遺伝的アルゴリズムを用いて多目的ポートフォリオの最適化とリバランシング

Multi-objective Portfolio Optimization and Re-balancing using Genetic Algorithms

ソーム　ヴィシャル [1*]　　伊庭　斉志 [1,2]　レオン　パラフォックス [1,3]
Vishal Soam[1*]　　Hitoshi Iba[1,2]　　Leon Palafox[1,3]
[1] 東京大学電気系工学専攻
[1]Engineering Department, The University of Tokyo, Tokyo

Abstract: The Portfolio Optimization problem is a multi-objective resource allocation problem where money to be allocated to the assets is the resource. The problem consists of the selection of assets from thousands of them available in the market, weigh them properly in the portfolio in order to minimize the risk and maximize the expected return of the investment. In our work, the main motive is to make the portfolio more realistic, apart from achieving better results. We introduce mainly: 1) The inclusion of real life constraints namely – realistic transaction costs, 2) The new co-ordinate ascent Genetic Algorithm, taking into consideration the traded volumes. We compare our results with simple GA based method and the index, and observe a noticeable improvement.

1 Introduction

Investment Portfolios are used by financial institutions, individuals, funds etc in the management of long term funds. According to the "*time value of money*" theory money loses value over time. 100 yen paid today is more than exactly the same amount paid next year. So it is wise to invest money but it is also risky to invest everything in one asset or a small number of assets, since the markets can change suddenly. The Markowitz Portfolio Theory [1] describes how to minimize the risk of a financial portfolio by the method of diversification. This model is the base till date to calculate the optimal distribution of capital in order to minimize risk and maintain a target return.

The large data sets and the constraints involved make the problem tough and almost impossible to solve by numerical methods and calls for approaches like evolutionary algorithms. The basic genetic algorithm flowchart is shown in figure 1.

Portfolio Optimization problem has been popular with the genetic algorithm community in recent years. However, the problem is big and most of the works concentrate on either 1) selection of assets,

連絡先：東京大学電気系工学専攻
E-mail: vishal @iba.t.u-tokyo.ac.jp

2) weighing the assets, 3) solving the problem for single scenarios or 4) including real life constraints with one of the first three.

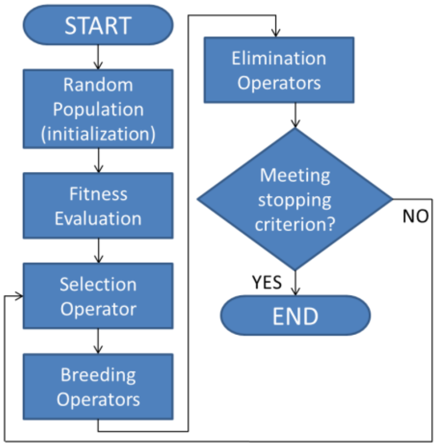

Figure 1: **The basic genetic algorithm flowchart**

In our work here, we use our approach for managing the portfolios for long-term. We concentrate on selecting the assets, weighing them, re-balancing the portfolios for the desired time period and also on the inclusion of some real-life constraints.

2 The Portfolio Problem

The Portfolio Optimization problem falls under the traditional resource allocation problem. Here the job is to distribute a limited "resource" to a number of "jobs" while satisfying some utility functions [2].

In Portfolio Optimization problem, the capital available for investment is the limited resource and the jobs are the varied assets in which this capital can be invested (like stocks, bonds forex assets etc.). Portfolio Return to be maximized and the Portfolio Risk to be minimized are the utility functions.

Markowitz's Modern Portfolio Theory [1] forms the basis of the Portfolio Optimization problem. This model could be solved by numerical methods like Quadratic Programming [3].

However, this model ignores real-life constraints (like, large number of assets, trading costs etc.). In this case the search space becomes large and non-continuous and unsolvable by numerical methods like Quadratic Programming. Here the Evolutionary Computation works better.

2.1 Modern Portfolio Theory

The MPT says, a financial asset can be represented in terms of its return and its risk. The return of an asset is the relative change of its value over time. The risk of an asset, on the other hand, is the variance of its return over time.

If we have N assets available in the market, a portfolio P can be defined as a set of N real valued weights ($w_0, w_1, w_2,.., w_N$). These weights must fulfill two restrictions [5] :

$$\sum w_i = 1 \quad (1),$$

$$0 \leq w_i \leq 1 \quad (2)$$

The utility of the portfolio is evaluated according to its *Estimated Return* and its *Risk*. The *Estimated Return and Risk* is calculated as follows:

$$R_p = \sum r_i w_i \quad (3) ,$$

$$\sigma_p = \sum\sum \sigma_{ij} w_i w_j \quad (4), \text{(both over } N)$$

Where N is the total number of assets in the portfolio, r_i is the given estimated return of asset i, w_i is the weight of the ith asset. Again, σ_p is the total risk of the portfolio, σ_{ij}, $i \neq j$ is the covariance between i and j and $\sigma_{ii}=\sigma_i^2$ is the deviation of the estimated return of asset i. Although these are the basic estimation methods for the risk and return of the portfolios, there have been work done in some researches suggesting the use of other methods as well [4],[6].

These two utility measures can be used

separately to determine the optimal portfolio, or they can be used combined. The *Sharpe Ratio* measures the tradeoff ratio between risk and return for a portfolio, and is defined as follows:

$$Sr = \frac{E(r_P) - Rriskless}{\sigma_P} \qquad (5)$$

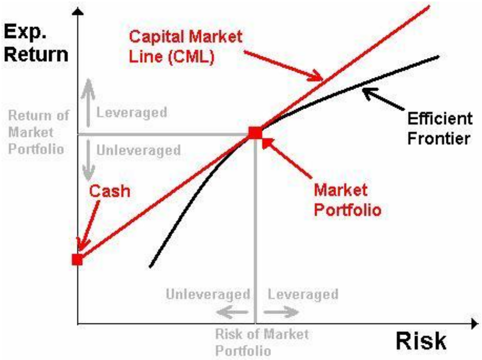

Figure 2**: Risk-expected return projection of candidate portfolios. The search space is bounded by the Efficient Frontier. Sharpe Ratio is the angle between a portfolio and the risk-free (cash) state.**

Here $R_{riskless}$ is the risk-free rate (ideal), an asset which has zero risk and a low return rate. In real life they do not exist since they are an ideal case, but they can be thought of as "money in cash" or an asset like government bonds of stable economies. The *figure 2* explains the relationship between these three utility measures.

2.2 Dynamic Market Behavior

Modern Portfolio Theory considers the Portfolio Optimization problem as a static problem and assumes that past or present of the portfolio has no effect on the future of the portfolio. However, economic changes, market changes and various changes in the policies make it a dynamic problem.

Therefore, as in our research, the Dynamic Portfolio Optimization or re-balancing as we call it, is the problem of generating a trading strategy that keeps the portfolio optimized with a high level of return and low risk in face of a dynamically changing market.

3 Related Works

There have been works which use a single array with real values for the weight of each asset [7, 8].

A relatively newer strategy has been to use both a binary and a real-valued array to select the assets of the portfolio. Here the binary array indicates the presence or absence of an asset and the real-valued array shows the weights assigned to each asset in the portfolio.

Another way is to use GP or tree-based structures to calculate suggested weights or ranking of assets [11][12][13][14].

4 Our Work

4.1 Representation of a Portfolio

We have seen a lot of works which particularly concentrates only on the selection of assets for the Portfolio. These researches basically use a binary array to represent a portfolio. Then we have also seen researches which use real-valued arrays to represent the weights assigned to the selected assets and then tune them to reach the final goal.

In our research we propose to use a two-step algorithm. First, we use a binary array to represent the inclusion or non-inclusion of a particular asset in the portfolio and then we use the real-valued array to represent the weights of those included assets (figure 3).

4.2 Selection Process

The goal of this operator is to select from the current population the individuals which will be used to construct the next population. The individuals with high fitness value in the population must have a higher probability of being chosen to be recombined than others.

In our proposed method we use: **1)** The fitness measure as the Sharpe Ratio to evaluate the individuals. **2)** Then we apply the Deterministic Tournament Selection (DTS). **3)** We apply the Elite Strategy which means that regardless of crossover or mutation, one or more best individuals are taken from the present population to the next one to tilt the final solution towards the previous best so as to minimize the transaction costs.

1	0	1	0	…	…	1
Asset 1	Asset 2	Asset 3	…			Asset N
0.26	0	0.35	0	…	…	0.39

***Figure 3*: The representation of a Portfolio. *1* and *0* represent the inclusion and non-inclusion of an asset in the portfolio in the binary array and the real-valued array represents the weights of the included assets.**

4.3 Crossover

The role of crossover operator is to perform exploitation of the search space by testing new solutions with characteristics of two good individuals. The offspring created contains information from both the parents.

We propose to use *k-point* technique (gave better results than the uniform/linear crossover in preliminary simulations). Refer to Figure 5 and Figure 6.

In this technique, *k* points are randomly chosen from the genetic representation and the offspring is composed by alternatively copying elements from two parents, changing from one parent to the other at each of the *k* points.

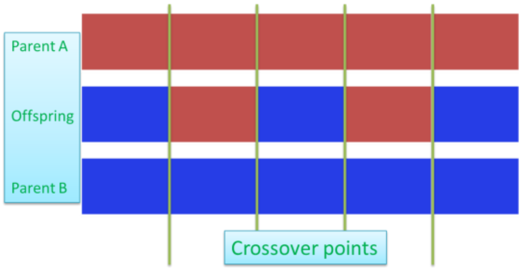

***Figure 4*: *K*-point crossover technique**

4.4 Local Search Mutation – Co-ordinate ascent

While the crossover operation performs the exploitation role of the genetic algorithm, the mutation performs the exploration. Mutation operator alters the gene values to explore better results and to avoid the local minima.

Binary (index) array example	1	0	1	0	1
	1	0	0	0	1
Weight array example	0.35	0.25	0.13	0.22	0.05
	0.35	0.21	0.13	0.22	0.09

***Figure 5*: Mutation technique**

We have used a new mutation approach: a 2-step ***co-ordinate ascent mutation***.

In the first step, the usual mutation operator is used. Here an asset is chosen randomly and its weight is altered and the portfolio goes through evaluation.

In the second step, we use the ***guided local search co-ordinate ascent mutation operator***.

Algorithm 1

```
Sharpe =Sharpe(weight)
for (i=0; i< portfolio array size; i++)
 base=weight[ i ]
  weight[ i ]+= alpha
  Plus_weight=weight/(1+alpha)
  Sharpe_Plus=Sharpe(Plus_weight)
  weight[ i ]-=2*alpha
```

```
 if weight[ i ]<0
   weight[ i ]=0
   Minus_weight=weight/(1-base)
 else
   Minus_weight=weight/(1-alpha)
 Sharpe_Minus=Sharpe(Minus_weight)
return weight of max_Sharpe
```

Figure 6: Guided local search co-ordinate ascent mutation operator.

Here, the last step compares the Sharpe, Sharpe_Plus and Sharpe_Minus and returns the weight array of the maximum out of these three. This step works really well, and it consequently removes all the insignificant weights from the portfolio making it more realistic.

4.5 The Re-balancing Problem

Rebalancing refers to making small adjustments in the portfolio over time, in order to maintain the estimated return and risk, according to the changes in the market conditions. Just as the optimization adjusts the weights of assets included in the portfolio during normal evolution, we expect that in the same way it can correct the weights of the final individual after changes in the market.

But since re-balancing means we have to pay transaction costs, the new objective during this problem becomes: *change in expected return in the portfolio > transaction costs included.*

We have seen works that use Euclidean Distance to measure the changes in the portfolio, but they have some drawbacks as well[15]. We propose to use the calculations used in investment banks:

$$Cost_i = \begin{cases} Cost_{fixed}, & \text{if } 0<T_i<T_{min} \\ T_i * \delta_c, & \text{if } T_i>T_{min} \end{cases} \quad (6)$$

Where $Cost_i$ is the transaction cost, T_i is the amount of *i*th asset traded, $Cost_{fixed}$ fixed amount charged for transactions below a certain amount, T_{min}, and δ_c is some small % of the total transaction value.

We also use the ***Seeding*** technique which means that while initializing the first random population for the re-balancing problem at time *t*, we also include the best individuals from the final population of the scenario at time *(t-1)*. This technique basically helps our search by inclining it towards the successful regions in the past scenarios (Figure).

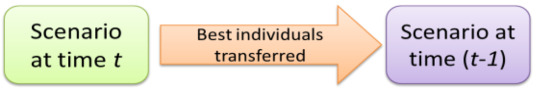

Figure 7: Seeding technique

4.6 Trading Volumes

"Volume precedes price" is a famous saying in the financial field. In fact, larger traded volume than normal, with an increase in price of an asset on *i*th *day* indicates that the price will continue to increase, at least for a few coming days. Same applies to the larger traded volume with a decrease in asset prices.

Thus, the traded volume has extremely important information hidden in them and we introduce this element while calculating the daily returns for assets in our portfolio. The point is that bigger trading volume than the average for a given period of time indicates the fore coming trend of an asset in the market and therefore, the returns should be weighted accordingly with the volume.

```
Algorithm 2
for a period of time t
 calculate average traded volume Vt
```

$R_{ait} = R_{ait} * V_{it/} V_t$

Where, R_{ait} = Return of asset a on the i^{th} day of period t, VR is the volume ratio = $V_{it/} V_t$, V_{it} is the traded volume on the i^{th} day of period t.

So basically, the volume ratio element inflates or deflates the return (or loss) depending on the volume traded.

5 Experiments

To verify the performance of our Portfolio and the inclusion of traded volume element in it, we performed several simulations.

We first evolved a portfolio with our algorithm, to compare it with the market index and also with the simple GA. We then re-balanced the portfolio each month for the next 12 scenarios (months) i.e. during the recession of 2008. The goal is to check how our algorithm works as compared to the market index and also to the simple GA. We also wanted to check the performance of the portfolio during the difficult market situations like the recession of 2008 when most of the assets produce losses.

5.1 Datasets and Parameters

1) Dow Jones Industrial Average (all 30 assets, relatively smaller dataset), 2) NASDAQ 100 (all 100 assets, bigger dataset). Data used for evolving the portfolio: Jan 2007 ~ Dec 2007. The portfolio is then rebalanced each month for the next 12 scenarios i.e. Jan 2008 ~ Dec 2008.

Number of generations: 200, crossover rate: 0.2, mutation rate: 0.2. The rate of return for the riskless asset: 1.035%. The parameter alpha in mutation step: 0.005.

5.2 Experimental Results and Discussion

Experimental results have been shown in the figure 8.

In figure 8(a), the profits results are better than the index itself. However, the returns are not satisfactorily high, especially in the months of July and August 2008. The reason for this can be that the fitness function tries to maximize the Sharpe Ratio and does not care about the returns as such.

However, the Sharpe Ratio in figure 8(b) outperforms the index as well as the simple GA, both with and without traded volume included. This indicates that although our algorithm does not care about the returns, the Sharpe Ratio is still very satisfactory, suggesting that the risk involved with the Portfolio is considerably low.

Figure 8(c) shows the comparison of profits involved with NASDAQ100. Here the Portfolio performs much better with higher returns compared to both, the index and the simple GA results. The difference in the Portfolio with Dow Jones and NASDAQ can be because of the differences in the assets involved with them. However, it might be indication of more aggressive exploration technique required for better results.

Figure 8(d) shows the Sharpe ratio calculated with NASDAQ 100 assets. Here as well, our Portfolio outperforms both, the index and the results of simple GA with and without traded volumes.

6 Conclusions

We have expanded the simple GA work with the inclusion of guided local search co-ordinate ascent mutation operator and the traded volumes.

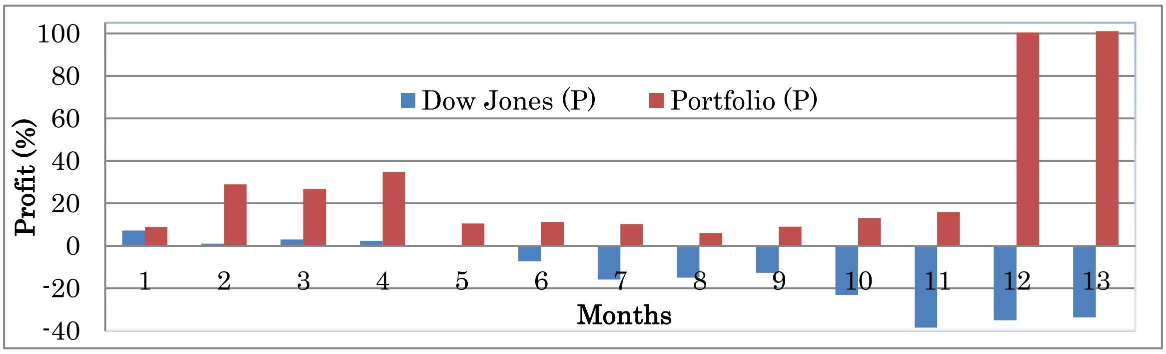

Figure 8(a)

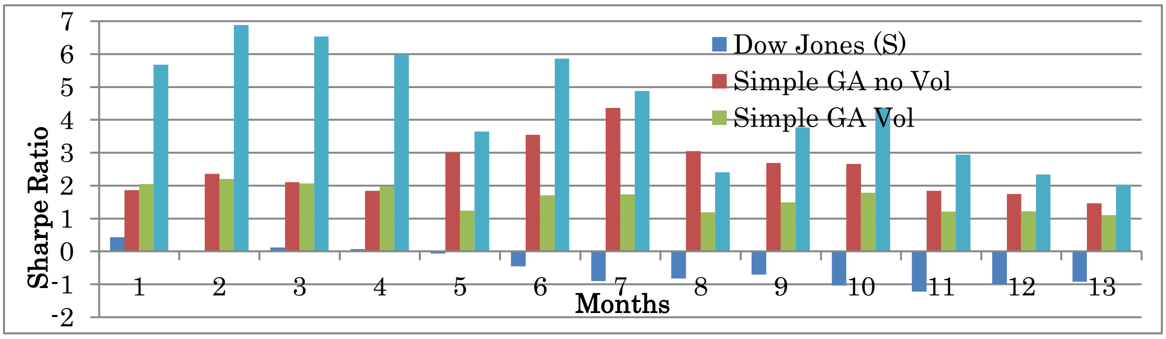

Figure 8(b)

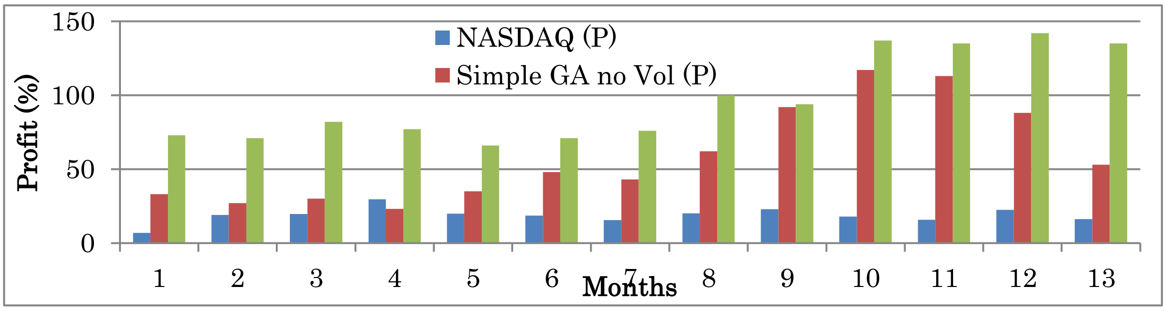

Figure 8(c)

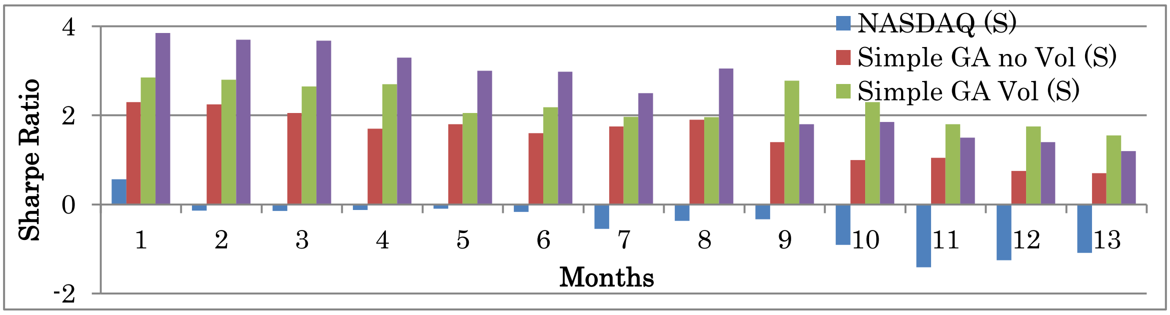

Figure 8(d)

Figure8: 8(a) and 8(b) compares the Profit and Sharpe ratio, of Dow Jones index and Portfolio, respectively. 8(c) and 8(d) compares the Profit and Sharpe Ratio, of NASDAQ index and Portfolio, respectively. Months are from December 2007 ~ December 2008 (13 months).

Simulation results show that our Portfolio performs better than the index and the simple GA algorithms. However, the lower than expected profits in Dow Jones scenarios motivates us to improve our algorithm.

References

[1] H. Markowitz. *Mean-Variance analysis in Portfolio Choice and Capital Market*. Basil Blackwell, New York, 1987.

[2] Toshihide Ibaraki and Naoki Katoh. *Resource Allocation Problems - Algorithmic Approaches*. The MIT Press, 1988.

[3] Yuh-Dauh-Lyu. *Financial Engineering and Computation*. Cambridge Press, 2002.

[4] Harish Subramanian, Subramanian Ramamoorthy, Peter Stone, and Benjamin J. Kuipers. Designing safe, profitable automated stock trading agents using evolutionary algorithms. In *GECCO 2006 - Genetic and Evolutionary Computation Conference*, pp. 1777–1784, Seattle, Washington, July 2006. ACM Press.

[5] Claus Aranha. Portfolio Management with Cost Model using Multi Objective Genetic Algorithms. Master of Science Thesis, The University of Tokyo, 2007

[6] Shu ping Chen, Chong Li, Sheng hong Li, and Xiong wei Wu. Portfolio optimization with transaction costs. *Acta Mathematicae Applicatae Sinica*, Vol. 18, No. 2, pp. 231–248, 2002.

[7] Ronald Hochreiter. An evolutionary computation approach to scenario-based risk-return portfolio optimization for general risk measures. In M. Giacobini et al., editor, *EvoWorkshops 2007*, No. 4448 in LNCS, pp. 199–207. Springer-Verlag,2007.

[8] Piotr Lipinski, Katarzyna Winczura, and Joanna Wojcik. Building risk-optimal portfolio using evolutionary strategies. In M. Giacobini et al., ditor, *EvoWorkshops 2007*, No. 4448 in LNCS, pp. 208–217. Springer-Verlag, 2007.

[9] Claus Aranha and Hitoshi Iba. Modelling cost into a genetic algorithm-based portfolio ptimization system by seeding and objective shar ing. In *Proc. of the Conference on EvolutionaryComputation*, pp. 196–203, 2007.

[10] Felix Streichert, Holger Ulmer, and Andreas Zell. Evolutionary algorithms and the cardinality constrained portfolio optimization problem. In D. Ahr, R. Fahrion, M. Oswald, and G. Reinelt, ditors, *Operations Research Proceedings*. Springer, September 2003.

[11] Claus Aranha. Development of Hybrid Evolutionary Techniques to Solve Resource Allocation Problems, Doctor of Information Sciences, The University of Tokyo, 2010.

[13] James Cunha Werner and Terence C. Fogarti. Genetic control applied to asset managements. In J.A. Foster et Al., editor, *EuroGP*, LNCS, pp. 192–201, 2002.

[14] Wei Yan and Christopher D. Clack. Evolving robust gp solutions for hedge fund stock selection in emerging markets. In *GECCO 2007 – Genetic and Evolutionary Computation Conference*, London, England, July 2007. ACM Press.

[15] Claus Aranha. Development of Hybrid Evolutionary Techniques to Solve Resource Allocation Problems, Doctor of Information Sciences, The University of Tokyo, 2010.