Two-Stage Stock Portfolio Construction: Correlation Clustering and Genetic Optimization

Sachin Joglekar
BITS Pilani - K. K. Birla Goa Campus
srjoglekar246@gmail.com

Abstract

Ideal portfolio creation has been the focus of considerable machine learning research in the domain of finance. In this paper, the development of a two-stage platform for generating stable stock-based portfolios is explored. The first stage involves clustering of stocks based on time-weighted correlations, using a modified version of the K-Means++ algorithm. This clustering helps in the quantification of portfolio diversification at a later stage. In the second step, a genetic paradigm is employed to optimize the returns of the portfolio in such a way as to ensure its diversification at the same time. This leads to the formation of a portfolio that shows a high and stable Markowitz ratio of returns/risk. The experimental results support the central hypothesis, and hint at possible commercial applications.

1 Introduction

The problem of optimal portfolio construction is one of the oldest in modern financial theory. Though a portfolio in its real sense may consist of various securities such as stocks, bonds and savings accounts, this paper only focuses on the most volatile of the lot - stocks. The first major step towards mathematical portfolio optimization was taken by Markowitz in his mean-variance portfolio theory (Markowitz 1952). In this paper, I base the notion of a portfolio's fitness mainly on Markowitz's theory, with influences from some ideas developed after his work.

The two major factors usually linked to a portfolio's fitness are the expected returns and associated risk. The expected returns measure the profits that a portfolio is expected to yield in a given time period. The risk, on the other hand, is a quantitative indicator of how volatile a portfolio is. Though returns are the obvious motivation behind constructing a portfolio, risk plays a major role in the confidence an investor is likely to have in it. Investors are usually more likely to prefer a stable portfolio with lesser returns than a very volatile one with high results. Hence, I maintain the focus of this paper on the creation of stable, low-risk portfolio that ensures decent returns with minimal volatility.

Reduction of risk is strongly associated with portfolio diversification. Mathematically, this corresponds to investment in securities (stocks, in this case) which show low correlations in their time series. This would ensure that negative trends in the prices of one stock would likely be made up for by positive trends in that of another - unless the market itself is crashing. To ensure that stocks with high correlations do not make up the entire portfolio, they are usually clustered in such a way that intra-cluster correlations are high, while inter-cluster correlations are low. This way, the investor can guard himself against under-diversification. Correlation-based clustering is performed in the first of the two stages of the proposed framework, to help measure a portfolio's diversification in the second stage. Since the clusters are formed such that inter-cluster correlations are low, the portfolio is guaranteed to be diversified if it contains stocks from various clusters.

To optimize the overall output with respect to returns as well as risk, we employ a genetic approach. Genetic Algorithms are stochastic optimization/search techniques based on the principles of evolution and natural selection. However, while most search algorithms attempt to search for an object of some sort, genetic algorithms search for optimal solutions in the solution space. GAs (Genetic Algorithms) start off with an initial pool of randomly generated candidate solutions, and then attempt to build fitter and fitter generations based on a pre-defined fitness function and the genetic operators of selection, crossover and mutation. Their strength lies in their ability to deal with non-continuous or non-smooth objective functions effectively - essentially the kind of objective involved with portfolio optimization.

I employ a GA to construct an optimized portfolio from the available range of stocks. Focus is maintained on ensuring that strongly correlated stocks don't make up the entire output, while at the same time improving the expected returns. The experimentation conducted by me using stocks from the Dow Jones Industrial Average [1] (DJI) index shows promising results consolidating my belief in the effectiveness of the proposed two-stage approach.

The rest of the paper is divided as follows- Section 2 talks about related work in this domain; Section 3 describes the portfolio generation framework developed; Section 4 presents some experimental evidence supporting the hypothesis and Section 5 draws conclusions and presents points

[1] https://www.djindexes.com

of discussion regarding future work and possible improvements to the work.

2 Related Work

The importance of incorporating correlations in portfolio optimization was first stressed by Markowitz in the Capital Asset Pricing Model (Markowitz 1959). Correlation factors also play a vital role in the usage of more recent techniques in finance, such as the Arbitrage Pricing Theory (John Campbell 1997) and Value at Risk (Paul Embrechts 1999). Most of the work on diversification using correlations attempts to classify or cluster strongly correlated stocks together, so as to simplify the process of constructing the 'right' portfolio. There are a variety of ways researchers define a $good$ or $optimal$ portfolio in literature. While some focus on high returns, others focus on diversification - the definition used depends on the objective behind constructing the portfolio.

Covariance is another statistical measure of the strength of relationships between stock price trends. This factor has been exploited in studies that try to calculate how underdiversified portfolios are on an average, such as (William N. Goetzmann 2008). Correlation can be thought of as a normalized form of covariance, that does not depend on the magnitudes of prices themselves. It has also been used to cluster stocks via graph-based clustering techniques to explain relationships between stocks of different sectors and markets (Rosén 2006). Usually, it is found that stocks of companies belonging to same industry sector and/or market do show strong correlations in their time series. However, it is not uncommon for stocks that are seemingly unrelated to show very similar trends in their time series

Genetic algorithms as a tool of optimization have started receiving considerable attention in the field of finance, in recent years (T. Vallée 2003). (Yang 2006) was one of the first works to suggest their application to generation of efficient and robust portfolios. The fitness function usually used to measure the 'fitness' of a portfolio is the Markowitz return/risk ratio (Chi-Ming Lin 2007) (Pankaj Sinha 2013). (Pankaj Sinha 2013) also advocates usage of a higher mutation rate (0.4) than the one usually used with genetic algorithms (0.01). This may stem from the need to explore unseen regions in the solution space in an attempt to arrive at the global optimum for the objective function.

(Rosén 2006) follows a promising two-step approach, like this paper, in an attempt to optimize stock portfolios. They first use the K-Means method to classify actions of a portfolio into classes so that their expected returns and VaR (Value at Risk) values are close to each other. Then, a dynamic optimization algorithm called Min-VaRMaxVaL is used to build a portfolio that has the highest average returns and lowest average VaR, as obtained by the classification.

3 Portfolio Generation Framework

An ideal portfolio, as described in financial theory, has two important characteristics- i) high returns and ii) low risk. In this section, we describe a two-stage portfolio generation procedure that ensures optimization of both the aforementioned factors. Correlation-based clustering is first performed on the available range of stocks to group the strongly correlated ones together. This helps in quantifying the diversification of candidate portfolios at a later stage. A genetic algorithm is then run in the second stage of the framework, with the aim of optimizing the output portfolio's fitness. The fitness function is defined in such a way so as to encompass sufficient returns as well as diversification, to reduce associated risk. The inputs to the algorithm involve the time series data for the list of stocks to be processed (usually 2 to 3 years worth) and the number of slots available in the target portfolio.

3.1 Correlation-based stock clustering

This section outlines the methodology followed for stock clustering based on time series correlations.

Time-weighted correlation The most common measure of correlation is the Pearson coefficient ρ, which ranges from -1 to 1 (-1 if negatively correlated, 1 if positively correlated, and 0 for no correlation at all). The formula for ρ of two time series X and Y can be defined as

$$\rho(X,Y) = \frac{cov(X,Y)}{\sigma_X \sigma_Y} \tag{1}$$

where $cov(X,Y)$ denotes the covariance of X and Y, and σ_X denotes the standard deviation of X.

Usually, the Pearson coefficient as obtained by applying (1) over two time series is used to denote the relationship between their trends. However, it is observed that the correlation between stock price trends usually changes over time- two stocks that show a strong correlation today may not have such a significant relationship sometime in the future. This may be attributed to 'global' changes that affect one stock differently than the other, or company-specific changes that affect only one particular stock. Therefore, this paper defines the concept of time-weighted correlation, which puts more emphasis on recent/current trends, as compared to older ones. The time-weighted Pearson correlation coefficient, ρ_t, for two stock series can be given by-

$$\rho_t(X,Y) = (\frac{1-r}{1-r^N}) \sum_{i=0}^{N} r^{i-1} \frac{(x_{-i} - \mu_X)(y_{-i} - \mu_Y)}{\sigma_X \sigma_Y} \tag{2}$$

where σ_X and x_{-i} denote standard deviation and the ith latest observation in X respectively; N is the total number of observations in the time series; and r is the decay constant, equal to a real number less than 1. r signifies the weightage given to the relationship between any two simultaneous observations in X and Y, as a function of their age with respect to the latest observations in the time series. In my implementation, I define $r = 0.998$ since this value gives the best results (determined experimentally).

ρ_t, like ρ, varies from -1 to 1. The only difference is that ρ_t gives more importance to the movements shown by newer data, as compared to older observations. This unique definition of 'correlation' enables one to draw a more accurate pic-

ture of the expected relationship between two stocks in the future under consideration. Moreover, with this definition, its not necessary that two companies of the same industry will show a high ρ_t value.

Modified K-Means++ clustering For the clustering step, the time series vectors , viz. the vectors containing historical Adjusted Closing prices of concerned stocks, are first normalized by standard deviation. This enables the algorithm to focus on the movements shown by individual stocks, and define 'correlation centroids' based on the relationships between them, in the Expectation-Maximization iterations.

K-Means++, as described by David Arthur and Sergei Vassilvitskii is a popular enhancement over the traditional K-Means clustering algorithm (David Arthur 2007). This technique suggests a unique way to generate the initial centroids for traditional K-Means, in an effort to avoid reaching local optima and quickening the process of convergence. Once the initial centroids are generated, the usual K-Means way of clustering is followed. The distance measure normally used for clustering by this technique is the Euclidean distance. However, since we aim to cluster stocks according to time-weighted correlations between their time-series, we use a distance measure as defined by $(1 - \rho_t(X, Y))$, where ρ_t is obtained from (2). The number of clusters, as input to the algorithm, is set to be equal to the number of portfolio slots as defined by the user.

The K-Means++ algorithm for correlation clustering, with k clusters, can be shown in steps as-

- A data point is chosen at random and added to C, the set of initial centroids.
- For each data point x, compute $D_{min}(x)$ given by

$$D_{min}(x) = argmin_{c \,\epsilon\, C}(1 - \rho_t(t_c, t_x)) \tag{3}$$

  where t_x denotes the normalized time series of data point (stock) x.
- Choose a data point randomly from the set, with each point x being assigned a weightage in the selection, proportional to $D_{min}(x)$.
- Repeat steps II and III until k centroids have been chosen.
- Proceed with standard K-Means using the aforementioned distance measure

The algorithm described above ensures that intra-cluster correlations are maximized, while inter-cluster correlations are minimized.

3.2 Genetic Algorithm

Every run of the genetic algorithm is begun with a pool of randomly generated portfolio 'chromosomes'. Each portfolio is composed of n $(stockname_i, w_i)$ pairs, where n is the cardinality of the output portfolio, as specified by the user. Each pair defines a constituent 'gene' of the candidate chromosome. $stockname_i$ and w_i denote the stock name and portfolio weight of the ith gene respectively. The initial weights w_i are chosen randomly, ensuring that their sum equals 1. Smoothing may be done to avoid the possibility of any stock being assigned a value of w_i that is too low to be realistically possible. Using such a description, a randomly generated portfolio 'chromosome' can be shown as in Table I.

$stockname$	$weight$
CSE	0.076
DLTR	0.214
SNDK	0.259
EQT	0.056
IDTI	0.099
AGX	0.151
TST	0.109
GBCI	0.036

Table 1: Portfolio example

The fitness function $f(P)$ used to determine the health of a portfolio is given by the product of its expected returns ($E[Returns_P]$), with its Shannon entropy ($H(P)$) with respect to the stock clusters generated in the first stage of the framework. Thus,

$$f(P) = E[Returns_P] * H(P) \tag{4}$$

where the $H(P)$ is defined as

$$H(P) = \sum_{i=1}^{N_c} -P_i \; log \; P_i \tag{5}$$

where N_c is the number of clusters generated in the first stage of the framework, and P_i is the sum of the weights of all stocks in the portfolio that belong to cluster i.

The expected portfolio returns value $E[Returns_P]$ is defined as

$$E[Returns_P] = \sum_{j=1}^{n} w_j \; E[Returns_j] \tag{6}$$

where $E[Returns_j]$ denotes the expected returns from the jth stock.

At each iteration, pairs of candidates are chosen for mating $n/2$ times (with repetition allowed) to make up the population for the next iteration. Each pair of parents yields two child portfolio chromosomes. However, instead of basing the selection solely on the fitness function, we use sigma scaling over the original fitness values. This promotes sufficient exploration of the solution state before convergence to an optimum. The sigma-scaled fitness function is given by

$$f_{ss}(P) = \frac{f(P) - \mu_f}{\sigma_f} \tag{7}$$

where μ_f denotes the mean of the fitnesses of the population candidates, while σ_f denotes their standard deviation.

This modified fitness score ensures that the algorithm doesn't converge directly towards the 'apparent' best solution right at the beginning of the procedure. Instead, while the variance shown by the fitness values is high, it explores the solution space, and gradually converges to an optimum result as the variance starts reducing. This avoids the problem of convergence to a local optimum.

Cluster 1	CWCO BCO SNDK ATK MGLN BCPC PCTI FBNC ARLP ANGO WIT
Cluster 2	HELE AGX EWBC LANC BMY MDU GRMN KOG ATMI
Cluster 3	MSI ATW NEU DIS APOG AMG CSE CPA WLK SHW VASC CF ASCMA UNF ECOL BKH ITC ALGT
Cluster 4	ADBE DRQ FISI IPAR IDTI CMCO HBAN HLS FFIN KEY FLO
Cluster 5	AAPL ESP MOFG CA GBCI VRSN SSW
Cluster 6	EQT AMZN
Cluster 7	INTC SHPG EGAS DLTR
Cluster 8	EXLP DVN TST RTI ORCL GTE AI DWA ICGE ROST

Table 2: Stock clusters formed

	C1	C2	C3	C4	C5	C6	C7	C8
C1	0.587	-0.147	-0.212	0.097	-0.406	-0.097	0.09	0.328
C2	-0.147	0.635	0.484	0.262	0.472	-0.110	0.100	-0.342
C3	-0.212	0.484	0.740	0.414	0.576	0.261	-0.337	-0.218
C4	0.097	0.262	0.414	0.636	0.413	-0.011	-0.292	0.231
C5	-0.406	0.472	0.576	0.413	0.770	0.074	-0.234	-0.178
C6	-0.097	-0.112	0.261	-0.011	0.074	0.758	-0.331	-0.005
C7	0.091	0.100	-0.337	-0.292	-0.234	-0.331	0.638	-0.129
C8	0.328	-0.342	-0.218	0.231	-0.178	-0.005	-0.129	0.564

Table 3: Average ρ_t values between stocks from pairs of clusters

For crossovers, the algorithm randomly redistributes the $(stockname_i, w_i)$ pairs found in the two parents, ensuring that two pairs with the same $stockname$ value don't end up in the same child. The weights in each of the children are then normalized to make sure that their sum equals one. A mutation on a child chromosome is performed by choosing a $(stockname_i, w_i)$ at random, and changing the $stockname$ to an arbitrary value from the available range.

Since we want a greater exploration of the solution space, we choose a rather high mutation rate of 0.2 and a crossover rate of 0.7. Elitism for one candidate portfolio is implemented, wherein the best candidate from every generation is carried forward to the next. This ensures that the 'best' portfolio, if created in a generation before the last one, is carried forward till the end. The algorithm is made to run until a user-decided I number of consecutive generations dont yield the same maximum fitness value (which would mean that the best solution hasn't shown an improvement since I iterations).

Once the genetic algorithm finishes its run, the best portfolio is chosen from the last generation of candidate portfolios based on the fitness score.

Thus, we get a portfolio that is sufficiently diversified (optimal entropy with respect to correlation-based clusters) and has a good return/risk ratio (Genetic optimization of fitness score).

4 Experiments and Results

The effectiveness of the proposed clustering approach is demonstrated using 72 of the top stocks as listed on MSN Money[2] during November 2013. The experimentation was conducted using Python-based code, with historic quotes (Adjusted Closing price) of two years (January 2011 January 2013) obtained from Yahoo! Finance[3] using the $ystockquote$[4] library. It was ensured that the stocks were obtained from various sectors including Capital Goods, Technology, Public Utilities, etc. The number of clusters was set to 7. As expected, the clusters didn't consist of stocks homogeneous with respect to industry (Table 2).

Table 3 shows the average time-weighted correlation between stocks taken from each cluster-cluster pair. It can be seen that for each cluster, its ρ_t value with respect to itself is the highest, while inter-cluster values are generally low. It was found that the average ρ_t between stocks of the same cluster was 0.667, while that between stocks of two different clusters was 0.027. With respect to the stock prices of two months following the training data, it was found that ρ_t among stocks of the same clusters was still quite high at 0.543, while that between stocks of different clusters had risen to 0.201, though still substantially lower than the intra-cluster value.

We then move on to the genetic algorithm implementation, to construct an optimal portfolio. As expected, the portfolio as output by the genetic algorithm is different after every run keeping the stock clusters same. Table 4 shows a portfolio constructed using the aforementioned stocks from MSN Money. It's corresponding returns and risk after 2 months post January 2013 are shown in the table. Figure 1 shows the highest fitness value in the population as a function of the progressing generations. As a standard, I constructed a portfolio having equal weightages with respect to all stocks under consideration, and included it's relevant values in Table 4. It is clear that the portfolio generated by the proposed framework shows a much higher returns/risk ratio

[2]http://money.msn.com/

[3]http://finance.yahoo.com

[4]https://pypi.python.org/pypi/ystockquote

$stockname$	$weight$	$stockname$	$weight$
CSE	0.134		
SNDK	0.331	Equal	
AI	0.060	Weights	
ATMI	0.087	To	
AGX	0.191	All	
TST	0.125	Stocks	
BMY	0.073		
$Two-month\ returns$	13.36%	$Two-month\ returns$	5.19%
$Two-month\ risk$	1.49	$Two-month\ risk$	3.22
$Returns/Risk\ ratio$	8.966	$Returns/Risk\ ratio$	1.612

Table 4: Comparison of portfolios

$stockname$	$weight$
AXE	0.124
GS	0.204
INTC	0.075
CSCO	0.110
GE	0.138
T	0.096
V	0.253
$One-month\ returns$	11.75%
$One-month\ risk$	7.215
$One-month\ Returns/Risk\ ratio$	1.628
$Two-month\ returns$	3.15%
$Two-month\ risk$	6.93
$Two-month\ Returns/Risk\ ratio$	0.45

Table 5: Portfolio example

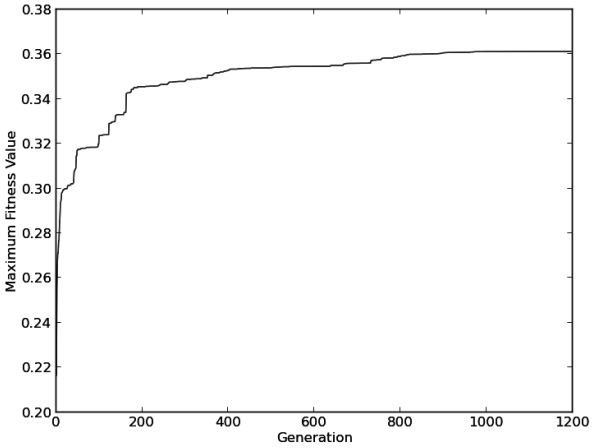

Figure 1: Highest fitness over progressing generations

(Percent Returns/Risk), compared to the standard portfolio. This can be claimed to be true for various time periods and cardinalities, after extensive experimentation.

To further illustrate the effectiveness of the framework, I constructed portfolios using the stocks from the DJI index, one of the most reputed market indexes in the world. Table 5 shows an example of such a portfolio. The training was done on historical data from the period August 2011 to August 2013. As seen in the table, the returns from the constructed portfolios for the one-month and two-month periods following August 2013 beat the corresponding DJI returns (9.6%, -7.7%). I performed 50 runs of the framework, keeping the stock clusters same. The mean and standard deviation shown by the returns from the constructed portfolios for a one-month period are 10.62% and 0.007% respectively. As for the portfolio risk, the mean and standard deviation shown by the values from the output portfolios for a one-month period are 7.346 and 0.8 respectively. During the same period, the risk shown by the DJI was 2.68. Though the risk portrayed by the framework output doesn't beat the DJI index, it is apparent that it is still quite stable to market fluctuations, and thus yields positive returns even when the index yields negative ones.

The experimental results as shown in this section conclusively prove that the proposed framework is largely successful in generating stable portfolios from given sets of stocks, provided that the appropriate training data is available.

5 Conclusions and Future Work

This paper details a stochastic yet sound framework for optimal stock portfolio construction. The results, as demonstrated in Section 4, are promising enough to suggest the effectiveness of the outlined methodology. The strength of the idea lies in the usage of a genetic algorithm to optimize returns as well as quantified diversification. Similar to biological evolution, the evolutionary technique used in this paper ensures that only those portfolios that are 'fit' enough to tackle the trends shown by historical data make it to the last generation. It should be noted that the focus of this work is to create a stable portfolio, rather than just a profitable one- in essence, the kind of portfolio an investor is likely to trust. The concept of time-weighted correlations, at the heart of the framework, gives a powerful notion of the $expectedtrends$ among stock prices. Considering the random walk theory, which states that stock prices tend to take unpredictable paths, the experimental results portray the potential of ρ_t as an indicator of future trends.

Furthermore, each of the two stages in the described algorithm has its own strengths and can be used for other related

tasks. For example, correlation clustering can be used for stock classification, while the genetic algorithm can be used for optimizing the weights over a fixed set of securities. The only necessary change would involve modifying the mutation technique to mutate a random gene's weightage field instead of the stockname. The presented framework can also be applied to portfolio generation utilizing other kinds of securities, provided the required data is available. Moreover, for a commercial application of my work, transaction costs would also need to be accounted for. These could be factored into the fitness score used for the selection operator as a part of the GA.

There are various ways in which the outlined framework could be further improved. For starters, the algorithm currently requires the user to input the number of target slots in the output portfolio. This requirement could be discarded by applying intelligent clustering methodologies like divisive clustering with an appropriate stopping criterion. As a result, the optimal number of clusters could be learnt and the same could be used as the cardinality for the output. Future work in this direction would also involve exploring dynamic extensions to the methodology. Since one portfolio cannot be perennially profitable, incremental transactions could be suggested to the user at regular intervals. However, since a GA is largely stochastic, the portfolio generated at any run may be substantially different in terms of its constituent stocks, with respect to any prior run. If these 'suggestions' are followed, the transaction costs would overshoot any profits made. This can be countered by biasing the GA process to certain stocks and weights by modifying the genetic operators of selection, crossover and mutation. Another way to do this would be to model transactions as chromosomes, and optimize the fitness of the resultant portfolio using another GA framework. In any case, the usefulness of my approach for one-time portfolio generation is apparent and should be explored for further possibilities.

6 Acknowledgments

I would like to express my gratitude towards Mr. Vishnu Bhaskar Nayak for the tremendous support and encouragement provided towards my work.

References

Aftalion, T. 2012. Genetic algorithms for portfolio optimization. Master's thesis, School of Computer Science, Carnegie Mellon University.

Chi-Ming Lin, M. G. 2007. An effective decision-based genetic algorithm approach to multiobjective portfolio optimization problem. *Applied Mathematical Sciences* 1(5):201–210.

David Arthur, S. V. 2007. K-means++: The advantages of careful seeding. *SODA 2007:1027-1035*.

Hachloufi, M. E. Stocks portfolio optimization using classification and genetic algorithms. *Applied Mathematical Sciences* 6(94):4673–4684.

John Campbell, Andrew Lo, A. C. M. 1997. *The Econometrics of Financial Markets*. Princeton: Princeton University Press.

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

Markowitz, H. 1959. *Portfolio Selection: Efficient Diversification of Investments*. John Wiley & Sons.

Mitchell, M. 1998. *An Introduction to Genetic Algorithms*. MIT Press.

Pankaj Sinha, Abhishek Chandwani, T. S. 2013. Algorithm of construction of optimum portfolio of stocks using genetic algorithm. *MPRA* (48204).

Paul Embrechts, Alexander McNeil, D. S. 1999. Correlation and dependence in risk management: Properties and pitfalls. *Risk 69-71*.

Pereira, R. 2000. Genetic algorithm optimization for finance and investments. *MPRA* (8610).

Rosén, F. 2006. Correlation based clustering of the stockholm stock exchange. Master's thesis, School of Business, Stockholm University.

Slimane Sefiane, M. B. 2012. Portdolio selection using genetic algorithm. *Journal of Applied Finance & Banking* 2.

T. Vallée, M. Y. 2003. Prsentation des algorithmes gntiques et leurs applications en conomie. *Revue dconomie politique* 4(2).

William N. Goetzmann, A. K. 2008. Equity portfolio diversification. *Review of Finance* 12:433–463.

Yang, X. 2006. Improving portfolio efficiency: A genetic algorithm approach. *Computational Economics* 28(1):1–14.