CONSTRUCTION OF A BASKET OF DIVERSIFIED PORTFOLIOS, VIA QUANTUM ANNEALING, TO AID IN CARDINALITY CONSTRAINED PORTFOLIO OPTIMIZATION

by

Emilija Kolar
B.B.A.

and

Ehsan Seyedin
Master of Applied Science

PROJECT SUBMITTED IN PARTIAL FULFILLMENT OF
THE REQUIREMENTS FOR THE DEGREE OF

MASTER OF SCIENCE IN FINANCE

In the Master of Science in Finance Program
of the
Faculty
of
Business Administration

SIMON FRASER UNIVERSITY

Fall 2014

Approval

Name: **Ehsan Seyedin and Emilija Kolar**

Degree: **Master of Science in Finance**

Title of Project: **Construction of a basket of diversified portfolios via quantum annealing to aid in cardinality constrained portfolio optimization**

Supervisory Committee:

____

Title and Name
Senior Supervisor
Academic Chair

____

Title and Name
Second Reader
Adjunct Professor

Date Approved: ______________________________

Abstract

In this project, we propose and investigate a new approach for solving portfolio optimization problems (POP) with cardinality constraints using an evolutionary algorithm based on the distribution of *diversified baskets* (EADDB).

The Diversified basket is the basket of portfolios each of which obtains one of the lowest risks. The distribution of the diversified basket indicates the probability of having each asset in the *diversified basket*. Finding the diversified basket is an NP-hard problem, and we exploit quantum annealing in order to approximate the diversified basket.

In particular, POP is mapped into D-Wave Two™, the first commercially available quantum computer, using one of two methods: *discretization*, and *market graph*. Each approach creates several instances of the problem of finding diversified baskets. D-Wave Two's output is an approximation to this diversified basket, and subsequently the distribution of diversified basket can be determined. Distribution of the diversified basket forms the basis of EADDB. The performance of the proposed EADDB has been evaluated on the Hang-Seng in Hong Kong with 31 assets, one of the benchmark datasets in the OR Library, and has been compared with heuristic algorithms.

Keywords: Portfolio Optimization Problem; Quantum Annealing; Diversification; Evolutionary Algorithm; NP Hard Problem; D-Wave Two™

Dedication

To our families, for making this opportunity within our grasp.

Acknowledgments

We would like to acknowledge the support and guidance of our senior supervisor, Andrey Pavlov, our second reader, Phil Goddard and the support of 1QB Information Technologies, a leading quantum computing research and software development company.

Table of Contents

List of Figures

List of Tables

1.0 Introduction

1.1 Unconstrained Portfolio Optimization

The first quantitative approach for portfolio asset selection was the Mean-Variance model introduced by Markowitz (1952). This has become the standard method for solving for the combination of assets delivering the highest possible expected level of return for the given level for risk, which is the standard deviation of the returns. Markowitz does this by determining the amount of total wealth that is to be invested in each market asset to achieve a target level of return while finding the minimum amount of variance associated with that target with the maximum amount of diversification. Diversification implies that the attractiveness of an individual asset when held in a portfolio can be different from when it is a stand-alone asset, so the relationship between assets needs to be carefully considered. An important assumption is that returns follow a multivariate Gaussian distribution, Chang, Meade, Beasley and Sharaiha (2000) point out that this implies expected return and variance explain return on a portfolio of assets completely.

Using the standard Markowitz mean-variance approach the unconstrained portfolio optimization problem is given as:

$$\min_{x \in \mathcal{R}^n} f(x) = \sum_{i=1}^{n} \sum_{j=1}^{n} c_{ij} x_i x_j$$
$$\text{subject to } \sum_{i=1}^{n} x_i = 1$$
$$\sum_{i=1}^{n} x_i r_i = r^*$$
$$\delta \leq x_i \leq \varepsilon, \forall i \in \{1, \ldots, n\}$$

Where

n is the number of assets

c is the covariance between the return of assets i and j

x is the weight of asset held

r is the expected return of the asset

r* is the target expected return

δ is the lower bound on asset weight

ε is the upper bound on asset weight

$\sum_{i=1}^{n} x_i = 1$ is defined as a budget constraint, and $\sum_{i=1}^{n} x_i r_i = r^*$ is defined as a return constraint. The result of the Mean Variance approach is a convex "efficient frontier" line through a set of points on a variance-mean plane formed by sets of diversified portfolios offering a given return for a minimum amount of risk. In other words this is the optimal set of assets and their weights (how much of the investor's total wealth will be invested in each of these assets) an investor should choose given the level of return they are seeking to make.

The Mean-Variance problem is a polynomial-time optimization problem, and since the objective function is quadratic and convex and the constraints are linear, the resulting problem is a quadratic program (Dias, 2002). Due to these objective function properties the quadratic programming solvers are recognized to be the most efficient in terms of computing power. However as Chang, Meade, Beasley and Sharaiha point out there are two weaknesses with using Quadratic Programming (QP):

1. The underlying assumption of multivariate normality is violated.

2. Constraining the number of assets in the portfolio is not possible even though it has high practical significance. These cannot be translated into a linear constraint, therefore cannot be solved with Quadratic Programming.

1.2 Constrained Portfolio Optimization

Through the use of QP, the efficient frontier generated by all possible combinations of the assets will be smooth only when the number of assets available is infinitely large. Therefore the Markowitz portfolio tends to have unreasonably large quantities of individual assets, otherwise the frontier will not be a smooth curve, making quadratic programming obsolete because of nonlinear constraints.

Due to transaction costs and managerial concerns for the number of assets that can be effectively managed at a given time we will consider applying the cardinality constraint. The nonlinear cardinality constraint limits the number of assets in the portfolio causing the problem to become computationally intractable (NP-Hard); Therefore finding an exact solution in a reasonable amount of time is not possible with QP making Heuristic methods a viable alternative. This constraint changes the classical QP into a mixed integer quadratic programming (MIQP) problem by adding binary variables into the Markowitz model (Moral-Escudero, Ruiz-Torrubiano, and Suarez, 2006).

Using the Markowitz mean-variance approach the cardinality constrained portfolio optimization problem is given as:

$$\min_{x \in \mathcal{R}^n} f(x) = \sum_{i=1}^{n} \sum_{j=1}^{n} c_{ij} x_i x_j$$

$$\text{subject to } \sum_{i=1}^{n} x_i = 1$$

$$\sum_{i=1}^{n} x_i r_i = r^*$$

$$\delta \leq x_i \leq \varepsilon, \forall\, i \in \{1, \dots, n\}$$

$$\#\, \text{supp}\ (x) = K$$

Where K is the desired number of assets to be held in the portfolio, and $\#\,\text{supp}\ (x) = K$ is defined as a cardinality constraint.

Figure 1 indicates the unconstrained efficient frontier (green line) and constrained efficient frontier for the choice of 4 out of 22 assets for a sample dataset. The figure in the bottom indicates the asset allocation (weight of each asset on the efficient frontier) for expected returns ranging from 0 to .016.

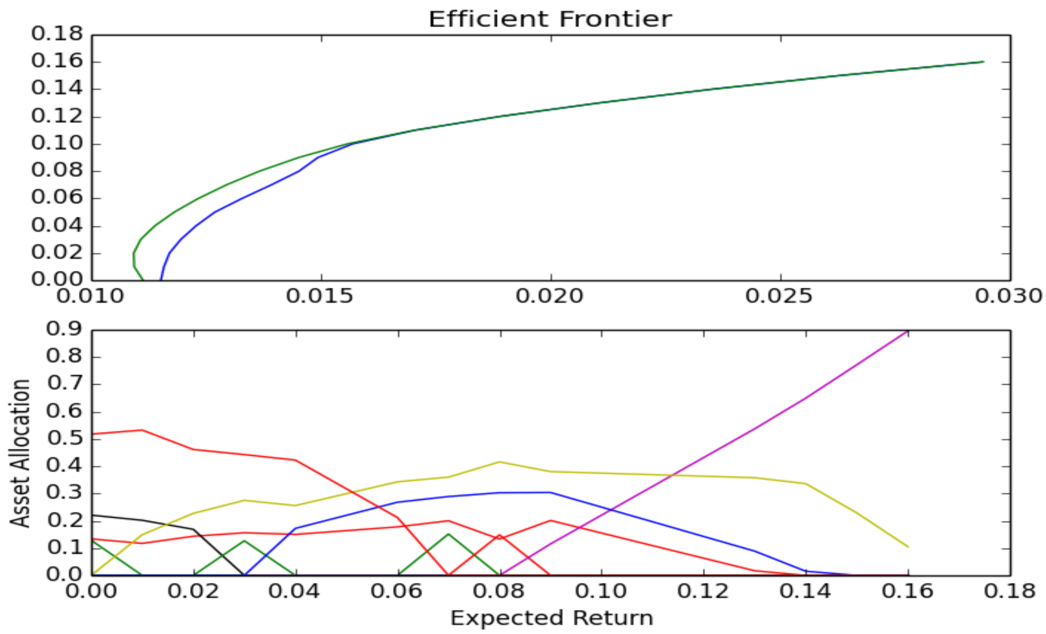

Figure 1: Efficient frontier and asset allocation found by exhaustive search for 22 Assets Choose 4

1.3 Outline of the Rest of the Paper

In section 2 we review the literature that deals with heuristic methods employed to solve portfolio optimization with a cardinality constraint. Section 3 describes the structure of genetic algorithms, and section 4 provides an introduction to the D-Wave Two™ quantum computer. An evolutionary algorithm based on the distribution of diversified baskets (EADDB) is discussed in section 5, which is followed by datasets in section 6 and experimental results in section 7. We then finish off by summarizing the significance of our results and introduce the idea of improving the EADDB method in the final sections, future work and conclusion.

2.0 Literature Review

Here we will review the literature that deals with heuristic methods to solve portfolio optimization with a cardinality constraint.

T.-J. Chang, N. Meade, J.E. Beasley, and Y.M. Sharaiha (2000) proposed finding the efficient frontier of a cardinality constrained portfolio by extending the standard mean-variance portfolio optimization model using three heuristic algorithms. The three algorithms were based on genetic algorithms, tabu search and simulated annealing. They revealed that by adding a cardinality constraint the efficient frontier could become discontinuous (Chang, Meade, Beasley, and Sharaiha, 2000). Using a heuristic technique an approximate solution can be found quickly where classic methods cannot find an exact solution or are too slow. In their approach, they assume that there is a decision maker who chooses the number of assets (K) by considering the trade-offs of risk, return and effort of managing a certain number of assets. In their Genetic Algorithm approach, Chang et al used a sample population size of 100 randomly generated portfolios. Each of these are then evaluated based on the objective function, achieving expected return with the minimum amount of variance possible. The GA then replaces the asset which the worst objective function value, continuously replacing dominated solutions. Using five data sets of 31 assets from Hang Seng, 85 assets from DAX 100, 89 assets from FTSE 100, 98 assets from S&P 100 and 225 assets from Nikkei 225. To benchmark their heuristics they compared their results to an unconstrained efficient frontier. They found that the GA heuristic, coded in FORTRAN and run on a Silicon Graphics Indigo workstation, was the best approximation tool to find the unconstrained efficient frontier. The results and algorithms have been made publicly available, and can be found along with more detail on the three heuristic approaches in their research paper.

Following the Chang et. al. (2000) paper, heuristic approaches have been further investigated in more detail for the purposes of greater efficiency and accuracy, supplemented by technological innovation (Woodside-Oriakhi, Lucas, Beasley, 2011) .

Using the data sets provided by Chang et. al. (2000), **Fernandez and Gomez in 2005** applied a method based on artificial neural networks, the Hopfield Network, to compute the cardinality constrained efficient frontier (CCEF). They found that using the Hopfield Network they were able to outperform the five benchmark problems by getting the lowest variance of return errors when the number of assets chosen to be in the portfolio is less than 10, upon which time it converges to the genetic algorithm (Fernandex and Gomez, 2005). They found that artificial neural network approach has an over-fitting problem and is readily trapped into local minima. This discovery was significant in that it validated ongoing exploration of the three heuristic approaches introduced to solve the CCEF. **Chiam, Tan and Mamum (2008)** also followed on the work done by Chang et. al. (2000) using evolutionary multi-objective (considering risk and return simultaneously) portfolio optimization to address the cardinality constraint which introduces a combinatorial optimization problem with a complex search space. In their approach the GA began with initializing a randomly generated population.

T. Chang, Yang, and K. Chang (2009) introduced a heuristic approach to portfolio optimization using the GA in different risk measures. They used historical daily data from the HANG SENG with 33 assets, FTSE with 93 and S&P 100 with 99 assets and set the cardinality constraint from 10 to 90, increasing it by an increment of 10 per run. To optimize they used the GA coded in C++ and run on a personal computer. They discovered that the CCEF with a value above one third of total assets in a index is dominated by those with a significantly smaller number of assets. They also showed that the computation time (CPU time) increased linearly with the increase of the number of assets in the portfolio and that investors should not waste their time trying to compute high cardinality constraint values since they are dominated by those with lower values. T. Chang, Yang, and K. Chang (2009) concluded that the GA method is an efficient tool to find efficient frontier based on a fixed amount of assets. In their approach the GA also began with initializing a randomly generated population.

Zhu, Y. Wang, K. Wang, and Chen (2011) introduced the Particle Swarm Optimization (PSO) technique, modeled to map out social behavior of bird flocking or fish schooling, to find the solution for the cardinality constrained optimization problem. PSO like the other heuristics reviewed starts off by initializing a population of random particles with an associated position and velocity where the search particles have a tendency to move towards the better search area over the objective function (Wang, Wang, and Chen, 2011). They compare the PSO approach to the GA. Like the GA, Zhu, Y. Wang, K. Wang, and Chen (2011) state that they are both population based, initialized with a population of randomly encoded solutions and the search updates the encodings over a series of iterations. The main difference is that PSO has no specific selection process. With their comparison, they found that the GA performed better in instances where the cardinality constraint was set to a significantly high value and encouraged further research to be done with hybrid models to improve efficiency.

Woodside-Oriakhi, Lucas, and Beasley (2011) stepped back from the heuristic approach to being their study. They utilized the latest CPLEX version 11.0 on the smaller test problems with up to 85 assets from the DAX 100. They concluded that solving for the CCEF QMIP using CPLEX was not computationally effective; therefore they were justified in applying a heuristic approach to solve for the CCEF. Their results showed that the most efficient heuristic approach is the GA using the data sets from the Chang, et. al. (2000). They innovated by pooling, consolidating efficient portfolios from each of the heuristics and eliminating any portfolios that are dominated, and then conducted another search. This resulted in a greater degree of accuracy but at the expense of efficiency.

3.0 Genetic Algorithms

Holland (1975) created the first Genetic Algorithm. The idea originated from Darwinian Evolution, it explores a solution space by applying the theory of natural selection. GA's belong to a class of stochastic search methods. To use it, solutions (in our case the portfolios with a chosen expected return and number of assets) are encoded into strings of bits (mimicking a decision based model using "0" if the asset is not present in the portfolio or "1" if the asset is selected to be in the portfolio). The fitness function, which evaluates the results of our objective function that achieves the expected return subject to minimizing variance on each portfolio generated, determines which of the initially generated random combinations of assets (portfolios) with defined parameters are the most suitable parent portfolios to generate the subsequent generation of portfolios (offspring). The higher the fitness value is, more optimal is the solution to the objective function (T.-J Chang, Yang, and K. Chang, 2009). The fitness function ranks the portfolios by variance, the larger the variance (risk) the lower the rank. The chosen parent portfolios generate subsequent portfolios (known as chromosomes in evolutionary terms) by applying crossover (if both parent portfolios contain the same asset the offspring portfolio will be assigned it in a random quantity between that of the two parents, but if only one parent has a certain asset then the offspring portfolio will have less change of being randomly being assigned that asset) and mutation values (usually done by randomly replacing one asset in the offspring portfolio by another which wasn't originally selected, introducing a small degree of stochastic variation, as seen in Woodside-Oriakhi, Lucas, and Beasley (2011)).

The basic steps of a simple GA as defined by T.-J. Chang, Yang, and K. Chang (2009):

1) Generate the initial randomly generated population
2) Evaluate the fitness of the individuals within the generated population
3) Select parents from the population considering applied constraints and carry them to the next generation applying crossover, mutation and reproduction.
4) The last population will be replaced by the new population
5) Repeat step 2 if the termination condition is not satisfied

The termination condition is very important for finding the optimal risky portfolio, due to the trade-off between efficiency and accuracy. The number of generations (iterations) needed to find the optimal solution has a large range because the initial population (set of portfolios) is randomly created and because of the probabilistic nature of the mating/pairing process (Wallace, 1994).

4.0 Technological Innovation: D-Wave Two™ Quantum Computer

4.1 The Hardware

The financial industry is among the strongest growth sectors for supercomputers driven by exponentially increasing data volumes causing greater data complexity. Currently the market is saturated by classical computers which perform one algorithmic operation at a time, quantum computing technology has the potential to perform multiple operations simultaneously.

D-Wave™ is a quantum computing company creating the first commercially available quantum computer, which uses adiabatic quantum annealing to solve intractable optimization problems. The technology we will be testing in our study to help us optimize a cardinality constrained portfolio will be an adiabatic quantum computer, the D-Wave Two, which has not yet been applied to the field of finance.

The niobium chip at the heart of the D-Wave is chilled to induce the quantum state. With the user modeling a problem into a search for the absolute minimum, The D-Wave Two has 512 qubits giving it the ability to perform 2^512 operations simultaneously to allow it to determine the lowest energy required to form the solution using the process of “Quantum Annealing” (Bunyk, Hoskinson, Johnson, Tolkacheva, Altomare, Berkley, Harris, Hilton, Lanting, and Whittaker, 2014). Neven, Denchev, Rose, and Macready (2009) have shown that the adiabatic quantum computer will yield solutions superior to those which can be achieved by classical heuristic solvers for discrete global optimization. We will explore this technology’s ability to improve the efficiency and accuracy of current cardinality constrained portfolio optimization methods. As the number of entangled qbits increases, with development of the technology, the size of the intractable optimization problems that can be addressed efficiently and accurately will increase. We anticipate that with the pace of development that D-Wave has shown, the D-Wave quantum computers, in conjunction with optimization software development, have a chance at becoming the next competitive advantage for financial institutions.

4.2 Ising Model

Quantum Annealing (QA) is a general method for finding the global minimum of a given objective function over a given set of possible solutions by a process using quantum fluctuations (Seki and Nishimori, 2012). A physical quantum annealing process is used by D-Wave hardware for an approximate minimization for a certain class of Ising objective functions. Optimization of an Ising objective function is an NP-hard problem. The Ising minimization problem of n variables $s_i \in \{-1, +1\}$ is given by:

$$\min_{\boldsymbol{s}} \sum_{(i,j)\in E} s_i s_j J_{ij} + \sum_{i \in V} s_i h_i$$

where $\boldsymbol{s} = [s_1, \dots, s_n]$, and J_{ij}’s and h_i’s are real coefficients.. Information about the Ising model is expressed by its primal graph $G = (V, E)$, where V is a set of vertices corresponding to the optimization variables, and E is the set of edges in the primal graph. Without loss of generality we can assume that $\boldsymbol{J}$ is upper-triangular. Sometimes, it is more convenient to model with binary variables rather than $\{-1, +1\}$ variables. Quadratic Unconstrained Binary Optimization problems (QUBOs) are of the form:

$$\min_{\boldsymbol{x}} \sum_{i>j} x_i x_j Q_{ij}$$

where $x_i \in \{0, 1\}$, and Q_{ij}’s are real quoficients. QUBOs are NP-hard problems, and are shown to be an effective representation of discrete optimization problem models. Ising and QUBO models are related to each other through the transformation $= \frac{s+1}{2}$.

Portfolio optimization problem is mapped into D-Wave Two using one of two methods: *discretization*, and *market graph* and we explore the fitness of each method. Each approach creates several instances for the problem of finding diversified baskets by modeling POP as a QUBO. The output of D-Wave Two is an approximation to this diversified basket, and subsequently the probability distribution of the diversified basket can be determined. The evolutionary algorithm based on the distribution of *diversified baskets* (EADDB) for solving cardinality constrained portfolio optimization problems is described in the next section.

5.0 Methodology: EADDB

In what follows, we propose an algorithm using ideas of evolutionary algorithms for portfolio optimization in the presence of cardinality constraints.

5.1 Discretization

Preliminaries

For a vector $\{x_1, x_2, \ldots, x_n\}$ in real or binary variables we define the support set of *x* to be

$$\text{supp }(x) = \{i: x_i \neq 0,\ i = 1, \ldots,\ n\}$$

Definition (CQBO)

A degree two polynomial in binary variables is called a Cardinality Constrained, Quadratic, Binary Optimization problem if it is equipped with a cardinality constraint # supp (δ) ≤ K on its feasible region.

Definition (Discretization)

Given a quadratic polynomial:

$$q = q(x_1, x_2, \ldots, x_n) = x^t \sum x + c^t x$$

In real variables and with a *departure point* $(x_1^0, \ldots, x_n^0)$ the discretization of *q* is the quadratic binary polynomial:

$$\tilde{q} = q(\delta_1, \delta_2, \ldots, \delta_n) = \tilde{x}^t \sum \tilde{x} + c^t \tilde{x}$$

Where $\tilde{x} = q(x_1\delta_1, x_2\delta_2, \ldots, x_n\delta_n)$.

We can solve the minimization problem of discretization of a quadratic real polynomial using QA and furthermore can solve cardinality constrained versions of the problem.

Definition

By a *population* we mean a matrix of zeros and ones, where each row indicates a support set.

Example.

For example if a population is given by:

$$X_0 = \begin{pmatrix} 1\ 0\ 0\ 1\ 1\ 1 & \cdots & 1 \\ \vdots & \ddots & \vdots \\ 1\ 1\ 1\ 0\ 0\ 1 & \cdots & 0 \end{pmatrix}$$

Then the first row vector $(1, 0, 0, 1, 1, \dots, 1)$ indicates the region of points supported by assets$(1, 4, 5, \dots, n)$. One way of constructing populations is to solve a binary representation of the Min-Variance Portfolio.

5.2 Main algorithm

Given *n* assets indexed by integers $\{1, \dots, n\}$ we let r_i be a constant called *expected return* of asset *i*. A symmetric positive semi-definite *n* × *n* matrix *C* = (c_{ij}) is also given. Let $\sigma_i = \sqrt{c_{ii}}$ be the risk associated to asset *i* and let the *correlation* between assets *i* and *j* be denoted by ρ_{ij}

$$\rho_{ij}\sigma_i\sigma_j = c_{ij}$$

Problem

The portfolio optimization problem (Markowitz model) with cardinality constraint is

$$\min_{x \in \mathcal{R}^n} f(x) = \sum_{i=1}^{n} \sum_{j=1}^{n} c_{ij} x_i x_j$$

$$\text{subject to } \sum_{i=1}^{n} x_i = 1$$

$$\sum_{i=1}^{n} x_i r_i = r$$

$$\delta \leq x_i \leq \varepsilon, \forall\, i \in \{1, \dots, n\}$$

$$\#\,\text{supp}\,(x) = K$$

Solution

1. First generate an initial population (X_0) by solving the discretization of $f(x)$ around departure point $(\frac{1}{K}, \dots, \frac{1}{K})$, in the presence of the cardinality constraint. An alternative selection strategy for the initial population is solving the quadratic program by removing the cardinality constraint and using that solution as the departure point.

Set L = 0 (first iteration of the algorithm).

<u>In iteration L of the algorithm, given population X_L:</u>

2. Find $P(x_i = 1) = p_i$ (probability of variable x_i being 1 in a generated population), for all variables by counting the number of 1's in *i-th* column of the Initial Population X_L, and then dividing the sum by *M*, i.e. $P\ (x_i = 1) = p_i = \frac{\sum_{l=1}^{M} X_L(l,i)}{M}$. M is the number of rows of the Initial Population X_L.

3. Draw a new population, $\mathfrak{X}_L$, using p_i's (we generate strings of 1's and 0's). If the number of 1's in a row is greater/smaller than K, we randomly removes/adds 1's in order to have exactly Knumber of 1's in each row.

Example. $\mathfrak{X}_L$ is a new population drawn from p_i's:

$$\mathfrak{X}_L = \begin{pmatrix} 1\ 0\ 1\ 1\ 1\ 0 & \cdots & 1 \\ \vdots & \ddots & \vdots \\ 0\ 1\ 1\ 0\ 1\ 1 & \cdots & 0 \end{pmatrix}$$

4. Use Quadratic Programming (QP) for finding the best solution of the continuous optimization problem for each row of $\mathfrak{X}_L$. W_L is a matrix in which each row indicates the best solution of the problem found by QP.

Example.

$$\mathfrak{X}_L = \begin{pmatrix} 1\ 0\ 1\ 1\ 1\ 0 & \cdots & 1 \\ \vdots & \ddots & \vdots \\ 0\ 1\ 1\ 0\ 1\ 1 & \cdots & 0 \end{pmatrix}$$

And after QP we get

$$W_L = \begin{pmatrix} 0.2\ 0\ 0.3\ 0.1\ 0.1\ 0 & \cdots & 0.4 \\ \vdots & \ddots & \vdots \\ 0\ 0.3\ 0.5\ 0\ 0.05\ 0.15 & \cdots & 0 \end{pmatrix}$$

Note. So far, each row in W_L has met all problem constraints (budget constraint, return constraint and cardinality constraint).

5. Select Parents. Parents are M' strings in W_L with lowest values of objective function f.
6. Use linear combinations of parents to generate a new matrix $\widetilde{W}_P$, in which each row is a linear combination of the selected parents.

Note. Let $W_{L,j}$ be the *j-th* row of matrix W_L. Then the *i-th* string of $\widetilde{W}_L$ is:

$$\widetilde{W}_{L,i} = \sum_{j=1}^{M\prime} W_{L,j}\, \pi(j,i)$$

where M' is the number of selected parents, and $\pi(j, i)$'s are the linear combination factors. Both M' and $\pi(j, i)$'s are parameters of the algorithm which might vary from one simulation run to another, and finding their optimal values is an open research problem.

Note. All row vectors $\widetilde{W}_{L,i}$ meets the return and budget constraints but violate the cardinality constraint.

8. For each row vector $\widetilde{W}_{L,i}$, solve the associated discretization of $f(x)$ in the presence of the cardinality constraint, and generate a new matrix X_{L+1}. If the number of iterations is less than the *maximum number of iterations*, go to step 2. The *maximum number of iterations* is another parameter of the algorithm, and finding its optimal value is an open research problem.

Note. Different stopping criteria such as convergence, number of iterations, etc could also be considered.

5.3 Alternative Approach: Market Graph

5. 3. 1 Introduction to market graphs

A market graph is a simple undirected graph depending on a constant threshold $\theta \in [-1, 1]$ with *n* vertices corresponding to *n* assets and is constructed in the following manner. Consider correlation coefficients

$$\rho_{ij} = \frac{\langle R_i R_j \rangle - \langle R_i \rangle \langle R_j \rangle}{\sqrt{\langle R_i^2 - \langle R_i \rangle^2 \rangle \langle R_j^2 - \langle R_j \rangle^2 \rangle}}$$

Where

$$R_i(t) = \ln \frac{P_i(t)}{P_i(t-1)}, \text{ and } \langle R_i \rangle = \frac{1}{N} \sum_{i=1}^{N} R_i(t)$$

and $P_i(t)$ is the price of asset *i* as a function of time. Then vertices *i* and j are connected by an edge whenever $\rho_{ij} \geq \theta$ (Boginski, Butenko, and Pardalos, 2005), (Jallo, and Budai, 2010).

Note. A maximum independent set of a market graph or a weighted market graph given a threshold θ is known as a diversified portfolio, for the correlation between any pair of assets in the portfolio is less than threshold θ.

The differentiating factor of our approach compared to that of the approaches explored in the literature review is to search the space first by exploring diversified portfolios and then finding the probability of each asset being in the diversified portfolios.

5.3.2 Population generation using market graphs

An alternative method of generating the population given a departure point $(x_1^0, \dots, x_n^0)$ is the following:

(a) Draw a market graph with parameter θ

(b) Find the Maximum Independent Set (MIS) (Bondy, and Murty, 1976) of the graph using QA (Choi, 2010).

(c) All possible MIS's for a given θ, and other MIS's of graphs generated by different θ's will construct our initial population. Note that there is no edge connected between any pair of the nodes in an MIS of the market graph, and consequently the correlation between any pair of the assets in the MIS is less than threshold θ. In other words, the assets in the MIS are uncorrelated with each other, and they form a diversified portfolio. Thus, all possible MIS's for a given θ constructs a diversified basket, which is the initial population of the algorithm.

The population is now a matrix in zeros and ones in which each row is a MIS of the weighted market graph for some θ. Two rows might be generated using different thresholds. $(X_0)_{ij} = 1$ Indicates that asset j is included in the i-th MIS.

Note. We are considering "all" possible MIS's of weighted market graphs – an MIS violating the cardinality constraint is included in the X_0 as well.

6.0 Datasets

We have used different datasets for performance evaluation of EADBB. Due to an excessive processing time for a larger number of assets in the discretization method, only portfolios with less than the number of assets compared to the Market graph method have been considered.

6.1 Datasets used in discretization's method

The first dataset includes daily prices of 22 stocks from January 2000 through to August 2013 (stock tickers: AAN, AA, ABC, ABT, ABV, AB, ABX, AB, ACE, ACI, ACO, ACT, ADC, ADM, AEC, AEE, AEG, AEM, AEO, AEP, AES, AET). Daily returns have been calculated using $R_i(t) = \ln \frac{P_i(t)}{P_i(t-1)}$.

Expected returns and covariance of stocks' returns have been computed accordingly using standard models/approaches. We have performed experiments on portfolios' with different number of assets ranging from 12 to 22 with the cardinality constraint set fixed at $K = 4$. Lower bound of each asset's weight in the support is set to $\delta = 0$, and the upper bound is set to $\varepsilon = 1$. The portfolio optimization problem has been solved by EADDB for 17 return points, ranging from 0 to 0.16 with increments of 0.01.

6.2 Datasets used in Market Graph's method

We have used a dataset for the portfolio optimization problem from the OR-Library (Beasley, 1990). We have evaluated EADDB's performance for port-1 (Hang-Seng in Hong Kong), one of the benchmark datasets in the OR Library (Beasley, 1990). Expected returns, and correlation between any pair of assets are given. In our experiments, the number of assets is 31, and the cardinality constraint is set to $K = 10$. The lower bound of each asset's weight in the support is set to $\delta = 0.01$, and the upper bound is set to $\varepsilon = 1$. The portfolio optimization problem has been solved by EADDB for 50 return points, ranging from 0.003 to 0.009 with equal return increments of 0.000135.

7.0 Experimental Results

It should be noted that the basic idea of EADDB is to find the mean-variance portfolio by reducing the search space using the distribution of a diversified basket. The results shown in this section are for the purpose of exploring the usefulness of the distribution of diversified basket method, so the results are instances of our simulation runs, and are not necessarily the best results that can be obtained by EADDB. In the future work section we elaborate on potential improvements to EADDB.

7.1 Discretization

Performance of EADDB has been evaluated in comparison with exhaustive search results on the given test dataset referenced in the dataset section above. The Number of iterations is set to be 4 in the evolutionary algorithm where the size of the population is set to the $number\ of\ assets$^2. Figure 2 demonstrates the efficient frontiers obtained by EADDB for the choice of 4 out of 12 assets, where the green line is the unconstrained and the blue line is the constrained efficient frontier. Efficient frontiers found by exhaustive search are shown in figure 3.

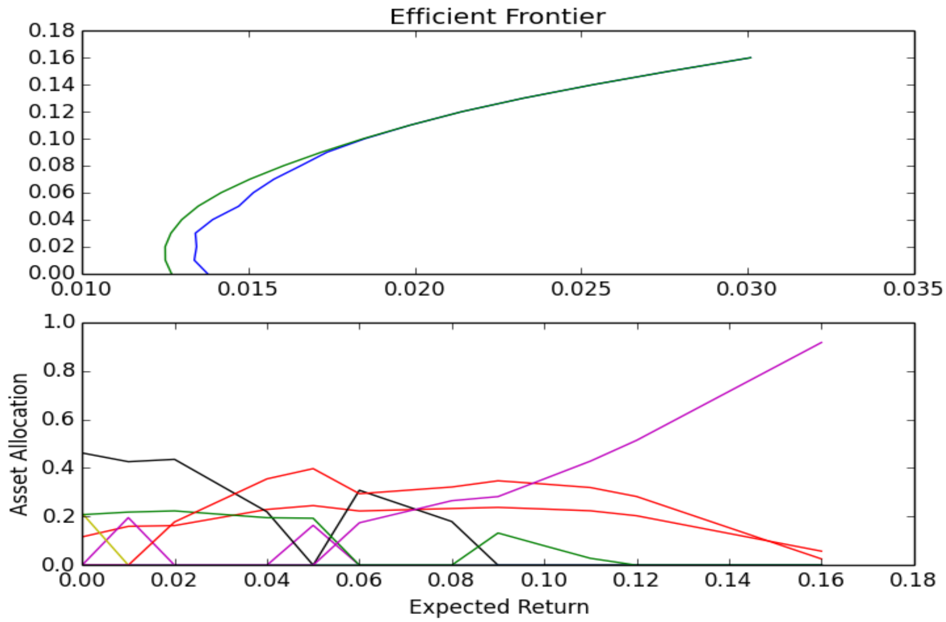

Figure 2: Efficient frontiers obtained by EADDB for 12 Assets Choose 4 using discretization

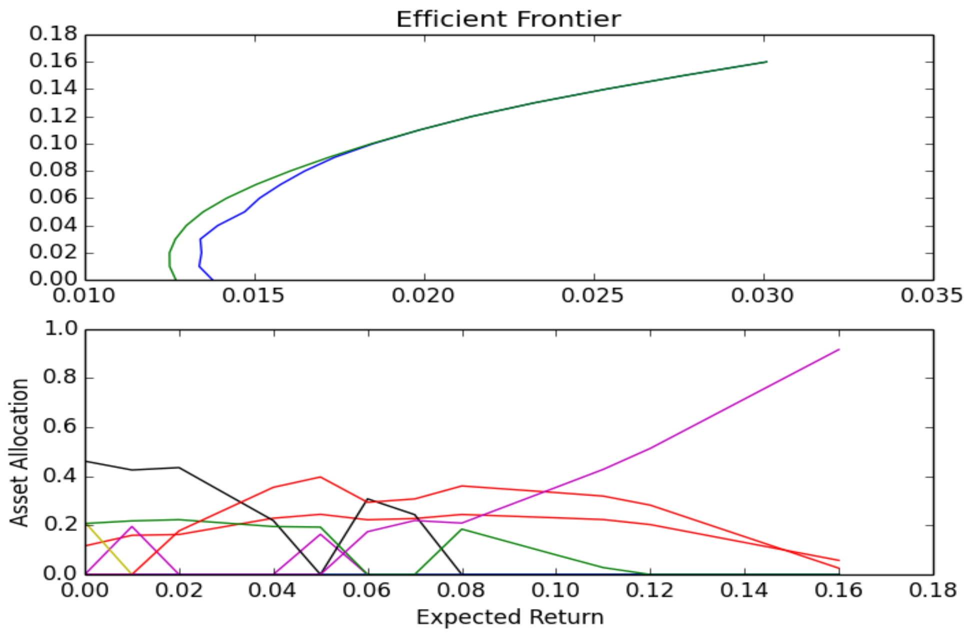

Figure 3: Efficient frontiers obtained by Exhaustive Search for 12 Assets Choose 4

Evolution of EADDB is demonstrated in Figure 4 for expected return 0.08, which is the only point missed by EADDB as shown in Figures 2 and 3. We consider the distribution of the diversified basket and the evolution of the algorithm for this return, Figure 4. In Figure 4, the orange bars in the first row indicate the weights of assets in the optimal support found by exhaustive search. Bars shown from the second row to the fifth row, which are labeled by Iterations 1 to 4, indicate the distribution of the diversified basket in each iteration of EADDB. Values marked under the horizontal line are weights of assets in a support with minimum risk found by EADDB.

The higher the bar, the more the associated asset with the bar occurs in the diversified basket. EADDB draws populations from the distribution of the diversified basket, which implies that assets with higher bars are most likely to occur in a drawn support. In other words, assets with the lowest bars are not likely to be in a drawn support, and consequently they are unlikely to be in a support which results in the mean-variance portfolio. Figure 4 indicates how likely the missed point in the diversified basket was. Experimental simulation runs for various expected returns demonstrate that global bests include rarely assets associated with the lowest-heights (i.e., the lowest probabilities in the diversified basket). Improved versions of EADDB can potentially employ new techniques using such information for reducing the search space in finding mean-variance portfolio.

Processing time in our empirical simulations was roughly 100 seconds in the case of 12 choose 4, and 500 seconds in the case of 22 choose 4. Although drawing the initial population from D-Wave output seems to be an effective departure point in our evolutionary algorithm, it is at the expense of the time it takes for D-Wave Two to solve the problem. Therefore the discretization method is not a good fit to be solved using D-Wave Two. This issue has been resolved by the market graph method, discussed in the next section.

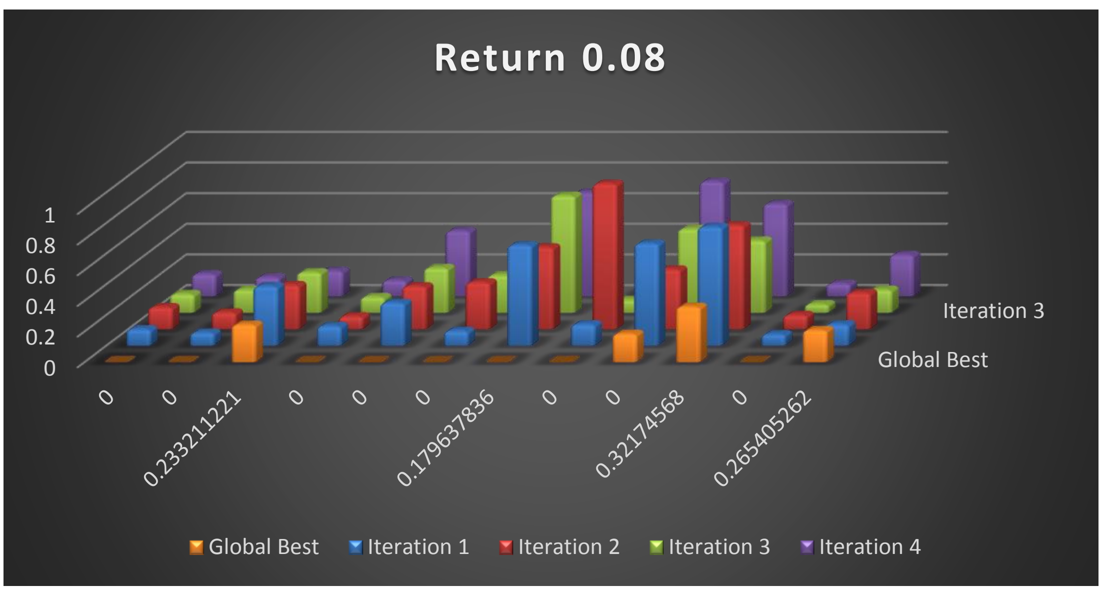

Figure 4 Evolution of EADDB for expected return 0.08 using discretization for 12 choose 4

For the case of 22 choose 4, EADDB misses more points. Figures 5 and 6 indicate the results using exhaustive search and EADDB for 22 choose 4, respectively.

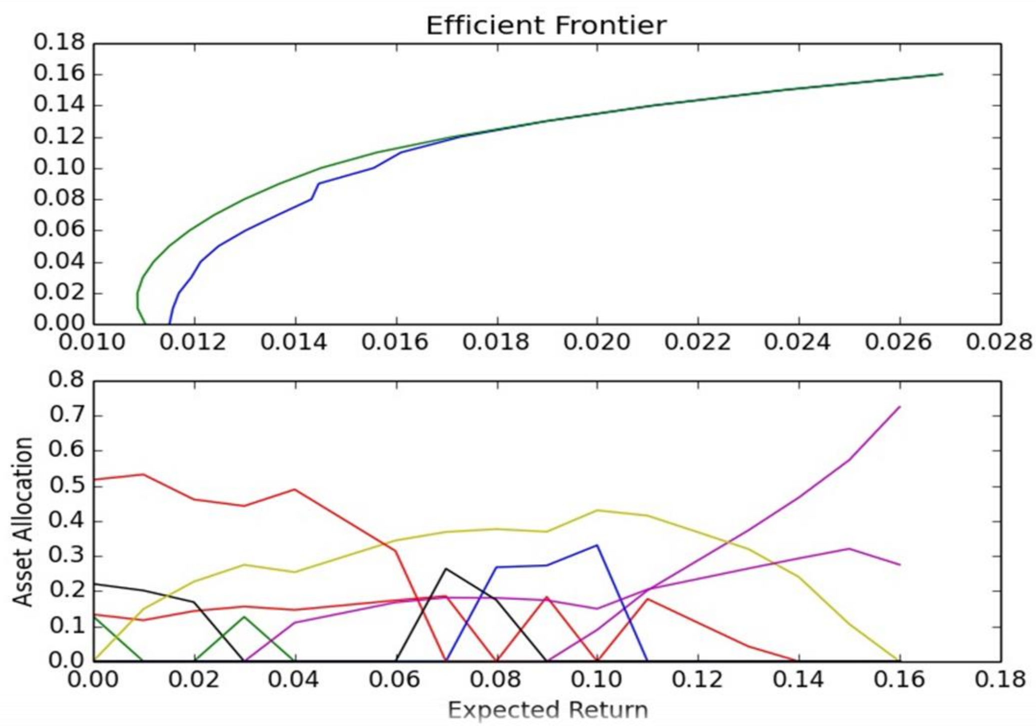

Figure 5 Efficient frontier and asset allocation found by exhaustive search for 22 choose 4

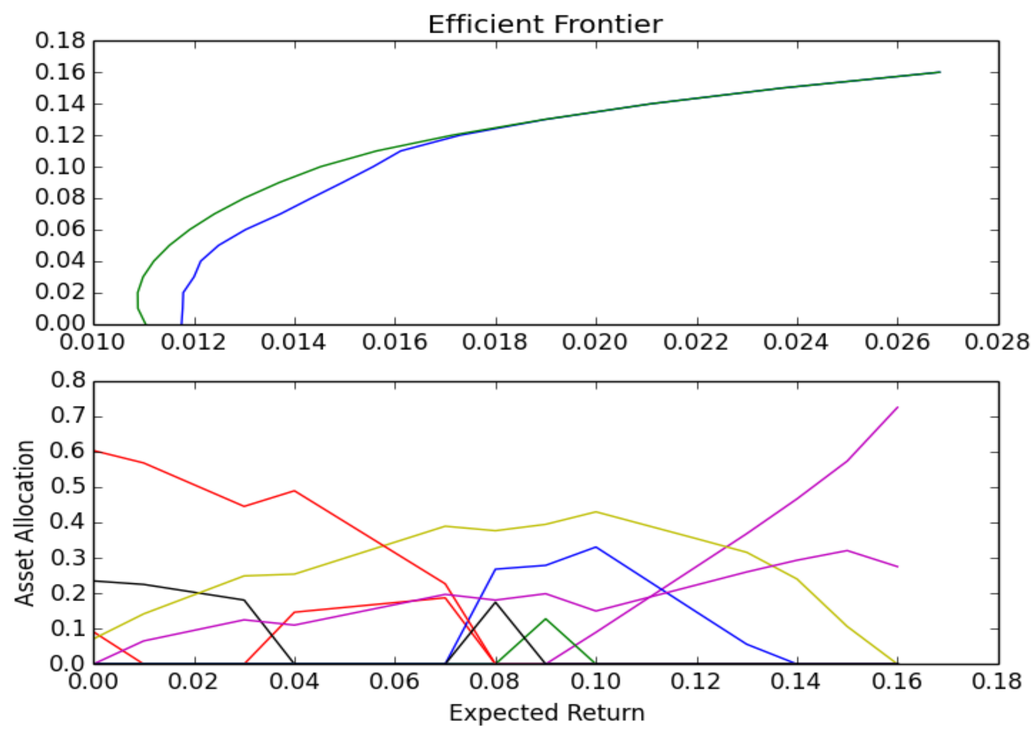

Figure 6 Efficient frontier and asset allocation found by EADDB for 22 choose 4 using discretization

Evolution of EADDB is demonstrated in Figure 7 for an expected return of 0.05. In this example, EADDB has found the global best of the objective function, ie optimal weights of the support with minimum risk. Note that assets with the lowest probabilities in the diversified basket are not included in the global best.

Figure 7 Evolution of EADDB for expected return 0.05 in case of 22 choose 4

7.2 Market Graph

We have compared the performance of EADDB for the choice of 10 out of 31 assets with heuristic algorithms used globally as benchmarks for POP with cardinality constraints (Chang et. al., and Woodside-Oriakhi et. al.). The number of iterations is set to 10 and the size of population is set to 50 in the evolutionary algorithm. Both evolutionary algorithm parameters (population size and number of iterations) have been chosen to be small numbers so it can give an insight about usefulness of the distribution of the diversified basket.

Figure 8 demonstrates the efficient frontier and asset allocation obtained by EADDB for the cardinality constrained portfolio. Computing exhaustive search results is a tedious task in this case. The error percentage is calculated based on the standard method explained in detail by Chang et. al. and T. Chang et. al.. It is defined as the distance between the constrained efficient frontier and the unconstrained efficient frontier, hence the unconstrained efficient frontier is drawn for the purpose of error calculations. Performance metrics in terms of computational time, error percentage, population size and number of iterations are given for EADDB and other heuristic algorithms in Table 1. As discussed earlier, parameters of evolutionary algorithm in EADDB have not been tuned to be optimal, and the results in table 1 are instances of our simulation runs, and are not necessarily the best answers that can be obtained by EADDB.

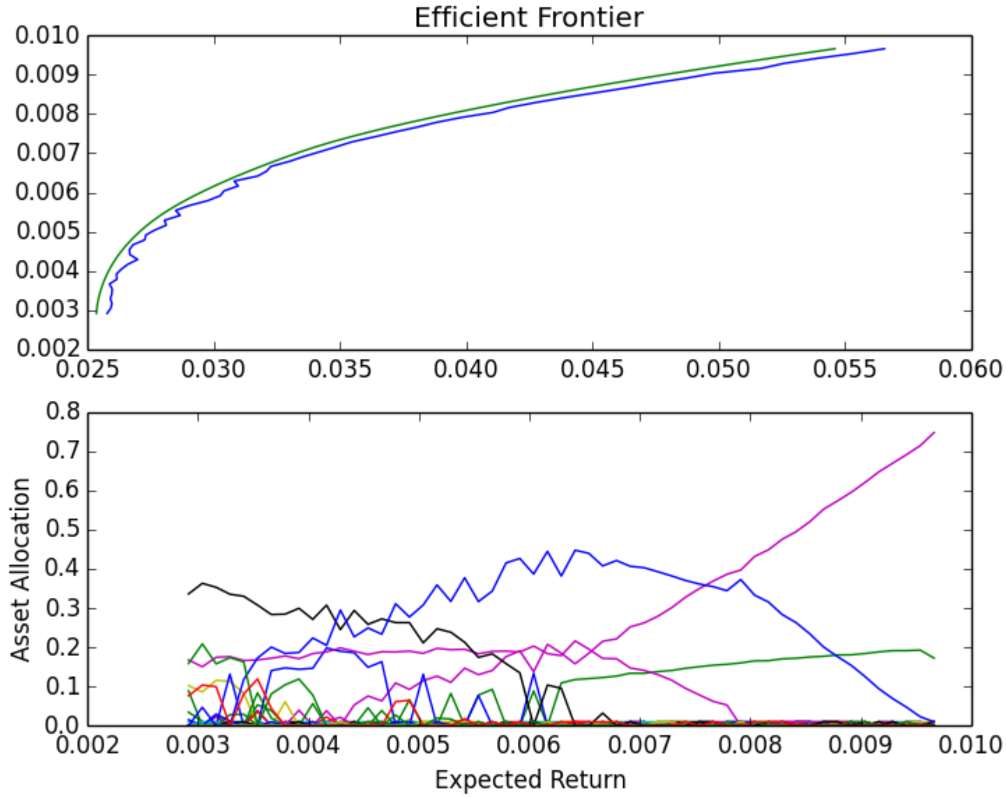

Figure 8: Efficient frontier and asset allocation obtained by EADDB for 31 choose 10 with market graph approach

	Population size	# of iterations	Time (s)	Average Error
Chang. et. al.	100	31000	172	0.9457
Woodside-Oriakhi et. al.	100	N/A	76	0.85
EADDB	50	10	529	2.3

Table 1: Performance metrics in terms of computational time, error percentage, population size and number of iterations for EADDB and other heuristic algorithms

It should be noted that the market graph method gave superior performance in terms of computational time compared to the discretization method because it fits D-Wave Two to a greater degree.

8.0 Future Work

This paper is the first research done in determining the usefulness of the distribution of diversified baskets in portfolio optimization, and it is a broad area of research which can be explored further in the future. New evolutionary techniques can be applied to make EADDB more efficient in terms of computational time and error. In particular, the more the problem is fit to D-Wave, the faster the evolutionary algorithm can be. Experimental simulation runs for various expected returns demonstrate that global bests rarely include assets associated with the lowest probabilities in the diversified basket. Improved versions of EADDB can potentially employ new techniques using this information for reducing the search space in finding efficient mean-variance portfolios.

9.0 Conclusion

In this project, we have proposed a new approach for constructing of diversified portfolios via quantum annealing, and have solved portfolio optimization problems with cardinality constraints using an evolutionary algorithms based on the distribution of *diversified baskets* (EADDB). In particular, POP is mapped into D-Wave Two using one of two methods: *discretization*, and *market graph*. The output given by the D-Wave Two is an approximation to this diversified basket, and subsequently the distribution of diversified basket can be determined. The distribution of the diversified basket forms the basis of EADDB. The performance of the proposed EADDB has been evaluated on the Hang-Seng in Hong Kong with 31 assets, one of benchmark datasets in the OR Library (Beasley, 1990), and has been compared with heuristic algorithms.

References:

Beasley, J.E., 1990. OR-Library: Distributing test problems by electronic mail. Journal of the Operational Research Society 41, 1069–1072. http://people.brunel.ac.uk/~mastjjb/jeb/orlib/portinfo.html

Boginski, V., Butenkob, S., Pardalosa, P., 2005. Mining market data: A network approach. Department of Industrial Engineering, 77843-3131

Bondy, John Adrian; Murty, U.S.R. (1976), Graph Theory with Applications, North-Holland p.101

Bunyk, P.I., Hoskinson, M., Johnson, W., Tolkacheva, E., Altomare, F., Berkley, A.J., Harris, R., Hilton, J.P., Lanting, T., Whittaker, J. (2014). Architectural considerations in the design of a superconducting quantum annealing processor. D-Wave Systems Inc. Retrieved from arXiv.org > quant-ph > arXiv:1401.5504

Chang, T.J., Meade, N., Beasley, J.E., Sharaiha,Y.M., 2000. Heuristics for cardinality constrained portfolio optimisation. Computers & Operations Research 27 1271-1302

Chang, T.J., Yang, S.C., Chang, K.J., 2009. Portfolio optimization problems in different risk measures using genetic algorithm. Expert Systems with Applications, International Journal 36 (2009) 1052-10537

Choi, V., 2010. Adiabatic Quantum Algorithms for the NP-Complete Maximum-Weight Independent Set, Exact Cover and 3SAT Problems. Quantum Information and Computation, Vol. 11, No. 7&8 (2011) 0638-0648

Dias, F., 2002. Quadratic Programming Applied to Modern Portfolio Selection, *working paper, Instituto de Matemática e Estatística Universidade de São Paulo, Brazil*

Fernandez A., Gomez S., 2005. Portfolio selection using neural networks. Computers and Operational Research 34 (2007) 1177-1191

Holland, J.H., 1975. Adaptation in Natural and Artificial Systems: An Introductory Analysis With Applications to Biology, Control, and Artificial Intelligence. University of Michigan Press, Ann Arbor, MI, USA.

Markowitz H., 1952. Portfolio selection. Journal of Finance 1(7):77-91

Moral-Escudero, R., Ruiz-Torrubiano, R., Suarez, A., 2006. Selection of optimal investment portfolios with cardinality constraints. In: Proceedings of the 2006 IEEE Congress on Evolutionary Computation, pp. 2382–2388

Neven, H., Denchev, V., Rose, G., Macready, W., 2012. QBoost: Training a Large Scale Classifier with the Quantum Adiabatic Algorithm. Journal of Machine Learning Research: Workshop and Conference Proceedings 25:333-348, 2012

Seki, Y., Nishimori, H., 2012. Quantum annealing with antiferromagnetic fluctuations. Physical Review E 85, 051112 (2012).

Wallace, D., 1994. A Probabilistic Specification-based Design Model: applications to search and environmental computer aided design. Massachusetts Institute of Technology. Dept. of Mechanical Engineering, http://hdl.handle.net/1721.1/11777

Woodside-Oriakhi M., Lucas C., Beasley J.E., 2011. Heuristic algorithms for the cardinality constrained efficient frontier. European Journal of Operational Research 213, 538–550

Zhu, H., Wang Y., Wang, K., Chen, Y., 2011. Particle Swarm Optimization (PSO) for the constrained portfolio optimization problem. Expert Systems with Applications, International Journal 38 (2011) 10161-10169