# Continuation/Research Progression Projects Form (7)

**Required for projects that are a continuation/progression in the same field of study as a previous project.**
***This form must be accompanied by the previous year's abstract and Research Plan/Project Summary.***

Student's Name(s) Abishek Ravindran

**To be completed by Student Researcher:** List all components of the current project that make it new and different from previous research The information must be on the form; use an additional form for previous year and earlier projects.

| Components | Current Research Project | Previous Research Project: Year: 2018-19 |
|---|---|---|
| 1. Title | Implementation of Novel Sector Weight and Google Trends Data Objectives using MOEA/D Curtails Systematic Risk for Quintessential Investors | Implementation of Simulated Annealing for Cardinality Constrained Portfolio Optimization with Sector Weight Constraints |
| 2. Change in goal/ purpose/objective | Portfolio optimization models fail to take into account risks associated with market failures and human behavior and are therefore tailored towards institutional investors. More broadly, this project proposed an extended portfolio optimization model that sought to account for Google Trends data and sector weights as objectives instead of constraints to bolster the applicability of portfolio models for the quintessential investor. | Portfolio optimization models fail to take into account sector weights, opening investors up to a great deal of systematic risk. Sector weights were implemented as a constraint within this model to reduce systematic risk for the retail investor. |
| 3. Changes in methodology | MOEA-D (a type of evolutionary algorithm), instead of simulated annealing, was implemented; I was required to code a completely new algorithm. In addition, this algorithm consisted of a population of 230-300 solutions. Moreover, this algorithm optimized for three objectives. Testing was completed on an in-sample and out-of-sample data set, for parameter fine-tuning and portfolio performance tests. Instead of the 2000 tests completed for cost function and penalty term values last year, a total of 51000 tests were completed this year. | A single objective form of simulated annealing was implemented, only minimizing the variance of the portfolio. The algorithm only calculated a single solution, and optimized only a single objective. Testing was completed on only an in-sample-data set for the penalty term values and cost function values for 1000 tests each. |
| 4. Variable studied | There were two sets of variables tested on the evolutionary algorithm developed. The first set, for parameter fine-tuning, included a set of six variables: portfolio return, variance, sector-weighting, fitness value, and population size, and adjusted fitness value on the in-sample data-set. On the out-of-sample data set, the future return, variance, and Sharpe ratios in addition to the aforementioned set were tested. | Only parameter fine-tuning was completed. The only two variables tested on the simulated annealing algorithm for this parameter fine-tuning were penalty values and cost function values. |
| 5. Additional changes | | |

Attached are:

☑ Abstract and Research Plan/Project Summary, Year 2018-19

I hereby certify that the above information is correct and that the current year Abstract & Certification and project display board properly reflect work done only in the current year.

Abishek Ravindran — Student's Printed Name(s)

Signature

01/25/19 — Date of Signature (mm/dd/yy)

# Implementation of Simulated Annealing for Cardinality Constrained Portfolio Optimization with Sector Weight Constraints

Abishek Ravindran

Abstract

Overallocation into sectors within a portfolio is a problem of paramount significance that constitutes risk for both the investor and systemic risk for the broader economy. The Markowitz model seeks to minimize risk for a specific level of return. While the model itself is simple, additional constraints must be added to replicate real-world conditions. When such restrictions are added, delineating a single solution becomes impractical, making heuristics a more suitable choice for application. Heuristics are practical methods that provide an adequate solution with imperfect information. Simulated annealing was employed in this study, a technique that simulates the process of cooling down metals to reduce disconformities and is widely known for its ability to accept worse solutions to more thoroughly search a solution space and avoid local minima or maxima. All code was written using C++, and all stock data was compiled using NASDAQ databases. To optimize the algorithm, initial parameters and constraints were tested through experimental analysis to ensure that the algorithm was offering optimal results. In addition, several weights were tested for the sector floor/ceiling constraint to determine the best form of the constraints. The code ran to completion, requiring a total of 10,000 iterations over 20 trials along with an average mean and variance for the cost function of 86009.5 and .000017, respectively. Future research will explore means of further optimizing the algorithm and the efficacy of other metaheuristics on the algorithm.

A. Background

While the objective for any computational problem is to locate the single optimal solution, problem complexity and incomplete information can make the delineation of a single solution infeasible. In these cases, heuristics and metaheuristic methods can be utilized to search the solution space and locate near-optimal solutions that are considered to be satisfactory much more efficiently. Heuristics are known to be problem-dependent methods that serve to locate a solution to a specific type of problem, while metaheuristics are algorithmic frameworks for search methods that can be implemented for a wide variety of problems (Moshref, 2018). A common metaheuristic that is applied to a variety of problems is stimulated annealing.

Stimulated annealing is derived from annealing, a metallurgical refinement method that involves the gradual cooling of material in order to decrease disconformities while increasing crystal size. Stimulated annealing looks to replicate this tangible process through the application of a probabilistic function that gradually decreases the probability of accepting nonoptimal solutions (Aarts, 2015). While accepting a solution that is considered worse may seem unideal, this is integral to the success of stimulated annealing. Functions that only seek better solutions, also known as greedy functions, tend to converge into local minimums or maximums. The acceptance of poor solutions is critical to complete a thorough search of the solution space. The iterative process of a stimulated annealing algorithm.

Stimulated annealing iterates through a three-step process in order to reach a final solution. The algorithm begins by generating a random solution that is to be compared to the current solution. This solution is to be near the current solution based on a predetermined or pseudorandom step in an effort to ensure that movement within the search space is completed in a structured manner while maintaining the probing nature of the algorithm (Aarts, 2015). The manner in which the new solution is generated is largely dependent on the constraints of the problem. When a problem requires adherence to strict constraints, the algorithm may only search the feasible solution space. In other instances, the unfeasible solution can be generated and have a penalty added (Chang, 2000). While allowing for unfeasible solutions may seem counterintuitive, these solutions ensure that the solution space is not excessively limited in scope and allow the algorithm to search a wider solution space, while the penalty ensures that these solutions are not ultimately accepted (Crama, 2003).

After generating a random solution, the algorithm determines whether or not to move to the new solution based on a designated probability. As previously stated, a key feature of stimulated annealing is the ability to accept a less ideal solution. A solution that has a greater proximity to a global minimum or maximum will be accepted by the algorithm with a probability of 1 (Baird, 2015). However, in the case of a less ideal solution, the algorithm will accept the solution through the use of a probabilistic function, based on the difference between the current and new solution, as well as the temperature.

The acceptance probability is directly proportional to the temperature and inversely proportional to the difference between the two solutions, where x is the original solution, is the new solution, and T is the simulated temperature. At high temperatures, the function is more willing to accept worse solutions in order to more thoroughly search the solution space. In later iterations, when the temperature is lower, the algorithm seeks a more precise solution and is therefore less likely to accept the new solution. In cases when the algorithm reaches a local minimum or maximum, the algorithm is either able to take a drastic step or move to a previous solution that has been stored if the algorithm is unable to diverge from the local minima (Chang, 2000).

Financial problems, such as the portfolio optimization problem or index-tracking problem, can be modeled with rather simple functions that can be solved through the use of basic linear and quadratic programming. However, these simple functions lack real-world constraints and consequently can produce solutions that would be considered infeasible in the real world (Sant'Anna,, 2017). Once such constraints are added to the model, a certain degree of complexity is created that requires the use of a metaheuristic. What should be noted is that, in each of the preceding problems, an exact solution is impossible to delineate, as historical data is not precisely indicative of future performance (Bodie, 2018).

An inherent tradeoff in investing is seen between risk and return. In order to receive a risk-free return, investors must be willing to accept a lower rate of return. In order to receive greater return, investors would be subject to greater systematic risk that cannot be hedged against. Given a fixed level of risk or return, the Markowitz Modern Portfolio Theory (MPT) can be solved to determine the maximum level of return or minimum level of risk, respectively (Markowitz, 1952). While an expected level of risk can be difficult to quantify, an expected level of return can be set by the investor based on their goals. For a given level of return, the function can be written as follows:

While the objective function is in itself rather simple, and can be solved through more precise methods such as quadratic programming, it is not representative of a real-world scenario. In order to make the model and portfolios representative of and applicable to the real-world, additional constraints must be added to the model. Such constraints are seen in the following table.

The budget constraint ensures that all capital is invested at all times. Within a portfolio, an investor wants all capital invested to accrue returns rather than being uninvested and being subject to inflation (Bodie, 2018). The cardinality constraint limits the number of assets that can be invested in, as it is difficult to manage a portfolio with an excessive number of assets. Moreover, the benefit that is garnered from diversification has diminishing marginal returns and is not present after a large number of assets have been purchased. The asset weight constraints ensure that a certain asset is not overweighed in the portfolio as a select asset is not subject to the volatility of the overall portfolio. The floor constraint is simply a measure to ensure that the portfolio does not have any short selling, or negative positions.

The sector weight constraints ensure that specific sectors within the portfolio are not overweighed. Market failures such as the tech bubble of 2001 and the financial crash of 2008 portray the need for sector diversification, as investors lost large sums of capital when portfolios were overweighed in sectors that were expected to see large future growth. However, these sectors saw large capital losses over short periods of time. The causes of these major downturns are difficult to pinpoint, but can be attributed to a series of factors. While an inherent factor is the careless financing activity that caused the initial problems within these companies, the implications of this speculative behavior is magnified by investors that began to sell of their interests in the affected sector to avoid any further market downturns (Bodie, 2018). To track the performance of these sectors over the respective market failures, SPDR ETFs were utilized for the analysis. Over the downturns of the market, the technology and financial sectors saw losses of 69.75% and 84.48%, respectively (NASDAQ, 2018). What should be noted is that these downturns extended into the greater market, portraying the need for diversified portfolios to avoid such market failures.

B. Methodology

Data for this experiment will be compiled for the markets for a ten-year period: from August 1st 2008 to August 1st 2018. The data set will consist of the daily returns for each constituent asset of the S&P 100 and will be compiled from the NASDAQ American Stock Exchange. A series of 8 stocks will be omitted from the data set for a lack of data (the assets became constituents of the S&P 100 in the past ten years) or due to a lack of stocks within the sector. Several sectors are insubstantial within the S&P and will therefore have no significant impact on the results. Moreover, they will greatly reduce the search space as the sector ceiling constraint will only allow for very small weights within these sectors. These sectors were real estate (.04 %), materials (.09%), and utilities (1.6%) (S&P, 2018).

The daily opening and closing prices will be recorded from the database, and log returns will be calculated using Excel. Where vf is the final value (closing price) and vi is the initial value (opening price).

Log returns pose several advantages in analysis. Being that the investment is assumed to be continuously compounding, it better approximates the real movement present within the markets. Moreover, the analysis is time additive when using log returns and is better able to approximate returns when using a large data set with short time intervals (University of Bocconi, 2015).

Initial data files will be created based on the log returns for the covariance matrix, average returns, and upper/lower bounds for both asset and sector weights. The covariance matrix and average return files will be created using the cov. and mean. functions in R, respectively. The upper and lower bounds will be recorded into Excel and read into txt. files also via R.

All implementation code will be written in C++ for several reasons. C++ provides a great amount of the object-oriented functionality of advanced languages, while providing the simplicity of more elementary languages that will be integral to the implementation and efficient computation of such large data sets. Vectors will be used heavily for the implementation of data sets and their performance in iterative processes.

In creating the stimulated annealing algorithm to solve the Markowitz problem, an initial scaffold for the code will be created to compile the necessary resources and methods. A wide array of methods is necessary, including computational, analytical, and supplementary move generation methods. While many of these methods can be written as components of the main method, the code will be simplified greatly through the use of accessory methods to aid in the algorithim.

The computational methods will be created for mathematical procedures that will be required within various aspects of the final code. These methods are anticipated to have simpler implementation code that will consist of a call for input values as parameters followed by calculations. The matrix-vector product and dot product methods will be structured largely based on the dgemv and ddot methods found in Linear Algebra PACKage (LAPACK). Since the original methods were written in Fortran 90, the code can be simplified greatly through the use of C++.

The analysis methods will be used to alter the financial data from the S&P 100 into more meaningful values, such as variance and returns. These methods will aid in creating a consistent data set that is crucial to the use of iterative processes when searching the solution space and will allow for simpler calculations when determining the initial structure of the portfolio and changes within the portfolio. These methods are similar to the computational methods in simplicity, taking in the data set or values as a parameter and completing the respective procedure to produce the necessary output.

The move generation methods will work in conjunction to aid in the iterative process of the algorithm to search the solution space. The choose case and step calculation methods are both created to ensure the code meets the constraints (cardinality and asset/sector weight constraints, respectively). The output of these methods will be utilized within the move generation method that would move a group of three assets randomly based on the return of the three assets using the following equations, based on a ball construction.

Finally, the penalty methods will ensure that any solutions that fail to meet the weight floor/ceiling constraints will be determined unacceptable by the cost function by adding an arbitrarily large value.

In order to test and debug the code, simple data sets of 3-5 assets will be used in problems that could be solved linearly for an exact solution to ensure that the code is producing relatively accurate results. While heuristic methods do not need to produce exact solutions, the solutions should provide results that are considered accurate and approximate the global optimum, especially for a small solution space. Many of the initial errors found within the code are anticipated to be derived from the move generation methods that are more difficult to track through the iterative process. For this, breaks can be placed into the code to locate logic errors.

After logic errors are removed from the code, parameters will be fine-tuned through the use of experimental analysis. Several of the parameters for the algorithm, including the initial temperature, decay rate, maximum number of cooldown iterations, and frozen count (maximum number of iterations the algorithm can remain at a single value), can be altered and greatly affect both the accuracy of the algorithm at reaching the global maximum and the efficiency in the convergence time of the algorithm. In order to test these parameters, 20 to 30 experiments will be run with different values for such parameters to determine the ideal set. What should be noted is that every possible value for each parameter cannot be tested, but better values can be gathered through the use of such an algorithm.

Although the asset weights were set at an arbitrary .3 (to ensure that the algorithm did not allocate more than 30 percent of the capital towards a single asset), the sector weights must be adjusted based on the size of the asset class. Several approaches will be attempted for the sector constraints. A market-capitalization approach will be taken, where the assets are allocated based on the weight of the sectors within the actual index. A cardinality approach is also to be tested, where the limits are allocated proportionally to the number of assets in the sector. Finally, a uniform approach is to be tested, where the weight limit for each sector is the same. However, this is not anticipated to perform well, as the size and number of companies in different sectors vary greatly.

References


Aarts, E., Korst, J., & Michiels, W. (2015). Chapter 7: Stimulated Annealing. Retrieved from https://pdfs.semanticscholar.org/4569/26ae7135a4450859197506e87e75451dd6e1.pdf.

Bodie, Z., Kane, A., & Marcus, A. J. (2018). Investments. New York, NY: McGraw-Hill Education.

Chang, T., Meade, N., Beasley, J., & Sharaiha, Y. (2000). Heuristics for cardinality constrained portfolio optimisation. Computers & Operations Research, 27(13), 1271-1302. doi:10.1016/s0305-0548(99)00074-x.

Crama, Y., & Schyns, M. (2003). Simulated annealing for complex portfolio selection problems. European Journal of Operational Research, 150(3), 546-571. doi:10.1016/s0377-2217(02)07841

Cruz, R. M., M., H., & M., R. (2011). Artificial Neural Networks and Efficient Optimization Techniques for Applications in Engineering. Artificial Neural Networks - Methodological Advances and Biomedical Applications. doi:10.5772/15293

Efficient frontier. (2018, August 22). Retrieved from https://en.wikipedia.org/wiki/Efficient_frontier

Glazer, P. (2018, April 14). Modern Portfolio Theory, and Portfolio Construction. Retrieved from https://hackernoon.com/an-exploration-of-cryptocurrency-investing-modern-portfolio-theory-and-portfolio-construction-22531c292ed9?gi=f6ed81061cf1

Kempthorne, Peter. “Lecture 14: Portfolio Theory.” Https://Ocw.mit.edu/Courses/Mathematics/18-s096-Topics-in-Mathematics-with-Applications-in-Finance-Fall-2013/Lecture-Notes/MIT18_S096F13_lecnote14.Pdf, Massachusetts Institute of Technology, 2013, ocw.mit.edu/courses/mathematics/.

Mansory, A., Ebrahimi, N., & Ramazani, M. (2014). Ranking of Companies based on TOPSIS-DEA Approach Methods (Case Study of Cement Industry in Tehran Stock Exchange). Pakistan Journal of Statistics and Operation Research, 10(2), 189. doi:10.18187/pjsor.v10i2.663

Markowitz, H. (1952). Portfolio Selection*. The Journal of Finance, 7(1), 77-91. doi:10.1111/j.1540-6261.1952.tb01525.x.

Martí, R., & Reinelt, G. (2011). The linear ordering problem: Exact and heuristic methods in combinatorial optimization. Heidelberg: Springer.

Moshref, J. (2018). Retrieved from http://www.mit.edu/~moshref/Heuristics.html

NASDAQ Stock Quotes and Stock Exchange News. (2018). Retrieved from https://www.nasdaq.com/

S&P Dow Jones Indices. (2018). Retrieved from https://us.spindices.com/indices/equity/sp-100

Sant'Anna, L. R., Filomena, T. P., & Caldeira, J. F. (2017). Index tracking and enhanced indexing using cointegration and correlation with endogenous portfolio selection. The Quarterly Review of Economics and Finance, 65, 146-157. doi:10.1016/j.qref.2016.08.008

Soler-Dominguez, A., Juan, A. A., & Kizys, R. (2017). A Survey on Financial Applications of Metaheuristics. ACM Computing Surveys (CSUR), 50(1).

University of Bocconi. Chapter 2: Financial Returns. (2015). Retrieved from http://didattica.unibocconi.it/mypage/dwload.php?nomefile=ch2_201620160825122429.pdf