Financial Portfolio Management using D-Wave's Quantum Optimizer: The Case of Abu Dhabi Securities Exchange

Nada Elsokkary and Faisal Shah Khan
Quantum Computing Research Group
Department of Applied Mathematics and Sciences
Khalifa University, Abu Dhabi UAE

Davide La Torre
Department of Economics, Management, and
Quantitative Methods, University of Milan, Milan Italy
Department of Mathematics
Nazarbayev University, Astana Kazakhstan

Travis S. Humble
Quantum Computing Institute
Oak Ridge National Laboratory
Oak Ridge, Tennessee USA

Joel Gottlieb
D-Wave Systems, Inc.
British Columbia, Canada

1. Introduction

Financial portfolio selection is the problem of optimal allocation of a fixed budget to a collection of assets (commodities, bonds, securities etc.) which produces random returns over time. The word optimal, however, can have different meanings. For instance, one could define optimal so that a solution to the portfolio problem maximizes expected or average value of the return. While this definition is intuitively appealing, it is also naive because the values of a random process can deviate from the expected value. A definition of optimal should therefore incorporate deviations from the expected value. In his 1952 paper [4], Harry Markowitz proposed just such a definition of optimal for the financial portfolio selection problem. Markowitz proposed that the problem has an optimal solution when an investor maximizes expected return *and* minimizes variance in the values of the return. In other words, an investor should look to maximize gain and minimize risk for a given financial portfolio.

Markowitz's portfolio selection theory also formalizes the wisdom of portfolio diversification, where instead of allocating the largest fractions of the budget to a select number of assets with largest expected return, an investor allocates his budget to many assets with the hope that the fortune of some of these assets will not be influenced by the misfortune of some others.

Formally, consider a portfolio with n assets. Let a_i be the percentage allocation of the total budget to asset i and let R_i denote the random variable representing the return from asset i. Further, let R denote the random variable for the return from the entire portfolio. Then

$$R = \sum_i a_i R_i$$

whereby the expected return from the the portfolio is

$$E(R) = \sum_i a_i E(R_i) \tag{1}$$

due to the linearity of expectation. Next, consider the variance of R which is given by

$$\mathrm{Var}(R) = \sum_{i,j} a_i a_j \mathrm{Cov}(R_i, R_j), \tag{2}$$

where $\mathrm{Cov}(R_i, R_j)$ is the co-variance of the random variables R_i and R_j and which reduces to the variance of R_i for $i = j$. An investor working with respect to Markowitz theory will maximize (1) while minimizing (2). The information that the investor needs to achieve these two tasks is the pairs of statistics $[E(R_i), \mathrm{Cov}(R_i, R_j)]$. In his paper, Markowitz refers to these pairs as (E, V) combinations and assures that it is always possible to identify an optimal one.

More mathematically, this is a bi-criteria quadratic optimization problem [5]–[7] in which the set of all

978-1-5386-3472-1/17/$31.00 ©2017 IEEE

optimal solutions yield the so-called efficient frontier or efficient portfolios. Among all optimal portfolios, the decision maker will make a choice based on his/her preferences. In the literature, this formulation is also known as the mean-variance model.

Note that for a given pair of assets i and j, their contribution to $\mathrm{Var}(R)$ is zero when either a_i or a_j is zero, that is, when the investor does not allocate any budget to asset i or j. However, this contribution can be made smaller by allocating positive budget to assets i and j and ensuring that $\mathrm{Cov}(R_i, R_j)$ is negative. A negative value for $\mathrm{Cov}(R_i, R_j)$ will imply that the returns from asset i and j influence each other inversely and an adverse change in the returns from one asset will mean a favorable change in the returns from the other. Hence, not only does Markowitz theory justify the maxim of diversification mentioned earlier, but it also provides a measure for diversification.

2. Portfolio Selection as Unconstrained Optimization

In a relaxed form, the Markowitz portfolio model can be formulated as an unconstrained quadratic constrained binary optimization (QUBO) problem as follows [1]–[3]:

$$\begin{aligned} \text{Minimize :} \quad & \theta_1 \sum_i [-\alpha_i E(R_i)] \\ & + \theta_2 \sum_{i,j} \alpha_i \alpha_j \mathrm{Cov}(R_i, R_j) \\ & + \theta_3 \left(\sum_i \alpha_i A_i - B \right)^2 \end{aligned}$$

subject to $\alpha_i \in \{0, 1\}$, A_i is the maximum amount of money that can be invested in the i-th asset, and where B is the total budget. In this relaxed formulation we suppose that fractional shares of stocks can not be purchased, so the decision maker either invests the total amount A_i into the i-th asset or nothing.

The objective function is composed by three terms: the expected return, the volatility of the investment, and a penalization term that takes into account the difference between the invested amount of money $\sum_{i=1} A_i \alpha_i$ and the total budget B. The positive weights $\theta_i, i = 1, 2, 3$, describe the relative importance of each criterion in the decision making process.

The QUBO form for the Markowitz portfolio model is ideally suited for running on the quantum computer available from the Canadian technology company, D-Wave Systems. D-Wave Systems has developed a specialized quantum processor that uses the principles of quantum information processing to solve unconstrained optimization problems. The D-Wave processor uses superconducting technology and the physics of quantum annealing to find the minimum energy eigenstate of the above Ising Hamiltonian operator, which corresponds to the optimal strategy in the Markowitz model. Unlike a conventional computer, a quantum computer utilizes quantum bits, or *qubits*, to encode and process information. The principles of quantum physics allow qubits to store information in superposition of logical basis states. For example, a single qubit can store any normalized superposition of the logical 0 and logical 1 states simultaneously. A quantum computer manipulates superposition across multiple qubits to process information. Measurements with a quantum computer causes these superposition states to collapse into definite logical states, either 0 or 1, with probabilities given by the respective coefficients.

3. Classical Programming and Execution

For the purposes of this case study (classical and quantum), we considered 63 of the 68 securities listed on the Abu Dhabi Securities Exchange (ADX) website using the weekly closing value of each over a period of one year, from December 1, 2015 to November 30, 2016.

The weekly closing values of all the securities from ADX were formatted into a matrix where each column corresponded to an equity (indexed by the number assigned to it) and each row corresponded to the week of that year (50th-53rd week of 2015, and 2nd- 49th week of 2016). If there was no closing value listed in any given week, the value zero was assigned to it. This makes the results of the optimization problem somewhat unrealistic; however, this should not be an issue since the purpose is to optimize the data and compare results rather than actually use the results.

Moreover, the abundance of zeros does not make the data set too special from a calculation perspective since the matrix used in the actual problem is a co-variance matrix of the data, thus it will have much fewer zeros given that each security has at least one closing value in the specified interval.

3.1. The Data

From the data gathered, the expected values and co-variance matrix were calculated using the columns as the random variables, 63 in total. A random vector of length 63 was generated and used so that each element corresponded to the individual budget allocated to an asset. If an equity is recommended by the algorithm, then its individual budget will be the amount invested into it. The total budget was set to 1000. This constrained the sum of the individual budgets of the equities recommended to be less than or equal to 1000. The $theta$ values in the model were set to $\frac{1}{3}$ each so that all components of the model were equally weighted.

3.2. MATLAB Execution

The co-variance matrix and expected values were read into the mathematical software MATLAB together with the budgets and the θ values. An anonymous function f was defined to be minimized by the built-in genetic algorithm function:

$$\begin{aligned} f = &@(a)((-\theta_1) \cdot a \cdot (Expectation)^T \\ &+ \theta_2 \cdot a \cdot Covariance \\ &\cdot a^T + \theta_3 \cdot ((aB_i) - TotalBudget)^2); \end{aligned} \tag{3}$$

The genetic algorithm was also constrained to find only integer solutions on the interval $[0, 1]^{\times}63$ in order to make the solution binary. The solution would be in the form of a binary vector of length 63 in which an entry of 1 represented the equities that are recommended.

However, since the genetic algorithm is not an exact method, the solutions vary from one execution to the next. Therefore, the function was executed n times (for a suitable n) and the solutions, function values, and execution time of each trial were recorded. The number of times each equity was part of the solution was recorded so that it was clear which equities were recommended the most number of executions.

Finally, the maximum number of most persistent equities were chosen, so long as they do not violate the total budget constraint. The MATLAB code and data files (in MS Excel format) can be found online [14]. We also point out that the MATLAB solution has been compared to a solution obtained using qbsolv, a heuristic QUBO solver from D-Wave. The next step is to validate and compare the solutions obtained from MATLAB with the quantum computer.

4. Quantum processing

We now show how quantum superposition states within the special purpose D-Wave processor may be used to effectively sample the space of potential portfolios. Computation then corresponds to finding the optimal portfolio as defined by the objective function above. The D-Wave processor uses quantum dynamics, namely quantum annealing, to isolate the portfolio selection, and we develop an implementation of portfolio selection using the D-Wave processor to recover the optimal portfolio. Our approach tests whether the adoption of the D-Wave quantum computer allows for a meaningful increment in computational performance for solving the Markowitz portfolio. We use D-Wave's quantum optimizer to find the optimal allocation of funds.

We map the QUBO form of Markowitz's portfolio selection problem into the well-known Ising model.

$$H = -\sum_j h_j Z_j - \sum_{i,j} J_{i,j} Z_i Z_j + \gamma \tag{4}$$

where the real-valued coefficients h_i and $J_{i,j}$ define, respectively, the bias for qubit i and the coupling between qubits i and j and the Pauli Z_i operator is for the i-th qubit.

$$J_{i,j} = \frac{-1}{4} \left(\theta_2 \mathrm{Cov}(R_i, R_j) + \theta_3 A_i A_j\right) \tag{5}$$

$$\begin{aligned} h_i = &\frac{-1}{2} \big(\theta_2 \mathrm{Cov}(R_i, R_i) + \theta_3 A_i^2 \\ &\quad -\theta_1 E(R_i) - 2B\theta_3 A_i\big) \end{aligned} \tag{6}$$

and $\gamma = \theta_3 B^2$.

4.1. Quantum Programming and Execution

Programming the D-Wave processor requires first reducing the Markowitz model to the Ising form in Eq. (4) [8]. This logical representation of the optimization function must then be encoded into the processor hardware. The D-Wave processor has a unique hardware connectivity graph called Chimera that corresponds to a rectangular array of bipartite unit cells. As shown in Eq. (5), the i-th and j-th qubits generally posses non-zero coupling between them. Mapping logical problems into the Chimera hardware therefore requires an embedding function that uses chains of coupled qubits to develop equivalent representations of the problem structure. There are a variety of embedding methods available [10]. The D-Wave Solver API (SAPI) provides an implementation of an embedding algorithm search from Cai, Maccready, and Roy [11], which uses a probabilistic search to find a valid embedding. We use the SAPI-provided embedding routine provided to program the dense Ising model into the Chimera hardware.

The resulting parameters for the embedded physical Ising model are then given in terms of the original Ising coefficients and an intra-chain coupling constant J [9]. We set the value of J to be approximately an order of magnitude greater than the largest value of an coefficient; this setting is optimized based on empirical observation.

Execution of the embedded physical program is based on the time-dependent Hamiltonian $H(t) = A(t)H_0 + B(t)H_1$, where H_0 is the transverse field term and H_1 is the physical Ising model. The annealing schedules $A(t)$ and $B(t)$ have a fixed form within the hardware but a scalable duration T. Larger values of T imply a slower schedule. We use an annealing time $T = 20\mu s$. A single program execution returns a binary string representing the candidate solution for the ground state of the physical Ising model. We decode this string into the solution for the QUBO problem. We collect an ensemble of such strings and generate a list of these candidate solutions. We then order the list of these solutions based on the calculated energy and pick the solution with the lowest energy. Our implementation of these programming steps rely on the XACC programming framework [12]. The XACC framework was designed to provide an extensible software framework for integrating novel computational accelerators, like quantum processors, into conventional scientific application codes.

Our implementation of portfolio optimization uses the D-Wave processor to find the optimal selection based on real data from the Abu Dhabi Securities Exchange. The prototype open-source code and data set are available online [13]. As seen in the data files for the averages and co-variances, there are a wide array of values for the selected stocks. As a simplification we have performed portfolio optimization using a current price A_i for stock S_i that matches the corresponding average $\mathrm{E}(S_i)$. We have also assumed a total budget for investment to be \$100. Finally, we have found the solution to this multi-objective optimization function to depend on the specific choices for θ_i. This is a well-known feature of using the weighted sum method to represent multi-objective optimization. For our demonstrations, we have settled on the use of $\theta_i = 1/3$ for all i.

For the above definition, we have used adiabatic quantum optimization with the D-Wave processor to search for the optimal portfolio. Repeated runs of the program for the same anneal time of 20 μs returns the same result of a portfolio selection that costs \$121.176. This cost does not strictly match our budget because the weight sum method provides only a trade-off between the different constraints. For example, changing the weights to $\theta_1 = 0.8$, $\theta_2 = 0.$, and $\theta_3 = 0.2$ produces a portfolio that costs \$119.007. However, this choice has clearly ignored the influence of the co-variance on minimizing risk. In addition, longer anneal times (up to 2 ms) yield very similar portfolios but with slightly lower costs. We find that some selection cannot be unambiguously determined from the program. This traces back to the embedded chains that return a strictly even distribution of the 0 and 1 outcomes. We separate these stocks from the remainder of the portfolio and compute this uncertain costs separately. For our problem set, this spread is typically from \$4 to \$40.

This data set is too large for direct embedding onto the D-Wave 128-qubit simulator, and was submitted to qbsolve. qbsolv divides QUBO problems into chunks and iterates on sub-QUBOs until either the best solution is found, or the input time limit runs out. The qbsolv solution is compared to the MATLAB-derived solution, with good agreement.

In summary, we have performed Markowitz portfolio optimization using the D-Wave processor.

5. Acknowledgments

This manuscript has been authored by UT-Battelle, LLC, under Contract No. DE-AC0500OR22725 with the U.S. Department of Energy. The United States Government retains and the publisher, by accepting the article for publication, acknowledges that the United States Government retains a non-exclusive, paid-up, irrevocable, world-wide license to publish or reproduce the published form of this manuscript, or allow others to do so, for the United States Government purposes. The Department of Energy will provide public access to these results of federally sponsored research in accordance with the DOE Public Access Plan.

References

[1] B. Aouni, Multi-Attribute Portfolio Selection: New Perspectives, Information Systems and Operational Research, 47 (2009), no. 1, 1-4.

[2] B. Aouni, C. Colapinto, D. La Torre, Financial Portfolio Management through the Goal Programming Model: Current State-of-the-Art, European Journal of Operational Research, 234 (2014), no. 2, 536-545.

[3] F. Ben Abdelaziz, B. Aouni, R. El Fayedh, Multi-objective stochastic programming for portfolio selection, European Journal of Operational Research, 177, (2007), no. 3, 18111823.

[4] H. Markowitz, Portfolio Selection, The Journal of Finance 7 (1952), 77- 91.

[5] Y. Q, R. E. Steuer, M. Wimmer, An Analytical Derivation of the Efficient Surface in Portfolio Selection with Three Criteria, Annals of Operations Research, (2016), DOI: 10.1007/s10479-015-1900-y.

[6] R.E. Steuer, Y. Qi, M. Hirschberger, Portfolio Selection in the Presence of Multiple Criteria, Handbook of Financial Engineering, Springer Science, New York, 3-24, 2008.

[7] C. Zopounidis, M. Doumpos, Multi-Criteria Decision Aid in Financial Decision Making: Methodologies and Literature Review, Journal of Multicriteria Decision Analysis, 11 (2002), no. 4-5, 167-186.

[8] T. S. Humble, et al. "An integrated programming and development environment for adiabatic quantum optimization," Computational Science & Discovery 7, 015006 (2014).

[9] K. Pudenz, Parameter Setting for Quantum Annealers, In IEEE High Performance Extreme Computing Conference, 1-6, 2016.

[10] T. D. Goodrich, B. D. Sullivan, and T. S. Humble. "Optimizing Adiabatic Quantum Program Compilation using a Graph-Theoretic Framework," arXiv:1704.01996 (2017).

[11] J. Cai, W. G. Macready, and A. Roy. A Practical Heuristic for Finding Graph Minors, arXiv:1406.2741, (2014).

[12] XACC Programming Framework, https://ornl-qci.github.io/xacc (2017).

[13] Markowitz Application Code, https://code.ornl.gov/qci/markowitz (2017).

[14] Markowitz Application Code, https://www.researchgate.net/project/Financial-Portfolio-Management-using-D-Wave-Quantum-Computer-The-Case-of-Abu-Dhabi-Stock-Exchange (2017).