means the system ends up in a configuration which is a Nash equilibrium.

### 4.1.1 Potential function

We define the natural potential function  $\Phi(x)$  as following:

$$\Phi(x) = \sum_{i=1}^n \sum_{k=1}^{x_i} f_i(k)$$

Note that, this potential function is introduced by Rosenthal[34] and recently used by Chien and Sinclair [11]. This function has the property that if at time  $t$ , only one task migrates to a new resource, the change in  $\Phi$  is exactly the gain of the task [34].

### 4.1.2 Summary of the convergence result

In order to state our convergence result we do need to define two more notations:

**Definition:**

$$R = \min_{i,j,l; f_i(l_i) > f_j(l_j+1)} \frac{f_i(l_i) - f_j(l_j+1)}{f_i(l_i)}$$

$$\alpha = \min_{i \in (0,n], l; f_i(l_i) > 0} \frac{f_i(l_i)}{l_i}$$

where  $l = (l_1, \dots, l_n)$  is an assignment.

**Theorem 4.1.1.** *Given any initial load vector  $X(0)$ . Let  $T$  be the number of steps taken by the protocol in Algorithm 4 to reach a Nash equilibrium for the first time. Then,*

$$E[T] = O\left(\frac{1}{C} \log \frac{\Phi(x_0)}{\Phi^*}\right) = O\left(\frac{4\beta \cdot n^2}{\alpha \cdot R^2} \cdot \log \frac{\Phi(X(0))}{\Phi^*}\right)$$

Where  $\Phi^*$  refers to the optimal(as minimum as possible) potential, and  $C = -\log\left(1 - \frac{\alpha \cdot R^2}{4\beta \cdot n^2}\right) \approx \frac{\alpha \cdot R^2}{4\beta \cdot n^2}$ .

## 4.2 Notation and preliminary results

In order to prove the convergence result we mentioned in the end of the previous section, we need to define some more notation and get some preliminary results. Note that after the reallocation process in a round, the load of some resources increase and the load of some other decrease. Let  $\Delta X_i(t) = X_i(t+1) - X_i(t)$ . We define  $P^+(t)$  as the set of resources whose load increase after the reallocation process is done in round  $t$ . (e.g.  $P^+(t) = \{i | \Delta X_i(t) > 0\}$ ). Similarly  $P^-(t)$  is the set of resources whose load decrease. (e.g.  $P^-(t) = \{i | \Delta X_i(t) < 0\}$ ).

**Definition:** Given a state  $X(t) = x$ ,  $u_i^+(x)$  is the number of tasks migrating to resource  $i$  from other resources in a round, similarly  $u_i^-(x)$  is the number of tasks migrating from resource  $i$  to other resources.

Similar to [4], we describe the transition from a state  $X(t) = x$ . Given the state  $x$ , let the number of migrations from resource  $i$  to resource  $j$  in a round be represented by random variable  $Y_{ij}(x)$ .

Independently, similar to the observation in [4], for every  $i \in [n]$ ,  $Y_{i1}(x), \dots, Y_{in}(x)$  are drawn from a multinomial distribution with the constraint  $\sum_{j=1}^n Y_{ij}(x) = x_i$ . The corresponding probabilities  $\rho_{ij}(x)$  are given by

$$\rho_{ij}(x) = \begin{cases} \frac{1}{n} \left( \frac{f_i(x_i) - f_j(x_j + 1)}{x_i \cdot (\beta_i + \beta_j)} \right) & \text{if } i \neq j \text{ and } f_i(x_i) > f_j(x_j + 1) \\ 0 & \text{if } i \neq j \text{ and } f_i(x_i) \leq f_j(x_j + 1) \\ 1 - \sum_{r \in [n]: r \neq i} \rho_{ir}(x) & \text{if } i = j. \end{cases}$$

We define the random variable  $\Psi$  which is called *virtual potential* as follows

**Definition:**

$$\Psi(x) = \sum_{i,j \in [n]} Y_{ij}(x) \cdot [f_j(x_j + 1) - f_i(x_i)]$$

We use the new notation  $\Psi$  for the sake of analysis. Fischer et al. also use a similar definition in their own settings[21, 20]. We have the following observation and lemma:

**Observation 4.2.1.**

$$\Psi(x) = \sum_i f_i(x_i + 1) \cdot u_i^+(x) - f_i(x_i) \cdot u_i^-(x)$$

*Proof.*

$$\begin{aligned} \Psi(x) &= \sum_{i,j \in [n]} Y_{ij}(x) \cdot [f_j(x_j + 1) - f_i(x_i)] \\ &= \sum_i f_i(x_i + 1) \cdot \sum_j Y_{ji}(x) - \sum_i f_i(x_i) \cdot \sum_j Y_{ij}(x) \\ &= \sum_i f_i(x_i + 1) \cdot u_i^+(x) - f_i(x_i) \cdot u_i^-(x) \end{aligned}$$

□

Let  $\xi_i(t, x) = \frac{\beta_i}{2} \mathbb{E}[\Delta X_i(t)^2 - |\Delta X_i(t)| | X(t) = x]$ . In the following lemma we show a lower bound for the *absolute* value of the expected potential drop is large.

**Lemma 4.2.2.**

$$\mathbb{E}[\Phi(X(t+1)|X(t) = x)] - \Phi(x) \leq \Psi(x) + \sum_i \xi_i(t, x)$$

*Proof.* Replacing  $\Phi(x)$  with its definition we get,

$$\begin{aligned} & \mathbb{E}[\Phi(X(t+1)|X(t) = x)] - \Phi(x) \\ &= \mathbb{E} \left[ \sum_{i \in P^+(t)} \sum_{j=1}^{\Delta X_i(t)} f_i(x_i + j) - \sum_{i \in P^-(t)} \sum_{j=1}^{|\Delta X_i(t)|} f_i(x_i - |\Delta X_i(t)| + j) | X(t) = x \right] \end{aligned}$$

Thus,

$$\begin{aligned} & \mathbb{E} \left[ \Phi(X(t+1)) - \Phi(x) - \Psi(x) | X(t) = x \right] \\ &= \mathbb{E} \left[ \sum_{i \in P^+(t)} \sum_{j=1}^{\Delta X_i(t)} [f_i(x_i + j) - f_i(x_i + 1)] + \sum_{i \in P^-(t)} \sum_{j=1}^{|\Delta X_i(t)|} [f_i(x_i) - f_i(x_i + \Delta X_i(t) + j)] \right. \\ & \quad \left. - \sum_{i \in P^+(t)} u_i^-(x) [f_i(x_i + 1) - f_i(x_i)] - \sum_{i \in P^-(t)} u_i^+(x) [f_i(x_i + 1) - f_i(x_i)] | X(t) = x \right] \\ &\leq \mathbb{E} \left[ \sum_{i \in P^+(t)} \beta_i \sum_{j=1}^{\Delta X_i(t)} (j-1) + \sum_{i \in P^-(t)} \beta_i \sum_{j=1}^{-\Delta X_i(t)} (|\Delta X_i(t)| - j) | X(t) = x \right] \\ &\leq \mathbb{E} \left[ \sum_{i \in P^+(t)} \beta_i \frac{\Delta X_i(t)(\Delta X_i(t) - 1)}{2} + \sum_{i \in P^-(t)} \beta_i \frac{-\Delta X_i(t)(-\Delta X_i(t) - 1)}{2} | X(t) = x \right] \\ &\leq \mathbb{E} \left[ \sum_i \frac{\beta_i}{2} |\Delta X_i(t)| \cdot (|\Delta X_i(t)| - 1) | X(t) = x \right] \\ &\leq \sum_i \frac{\beta_i}{2} (\mathbb{E}[\Delta X_i(t)^2 | X(t) = x] - \mathbb{E}[|\Delta X_i(t)| | X(t) = x]) \\ &= \sum_i \xi_i(t, x) \end{aligned}$$

□

**Observation 4.2.3.** For every  $1 \leq i \leq n$ ,

$$\xi_i(t, x) \leq E[u_i^+(x)^2] + E[u_i^-(x)^2] - E[u_i^+(x)] - E[u_i^-(x)]$$

*Proof.* In order to prove this observation, we prove the following inequality

$$[\Delta X_i(t)|X(t) = x]^2 - |\Delta X_i(t)|X(t) = x| \leq [u_i^+(x)]^2 + [u_i^-(x)]^2 - u_i^+(x) - u_i^-(x)$$

We consider two different cases, and for simplicity we omit the conditioning on  $X(t) = x$  where it is clear:

- Case 1,  $u_i^+(x) = 0$ :

$$[\Delta X_i(t)]^2 - |\Delta X_i(t)| = [u_i^+(x) - u_i^-(x)]^2 - |u_i^+(x) - u_i^-(x)| = [u_i^-(x)]^2 - u_i^-(x).$$

- Case 2,  $u_i^+(x) \geq 1$ :

$$\begin{aligned} [\Delta X_i(t)]^2 - |\Delta X_i(t)| &= [u_i^+(x) - u_i^-(x)]^2 - |u_i^+(x) - u_i^-(x)| \\ &= [u_i^+(x)]^2 + [u_i^-(x)]^2 - 2 \cdot u_i^+(x) \cdot u_i^-(x) - |u_i^+(x) - u_i^-(x)| \\ &\leq [u_i^+(x)]^2 + [u_i^-(x)]^2 - 2 \cdot u_i^+(x) \cdot u_i^-(x) - u_i^+(x) + u_i^-(x) \\ &\leq [u_i^+(x)]^2 + [u_i^-(x)]^2 - u_i^+(x) - u_i^-(x). \end{aligned}$$

□

**Observation 4.2.4.**

$$1. E[u_i^+(x)]^2 = \left( \sum_{j \neq i} E[Y_{ji}(x)] \right)^2 \leq (n-1) \cdot \sum_{j \neq i} E[Y_{ji}(x)]^2$$

$$2. E[u_i^-(x)]^2 = \left( \sum_{j \neq i} E[Y_{ij}(x)] \right)^2 \leq (n-1) \cdot \sum_{j \neq i} E[Y_{ij}(x)]^2$$

*Proof.* Use Cauchy-Schwarz inequality. ■

**Observation 4.2.5.**  $\text{VAR}[\Delta X_i(t)|X(t) = x] = \sum_{j=1}^n \text{VAR}[Y_{ji}(x)]$

*Proof.* Given the state  $X(t) = x$ , for each  $1 \leq i \leq n$ , the number of task migrating to the resource  $i$ ,  $u_i^+(x)$ , and the number of tasks migrating to other resource from  $i$ ,  $u_i^-(x)$  are independent of each other.  $[\Delta X_i(t)|X(t) = x] = u_i^+(x) - u_i^-(x)$ , thus  $\text{VAR}[\Delta X_i(t)|X(t) = x] = \text{VAR}[u_i^+(x)] + \text{VAR}[u_i^-(x)]$ . Moreover for every  $r_a \neq r_b$ , and  $i$ ,  $Y_{r_a,i}(x)$  and  $Y_{r_b,i}(x)$  are independent. We use these facts to complete the proof:

$$\begin{aligned} \text{VAR}[u_i^+(x)] + \text{VAR}[u_i^-(x)] &= \text{VAR}\left[\sum_{j \neq i} Y_{ji}(x)\right] + \text{VAR}\left[\sum_{j \neq i} Y_{ij}(x)\right] \\ &= \sum_{j \neq i} \text{VAR}[Y_{ji}(x)] + \text{VAR}[x_i - Y_{ii}(x)] \\ &= \sum_{j \neq i} \text{VAR}[Y_{ji}(x)] + \text{VAR}[Y_{ii}(x)] \\ &= \sum_j \text{VAR}[Y_{ji}(x)]. \end{aligned}$$

□

### 4.3 Convergence to Nash equilibria

In this section we present our convergence to Nash equilibria result. First we prove the following simple observation.

**Observation 4.3.1.**

$$\sum_i \frac{\beta_i}{2} \text{VAR}[\Delta X_i(t)|X(t) = x] \leq \sum_i \sum_{j \neq i} \left( \frac{\beta_i + \beta_j}{2} \right) \text{VAR}[Y_{ij}(x)]$$

*Proof.* Due to observation 4.2.5,  $\sum_i \frac{\beta_i}{2} \text{VAR}[\Delta X_i(t) | X(t) = x] = \sum_i \frac{\beta_i}{2} \sum_j \text{VAR}[Y_{ji}(x)]$ .

$$\begin{aligned}
\sum_i \frac{\beta_i}{2} \sum_j \text{VAR}[Y_{ji}(x)] &= \sum_i \frac{\beta_i}{2} \cdot \left( \text{VAR}[Y_{ii}(x)] + \sum_{j \neq i} \text{VAR}[Y_{ji}(x)] \right) \\
&\leq \sum_i \frac{\beta_i}{2} \cdot \left( x_i \rho_{ii} (1 - \rho_{ii}) + \sum_{j \neq i} \text{VAR}[Y_{ji}(x)] \right) \\
&\leq \sum_i \frac{\beta_i}{2} \cdot \left( x_i (1 - \sum_{j \neq i} \rho_{ij}) \left( \sum_{j \neq i} \rho_{ij} \right) + \sum_{j \neq i} \text{VAR}[Y_{ji}(x)] \right) \\
&\leq \sum_i \frac{\beta_i}{2} \cdot \left( x_i \sum_{j \neq i} (1 - \rho_{ij}) \rho_{ij} + \sum_{j \neq i} \text{VAR}[Y_{ji}(x)] \right) \\
&\leq \sum_i \frac{\beta_i}{2} \cdot \left( \sum_{j \neq i} \text{VAR}[Y_{ij}(x)] + \sum_{j \neq i} \text{VAR}[Y_{ji}(x)] \right) \\
&= \sum_i \sum_{j \neq i} \left( \frac{\beta_i + \beta_j}{2} \right) \text{VAR}[Y_{ij}(x)].
\end{aligned}$$

□

**Lemma 4.3.2.**  $\sum_i \xi_i(t, x) + \frac{\Psi(x)}{2} < 0$

*Proof.* For simplicity, in this proof we omit the conditioning on  $X(t) = x$  where it is clear. Due to observation 4.2.3 we have,

$$\begin{aligned}
\sum_i \xi_i(t, x) + \frac{\Psi(x)}{2} &\leq \sum_i \frac{\beta_i}{2} \text{E}[u_i^+(x)^2] + \sum_i \frac{\beta_i}{2} \text{E}[u_i^-(x)^2] \\
&\quad - \sum_i \frac{\beta_i}{2} \text{E}[u_i^+(x) + u_i^-(x)] + \frac{\Psi(x)}{2} \\
&\leq \sum_i \frac{\beta_i}{2} \text{VAR}[\Delta X_i(t)] + \sum_i \frac{\beta_i}{2} \text{E}[u_i^+(x)]^2 + \sum_i \frac{\beta_i}{2} \text{E}[u_i^-(x)]^2 \\
&\quad - \sum_i \frac{\beta_i}{2} \text{E}[u_i^+(x) + u_i^-(x)] + \frac{\Psi(x)}{2}
\end{aligned}$$

Due to observations 4.2.4, 4.2.5, 4.3.1 and doing some substitutions, we get

$$\begin{aligned} \sum_i \xi_i(t, x) + \frac{\Psi(x)}{2} &\leq \sum_{i,j} \left[ \left( \frac{\beta_i + \beta_j}{2} \right) \text{VAR}[Y_{ij}(x)] \right. \\ &\quad \left. + (n-1) \cdot \left( \frac{\beta_i + \beta_j}{2} \right) \cdot \text{E}[Y_{ij}(x)]^2 - \left( \frac{\beta_i + \beta_j}{2} \right) \text{E}[Y_{ij}(x)] \right. \\ &\quad \left. + \left( \frac{f_j(x_j + 1) - f_i(x_i)}{2} \right) \text{E}[Y_{ij}(x)] \right] \end{aligned}$$

In the above inequality, we substitute  $\text{VAR}[Y_{ij}(x)]$  with  $x_i \rho_{ij} (1 - \rho_{ij})$ , and  $\text{E}[Y_{ij}(x)]$  with  $x_i \rho_{ij}$ . Based on the definition of  $\rho_{ij}$  each term in the above summation would be negative, and so the summation itself.  $\square$

**Corollary 4.3.3.**

$$\text{E}[\Phi(X(t+1)) | X(t) = x] - \Phi(x) \leq \frac{\Psi(x)}{2}$$

*Proof.* By applying Lemma 4.2.2 and 4.3.2, we have

$$\begin{aligned} \text{E}[\Phi(X(t+1)) | X(t) = x] - \Phi(x) &\leq \Psi(x) + \sum_i \xi(x, t) \\ &= \frac{\Psi(x)}{2} + \frac{\Psi(x)}{2} + \sum_i \xi(x, t) \\ &\leq \frac{\Psi(x)}{2}. \end{aligned}$$

$\square$

**Corollary 4.3.4.** *Assume that at step  $t$  the system is not at some Nash equilibrium. We have*

$$\text{E}[\Phi(X(t+1)) | X(t) = x] \leq \left( 1 - \frac{\alpha \cdot R^2}{4\beta \cdot n^2} \right) \Phi(x)$$



*Proof.*

$$\begin{aligned}
\mathbb{E}[\Psi(x)] &= \sum_{i,j} \mathbb{E}[Y_{ij}(x)] \cdot (f_j(x_j + 1) - f_i(x_i)) \\
&= \sum_{i,j} x_i \cdot \rho_{ij} \cdot (f_j(x_j + 1) - f_i(x_i)) \\
&= -\frac{1}{n} \sum_{i,j} \frac{1}{(\beta_i + \beta_j)} \cdot ((f_i(x_i) - f_j(x_j + 1)))^2 \\
&\leq \frac{-R^2}{2\beta \cdot n} \cdot \max_k f_k(x_k)^2 \\
&\leq \frac{-R^2}{2\beta \cdot n^2} \cdot \sum_{i=1}^n f_i(x_i)^2 \\
&\leq \frac{-\alpha \cdot R^2}{2\beta \cdot n^2} \Phi(x)
\end{aligned}$$

Due to corollary 4.3.3,  $\mathbb{E}[\Phi(X(t+1)) | X(t) = x] - \Phi(x) \leq \Psi(x)/2$ , thus with substitution the proof can be completed.  $\square$

**Lemma 4.3.5.** *Let  $Z(0), Z(1), \dots$  be a sequence of random variables with support set  $S = \{s_1, s_2, \dots, s_h\}$ . Suppose  $\exists \lambda \leq 1 : \mathbb{E}[Z(t+1) | Z(t) = s_k] \leq \lambda \cdot s_k$ , for all  $1 \leq k \leq h$ , and  $t \geq 0$ . We have*

$$\mathbb{E}[Z(t)] \leq \lambda^t \mathbb{E}[Z(0)]$$

for all  $t \geq 0$ .

*Proof.* See appendix 4.3.5.  $\square$

Next we proceed to show our main result of this section. For every assignment  $x = (x_1, \dots, x_n)$ , we define  $\Gamma(x)$  as following:

$$\Gamma(x) = \begin{cases} 0 & \text{if } x \text{ is a Nash equilibrium} \\ \Phi(x) & \text{otherwise.} \end{cases}$$

**Lemma 4.3.6.** *For every  $t \geq 0$ , we have*

$$E[\Gamma(X(t))] \leq \left(1 - \frac{\alpha \cdot R^2}{4\beta \cdot n^2}\right)^t \Gamma(X(0))$$

*Proof.* Setting  $\lambda = \left(1 - \frac{\alpha \cdot R^2}{4\beta \cdot n^2}\right)$ , by applying Corollary 4.3.4 and Lemma 4.3.5 is folklore.  $\square$

**Lemma 4.3.7.** *For  $t, \delta > 0$ , if  $E[\Gamma(X(t))] < \delta \cdot \Phi^*$  then  $X(t)$  is a Nash equilibrium with probability greater than  $\delta$ .*

*Proof.* Use Markov inequality.  $\square$

**Theorem 4.3.8.** *Given any initial load vector  $X(0) = x_0$ . Let  $T$  be the number of steps taken by the protocol in Algorithm 4 to reach a Nash equilibrium for the first time. Then,*

$$E[T] = O\left(\frac{1}{C} \log \frac{\Phi(x_0)}{\Phi^*}\right) = O\left(\frac{4\beta \cdot n^2}{\alpha \cdot R^2} \cdot \log \frac{\Phi(x_0)}{\Phi^*}\right)$$

## 4.4 Justification

In this section we show that the maximum slope of latency functions,  $\beta$ , is the relevant parameter in our analysis, and fast convergence is not possible under protocols which do not depend on some properties of latency functions.

### 4.4.1 The maximum slope is the relevant parameter

Assume there are  $n$  tasks and only 2 resources,  $R_1$  and  $R_2$ . The latency functions are  $f_1(k)$  and  $f_2(k)$  which defined as follows.

$$f_1(k) = \begin{cases} 0 & \text{if } k = 0 \\ \beta_1 & \text{if } 0 < k \leq m \end{cases}$$

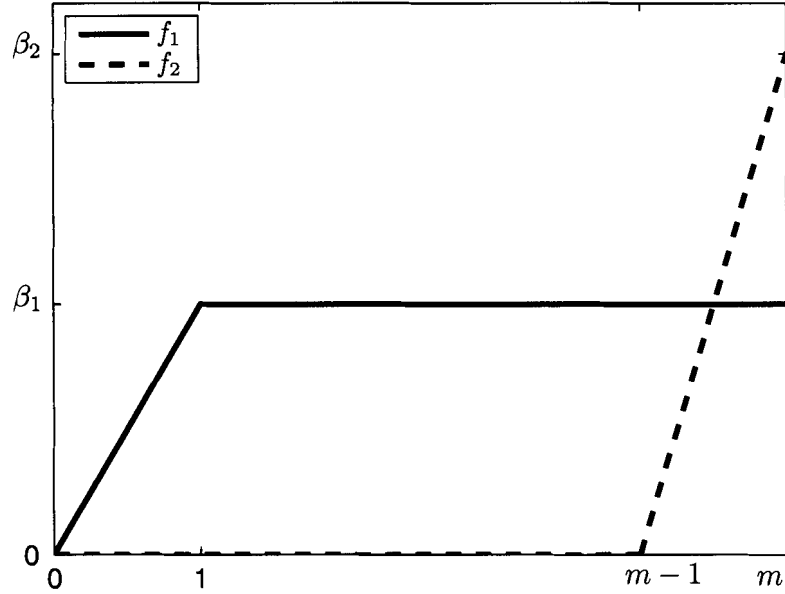


Figure 4.1:  $\beta$  is the relevant parameter for the analysis

$$f_2(k) = \begin{cases} 0 & \text{if } 0 \leq k < m \\ \beta_2 & \text{if } k = m \end{cases}$$

Let  $\beta_1$  be significantly larger than  $\beta_2$ . The only state at Nash equilibrium is when  $n - 1$  tasks share resource  $R_1$ , and a single task uses resource  $R_2$ . It is easy to observe that reallocation protocols like the ones in Algorithms 1, 2, and 3 in which the migration probability of a task only depends on the delay experienced by the task, and the current delay of the sampled resource, does not converge fast to the equilibrium.

## Future work

We considered three different models of the classical load balancing setting and studied the speed of convergence to Nash equilibria under reasonable reallocation protocols. Although we managed to achieve some of our goals, a number of open questions remains.

**The uniform case.** Comparing with the natural protocol in [4], we introduced a reallocation protocol whose migration probability had an extra multiplicative factor  $\rho = 1/8$ . Having the slowdown factor  $\rho$ , we showed that the expected potential would drop after each step. It is still an open question for us to show the necessity of the slowdown factor. We conjecture that the slowdown factor is only for the sake of analysis and one might come up with a better analysis of protocol in [4].

**The weighted case.** Although we showed a reasonable lower bound, our result is not the tightest one. An interesting open problem would be showing a tight analysis.

**Resources with different latencies.** Fischer et. al. in [20] use a technique so called *copying successful users strategies* to achieve fast convergence to Nash equilibrium for Wardrop's model. Although their protocol needs quite bit global knowledge, it is worth thinking about applying their technique to the atomic congestion game problems. Furthermore our results in this part is not tight. We believe that it is very hard to get better upper bounds, but still it is worth trying.

## Omitted proofs

### A.1 Proof of Part (2) of Observation 3.1.2

*Proof.* To prove Part (2), by definition of  $\Phi$  we have

$$\begin{aligned}
 & E[\Phi(X(t+1))|X(t) = x] \\
 &= \sum_{i=1}^n E[X_i(t+1)^2|X(t) = x] - n\bar{x}^2 \\
 &= \sum_{i=1}^n \left( E[X_i(t+1)|X(t) = x]^2 + \text{var}[X_i(t+1)|X(t) = x] \right) - n\bar{x}^2 \\
 &= \sum_{i=1}^n \left( E[X_i(t+1)|X(t) = x]^2 - \bar{x}^2 \right) + \sum_{i=1}^n \text{var}[X_i(t+1)|X(t) = x] \\
 &= \sum_{i=1}^n \left( E[X_i(t+1)|X(t) = x] - \bar{x} \right)^2 + 2\bar{x} \sum_{i=1}^n \left( E[X_i(t+1)|X(t) = x] - \bar{x} \right) \\
 &\quad + \sum_{i=1}^n \text{var}[X_i(t+1)|X(t) = x] \\
 &= \sum_{i=1}^n \left( E[X_i(t+1)|X(t) = x] - \bar{x} \right)^2 + \sum_{i=1}^n \text{var}[X_i(t+1)|X(t) = x].
 \end{aligned}$$

Notice that  $\sum_{i=1}^n E[X_i(t+1)|X(t) = x] = n\bar{x}$  and thus  $2\bar{x} \sum_{i=1}^n (E[X_i(t+1)|X(t) = x] - \bar{x}) = 0$ . □

## A.2 Proof of Lemma 4.3.5

*Proof.* Let  $s$  be a vector as following

$$[s_1, s_2, \dots, s_h]^T$$

and let  $A$  be a two dimensional matrix whose entry  $A_{ij}$  equals to  $P(X(t+1) = s_i | X(t) = s_j)$ .  $A_{ij}$  is independent of  $t$ . Our assumption implies  $A \cdot s \leq \lambda \cdot s$ . The rest of the proof is by induction on  $t$ . The base of the induction is trivial. Assume  $E[Z(t)] \leq \lambda^t E[Z(0)]$  as the induction hypothesis, then we have

$$\begin{aligned} E[Z(t+1)] &= A^{t+1} \cdot s \\ &\leq A \cdot \lambda^t \cdot s \\ &\leq \lambda^{t+1} \cdot s = \lambda^{t+1} \cdot E[Z(0)] \end{aligned}$$

□

# Bibliography

- [1] Heiner Ackermann, Heiko Röglin, and Berthold Vöcking. On the impact of combinatorial structure on congestion games. In *Proceedings of the 47th Annual IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 613–622, 2006.
- [2] B. Awerbuch, Y. Azar, and A. Epstein. The price of routing unsplittable flow. In *Proc. 37th Annual Symposium on Theory of Computing (STOC)*, pages 57–66, 2005.
- [3] Baruch Awerbuch and Robert D. Kleinberg. Adaptive routing with end-to-end feedback: distributed learning and geometric approaches. In *STOC*, pages 45–53, 2004.
- [4] P. Berenbrink, T. Friedetzky, L.A. Goldberg, P. Goldberg, Z. Hu, and R. Martin. Distributed selfish load balancing. In *Proc. of the 16th ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 354–363, 2006.
- [5] P. Berenbrink, T. Friedetzky, I. Hajirasouliha, and Z. Hu. Convergence to equilibria in distributed, selfish reallocation processes with weighted tasks. In *Proc. of the 15th Annual European Symposium on Algorithms (ESA)*, 2007.
- [6] P. Berenbrink, T. Friedetzky, I. Hajirasouliha, and P. Valadkhan. Convergence to equilibria in distributed, selfish reallocation processes with arbitrary resources. 2007.
- [7] P. Berenbrink, T. Friedetzky, and Z. Hu. A new analytical method for parallel, diffusion-type load balancing. In *Proceedings of the 20th IEEE International Parallel and Distributed Processing Symposium (IPDPS06)*, pages 1–10, 2006.
- [8] Petra Berenbrink, Artur Czumaj, Angelika Steger, and Berthold Vöcking. Balanced allocations: The heavily loaded case. *SIAM J. Comput.*, 35(6):1350–1385, 2006.
- [9] A. Blum, E. Even-Dar, and K. Ligett. Routing without regret: On convergence to nash equilibria of regret-minimizing algorithms in routing games. In *In Proc. of the 25th PODC Symposium*, pages 45–52, 2006.

- [10] Avrim Blum, Eyal Even-Dar, and Katrina Ligett. Routing without regret: on convergence to nash equilibria of regret-minimizing algorithms in routing games. In *PODC*, pages 45–52, 2006.
- [11] S. Chien and A. Sinclair. Convergence to approximate nash equilibria in congestion games. In *proc. of the 17th ACM-SIAM Symposium on Discrete Algorithms(SODA)*, 2007.
- [12] G. Christodoulou and E. Koutsoupias. The price of anarchy of finite congestion games. In *Proc. 37th Annual Symposium on Theory of Computing (STOC)*, 2005.
- [13] G. Christodoulou, V.S. Mirrokni, and A. Sidiropoulos. Convergence and approximation in potential games. In *STACS*, 2006.
- [14] A. Czumaj and B. Vöcking. Tight bounds for worst-case equilibria. In *Proc. 13th Annual Symposium on Discrete Algorithms, SIAM, Philadelphia, PA*, pages 413–420, 2002.
- [15] Kousha Etessami and Andreas Lochbihler. The computational complexity of evolutionarily stable strategies. *Electronic Colloquium on Computational Complexity (ECCC)*, (055), 2004.
- [16] E. Even-Dar, A. Kesselman, and Y. Mansour. Convergence time to nash equilibria. In *Proc. of the 30th International Colloquium on Automata, Languages and Programming (ICALP)*, pages 502–513, 2003.
- [17] E. Even-Dar and Y. Mansour. Fast convergence of selfish rerouting. In *Proc. of the 16th ACM-SIAM Symposium on Discrete Algorithms (SODA), Vancouver, British Columbia*, pages 772–781, 2005.
- [18] A. Fabricant, C.H. Papadimitriou, and K. Talwar. The complexity of pure nash equilibria. In *Proc. of the 36th Annual Symposium on Theory of Computing (STOC)*, pages 604–612, 2004.
- [19] S Fischer. *Dynamic Selfish Routing*. PhD thesis, Lehrstuhl für Algorithmen und Komplexität, RWTH Aachen, June 2007.
- [20] S. Fischer, H. Räckem, and B. Vöcking. Fast convergence to wardrop equilibria by adaptive sampling methods. In *Proc. 38th Annual Symposium on Theory of Computing (STOC), Seattle, WA.*, pages 276–283, 2006.
- [21] S. Fischer and B. Vöcking. Adaptive routing with stale information. In *Proc. 24th Annual ACM SIGACT/SIGOPS PODC Symposium*, pages 276–283, 2005.
- [22] Simon Fischer and Berthold Vöcking. On the evolution of selfish routing. In *ESA*, pages 323–334, 2004.
- [23] Simon Fischer and Berthold Vöcking. On the structure and complexity of worst-case equilibria. In *Proc. 1st Workshop on Internet and Network Economics (WINE)*, number 3828 in Lecture Notes in Computer Science, pages 151–160. Springer-Verlag, December 2005.



- [24] M. Goemans, V.S. Mirrokni, and A. Vetta. Sink equilibria and convergence. In *FOCS*, pages 142–154, 2005.
- [25] P. Goldberg. Bounds for the convergence rate of randomized local search in a multiplayer load-balancing game. In *Proc. of the 23rd Annual ACM SIGACT-SIGOPS Symposium on Principles of Distributed Computing (PODC), St. John's, Newfoundland*, pages 131–140, 2004.
- [26] Ara Hayrapetyan, Éva Tardos, and Tom Wexler. The effect of collusion in congestion games. In *STOC*, pages 89–98, 2006.
- [27] Ackermann Heiner, Skopalik Alexander, and Vöcking Berthold. On the complexity of pure nash equilibria in player-specific network congestion games. In *Proc. of the 3rd International Workshop on Internet and Network Economics (WINE) San Diego, USA*, 2007.
- [28] Samuel Ieong, Robert McGrew, Eugene Nudelman, Yoav Shoham, and Qixiang Sun. Fast and compact: A simple class of congestion games. In *AAAI*, pages 489–494, 2005.
- [29] E. Koutsoupias and C.H Papadimitriou. Worst-case equilibria. In *Proc. 16th Annual Symposium on Theoretical Aspects of Computer Science (STACS), Trier, Germany*, pages 404–413, 1999.
- [30] Vahab S. Mirrokni and Adrian Vetta. Convergence issues in competitive games. In *APPROX-RANDOM*, pages 183–194, 2004.
- [31] Michael Mitzenmacher. How useful is old information? *IEEE Trans. Parallel Distrib. Syst.*, 11(1):6–20, 2000.
- [32] J.F. Nash. Equilibrium points in n-person games. In *Proc. of National Academy of Sciences, volume 36*, page 4849, 1950.
- [33] G. Owen. *Game Theory, 3rd ed.* Academic Press, 1995.
- [34] R. W. Rosenthal. A class of games possessing pure-strategy nash equilibria. In *International Journal of Game Theory*, pages 65–67, 1973.
- [35] T. Roughgarden. Stackelberg scheduling strategies. In *Proc. 33rd Ann. ACM. Symp. on Theory of Comput. (STOC)*, pages 104–113, 2001.
- [36] T Roughgarden. *Selfish Routing*. PhD thesis, Cornell University, 2002.
- [37] T Roughgarden. *Selfish Routing and the Price of Anarchy*. MIT Press, 2005.
- [38] T. Roughgarden and E. Tardso. How bad is selfish routing? *Journal of the ACM*, pages 45–52, 2006.
- [39] A. Skopalik and B. Vocking. Inapproximability of congestion games. 2007.
- [40] Berthold Vöcking. Congestion games: Optimization in competition. In *ACiD*, pages 9–20, 2006.

- [41] J. G. Wardrop. Some theoretical aspects of road traffic research. In *Proc. of the Institute of Civil Engineers, Pt. II*, pages 325–378, 1952.
- [42] Jörgen Weibull. Evolutionary game theory. *J. Artificial Societies and Social Simulation*, 2(1), 1999.