Portfolio Optimization Using
Multi-Objective Particle Swarm Optimization

Viriya Yimying, National Institute of Development Administration, Thailand
Ohm Sornil, National Institute of Development Administration, Thailand

The Asian Conference on Technology, Information & Society 2015
Official Conference Proceedings

Abstract
Portfolio optimization is an important problem in finance. Its goal is to discover an efficient frontier which yields highest expected return on each level of portfolio variance. The problem has multiple objectives, and its search space is large. Multi-objective particle swarm optimization is a multi-objective optimization method, developed from particle swarm optimization, by incorporating non-dominated sorting and crowding distance. This research proposes a portfolio optimization technique based on multi-objective particle swarm optimization. Two objectives used in the research are maximization of return and minimization of portfolio risk. The technique is evaluated using daily stock total return index gross dividends from Stock Exchange of Thailand between 2006 and 2014. The technique is deployed in unknown trading periods, and the results are compared with standard market benchmarks. The results show that the proposed technique performs well in comparisons with the market benchmarks.

Keywords: Portfolio optimization, Multi-objective particle swarm optimization, Markowitz's model, portfolio management

Introduction

Investors throughout the world are interested in portfolio management. The main focuses for this problem are on expected return and risk management. Portfolio theory, first introduced by (Markowitz, 1952, 1959), is applied to portfolio allocation to aid security selection and asset allocation to gain the highest expected return while having an acceptable risk level. Later on, this theory was developed into others theories such as the capital market theory.

There are many constraints that a fund manager has to consider before making decisions on investment allocation, such as those defined by the investment committee and by the securities and exchange commission, such as the maximum and the minimum weights of shares, the portfolio risk, and the acceptable value at risk. Besides, there are other factors that the fund manager should consider such as liquidity and dividend yield (Clarke et al., 2002). Because the search space of portfolio optimization is large and not suitable for the Brute force method while the population random sampling yields inconsistent solutions. A better approach is needed to obtain accurate and suitable solutions quickly.

Multi-objective particle swarm optimization (MOPSO) is developed from particle swarm optimization (PSO), introduced by Eberhart & Kennedy (1995), and based on the herd's behavior or swarm intelligence. A flock of birds or a swarm seeks for food by communicating with one another to assemble where they find good food. Along the way, if better food sources are discovered, they communicate back and fly to the best sources together. Later, Moore & Chapman (1999) applied PSO to search for multi-objective solutions, and there currently are numerous researches on applying PSO to various problems. At the same time, Raquel & Naval Jr. (2005) presented MOPSO which employs non-dominant sorting and crowding distance methods from Non-Dominated Sorting Genetic Algorithm-II (NSGA-II), created by Deb et al. (2000) and mutation by Coello et al. (2002, 2004). MOPSO has the same principle as PSO which males it suitable to find the best search space spot in a short time. PSO uses real-valued encoding and vector-based calculation and thus lends itself well to real-valued problems (Coello et al., 2002). Moreover, Mishra et al. (2009) compared the results of MOPSO and those of NSGA-II on a portfolio optimization problem without investment constraints. The results show the superiority of MOPSO over NSGA-II.

This research presents a portfolio optimization technique using MOPSO with investment constraints. In the rest of the paper, Section 2 presents the proposed technique. In Section 3, the technique is evaluated using actual stock prices from the stock exchange of Thailand, and the results are presented. Section 4 provides concluding remarks.

Proposed Technique

PSO is a population-based search algorithm, simulating the social behavior of birds within a flock. It is found to be very effective in a wide variety of applications and able to produce good results at a very low computational cost. PSO relies on two mechanisms: parent representation and fine tuning of the parameters.

A particle is a member (individual) of the swarm. Each particle represents a potential solution to the problem being solved. The position of a particle is determined by the solution it currently represents. PSO uses an operator that sets the velocity of a particle to a particular direction. The direction is defined by both the particle's greatest success (personal best or *pbest*) and the best particle of the entire swarm (global best or *gbest*). If the direction of the personal best is similar to the direction of the global best, the angle of potential directions will be small, whereas a larger angle will provide a larger range of exploration. Particles are flown through the search space. Changes to the positions of particles within the search space are based on the social-psychological tendency of individuals to emulate the success of other individuals.

The solution set of a problem with multiple objectives does not consist of a single solution. Instead, in multi-objective optimization, we aim to find a set of different solutions, i.e., the Pareto optimal set. In MOPSO, a swarm is first initialized. A set of leaders is also initialized with the non-dominated particles from the swarm. The set of leaders is usually stored, and quality measures are calculated for all the leaders. At each generation, a particle is flown. The particle is evaluated, and its corresponding *pbest* is updated. A new particle replaces its *pbest* particle usually when this particle is dominated or if both are incomparable (i.e., they are both non-dominated with respect to each other). After the particles are updated, the set of leaders is updated. Finally, the quality measure of the set of leaders is recalculated. This process is repeated for a certain number of iterations.

Portfolio Optimization Using MOPSO

The MOPSO process is shown in Figure 1. First, a number of particles are defined. Too few particles will not yield inclusive solution while too many particles will slow down the MOPSO process. From experiments, we find that the most suitable number is 200 particles, which is then set as the number of particle vectors. Elements of a vector are variables of a solution, i.e., portfolio weights. The initial values of weights are randomly set, and the sum of all weights w_i is equal to 1.

$$\sum_{i=1}^{n} w_i = 1$$

where w_i is the investment weight of security i, and n is the number of elements in a vector.

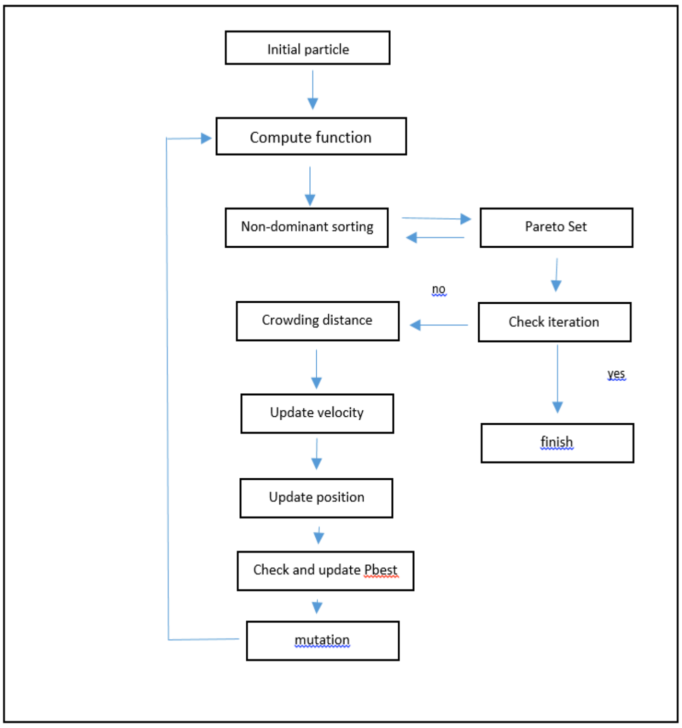

Figure 1: The MOPSO algorithm

Each vector is checked for any violation of the constraints. Objective values for each vector are then calculated. Two objective functions used in this study are:

Objective 1: Maximizing the expected return:

$$\text{Maximize } E(r_p) = \sum_{i=1}^{n} w_i r_i$$

where $E(r_p)$ is the expected rate of return of portfolio p
w_i is the investment weight of security i in portfolio p
r_i is the expected rate of return of security i.

Objective 2: Minimizing the portfolio risk

$$\text{Minimize } \sigma_p^2 = \sum_{i=1}^{m} w_i^2 \sigma_i^2 + 2\sum_{i=1}^{m}\sum_{j=1}^{m} w_i w_j \sigma_{ij}$$

The covariance of securities i and j (σ_{ij}) can be calculated as:

$$\sigma_{ij} = \frac{1}{n}\sum_{k=1}^{m}(r_{ik} - E[r_i])(r_{jk} - E[r_j])$$

where σ_p^2 is the portfolio variance
σ_{ij} is the covariance of securities i and j
m is the number of days

σ_i^2 is the variance of security i

r_{ik} is the daily return of security i on day k.

When i is equal to j, σ_{ii} becomes σ_i^2 (the variance of security i). The daily return r_{ik} can be calculated as:

$$r_{ik} = \left[\frac{close\ price_{ik} \times number\ of\ shares_{ik}}{(close\ price_{i,k-1} \times number\ of\ shares_{i,k-1}) \pm (adjust\ price_i \times adjusted\ shares_i)} - 1 \right] + total\ dividend\ yield_{ik}$$

$$total\ dividend\ yield_{ik} = \frac{dividend\ per\ share_i \times number\ of\ shares_{i,k-1}}{(close\ price_{i,k-1} \times number\ of\ shares_{i,k-1}) \pm (adjust\ price_i \times adjusted\ shares_i)}$$

where *close price*$_{i,k}$ is the closing price of security i on day k
number of shares$_{i,k}$ is the number of outstanding shares i on day k
dividend per share$_i$ is the cash dividend per share of security i
adjust price$_i$ is the price after adjustment (by the corporate)
adjusted share$_i$ is the number of shares after adjustment.

Non-dominant particles yield values on the Pareto front which are the best solutions of a multi-objective problem. Non-dominant sorting is performed to find non-dominant particles by comparing each particle to other particles with respect to each objective. If a particle is worse than any particle in an objective, that particle is dominated and eliminated. When non-dominant results are obtained, they are stored in the Pareto set in a sorted order, and crowding distances between two consecutive particles are calculated from the population. The crowding distance of particle i can be calculated as follows:

$$d_i = \sum_{j=1}^{n} \left(\frac{f_{j,i+1} - f_{j,i-1}}{f_{j,\max} - f_{j,\min}} \right)$$

where n is the number of objectives, and j is the particle order. Once the crowding distances for all particles are obtained, they are sorted from maximum to minimum before selecting the values to be the goal selection. From experiments, we find that the selection should be performed in steps. If the size of the Pareto set is less than 5, all particles are selected while if there are more than 5 particles, select the top 30 particles. Then, the *gbest* values are determined. Each particle value is *pbest* to calculate the velocity in order to find its new position x_{ij} according to the equation below:

$$x_{i,j}(t+1) = x_{i,j}(t) + v_{i,j}(t+1)$$

$$v_{i,j}(t+1) = \omega v_{i,j}(t) + c_1 r_{1,j}(t)[pbest_{i,j}(t) - x_{i,j}(t)] + c_2 r_{2,j}[gbest_j(t) - x_{i,j}(t)]$$

where $x_i(t)$ is the i-th particle's position at iteration t with respect to objective j
$v_{i,j}(t)$ is the velocity of particle i at iteration t with respect to objective j
c_1, c_2 are constant velocities where $c_1 + c_2 \leq 4$

r_1, r_2 are random values for speed adjustment where $r_1, r_2 \in U(0,1)$

ω is the inertia weight where $\omega > \frac{1}{2}(c_1 + c_2) - 1$

In our research, we adopt a commonly used values of $c_1 = 1.494$ and $c_2 = 1.494$ (Van den Bergh, 2006), and r_1 and r_2 are random values between 0 and 1, and they are independent from each other. The number of iterations is set at 3,000. However, we find that after 1,500 iterations, the Pareto front generally is unchanged. The inertia weight ω helps reducing the velocity of a particle to control severe movements. Its value is varied according to (Corazza & Komilov, 2009) as follows:

$$\omega = w_{\max} - \left(\frac{w_{\max} - w_{\min}}{iteration_{\max}}\right) \times iteration$$

where w_{max} and w_{min} are the maximum and minimum allowable security weights.

Portfolio Optimization Constraints

Two constraints are imposed on weights of securities in a portfolio. First, to not overemphasize on a particular stock, each stock must account for no more than 10% of the total portfolio. In addition, the proportion of an industry must not exceed 40% of the total portfolio.

Mutation Operation with Constraints

After updating velocity, the mutation operation is performed. Our mutation operation is a modification from the original operation by Coello et al. (2002) which mutates only one variable of a particle vector. Since a portfolio optimization problem considers many constraints, mutating only one variable decreases the effects of mutation. Our modified mutation makes changes to every value in a vector to expand the search space, as shown in Figure 2.

```
for unit = 1 to number_of_particles
    mutation_rate = 0.5;
    if (1-iteration/max_iteration)^(5/mutation_rate) > rand
        for element = 1 to vector_size
            mutation_range = (weight_max – weight_min) *
                                        (1-
         iteration/max_iteration)^(5/mutation_rate);
            UB = particle(unit, element) + mutation_range;
            LB = particle(unit, element) – mutation_range;
            particle(unit, element) = (rand(1)*(UB-LB))+LB;
        end for
    end if
end for
```

Figure 2: The mutation operation

Each weight in a particle vector is verified if there is any violation of the constraints. If any, adjustments are made to limit the weight values according to the constraints. The industry proportions are checked, and adjustments are performed to limit those proportions by equally updating the weights that are in the same direction of the difference. Then, the final adjustment is made in order for the total sum of weights in a portfolio to be 1(or 100%). After adjustments, the process of finding non-dominant particles as described earlier is repeated until reaching the specified number of iterations.

Experimental Evaluations

According to the capital market theory by Markowitz (1952), risk can be divided into two types: systematic risk and unsystematic risk. Systematic risk cannot be eliminated because it is the stock market risk while unsystematic risk can be eliminated through diversification. This is because stocks and shares in different industries have different returns depending on the business cycles. For this reason, we select 5 stocks with highest capitals from each industry group, i.e., Agro and Food (AGRO), Consumer Products (CONSUMP), Financials (FINCIAL), Industrials (INDUS), Property and Construction (PROPCON), Resources (RESOURC), Services (SERVICE), and Technology (TECH) industries, as follows:

AGRO industry consists of CPF, MINT, TUF, TF and KSL
CONSUMP industry consists of SUC, ICC , MODERN, TR and SITHAI
FINCIAL industry consists of SCB, KBANK, BBL, KTB, BAY and TPC
INDUS industry consists of TPC, STANLY, TCCC, VNT and SSI
PROPCON industry consists of SCC, CPN, SCCC, LH and PS
RESOURC industry consists of PTT, PTTEP, GLOW, TOP and RATCH
SERVICE industry consists of CPALL, AOT, BDMS, MAKRO and BIGC
TECH industry consists of ADVANC, INTUCH, TRUE, DELTA and JAS.

The proposed method is evaluated using four sets of data which span 4 different periods of time which are: set 1 (2006 – 2010), set 2 (2007 – 2011), set 3 (2008 - 2012), and set 4 (2009 – 2013).

Results

To find an optimal portfolio, financiers typically apply a proportion variation calculation to create an efficient frontier line. The Monte Carlo method is a popularly used one. It randomizes the variables according to the constraints for a portfolio. Once a random portfolio is obtained, portfolio risk and expected return are calculated. The portfolio with the highest return at the same risk level will be on the efficient frontier.

Efficient Frontiers Generated by MOPSO and the Monte Carlo Method

Results of portfolio optimization by MOPSO and the Monte Carlo method are shown in Figures 3, 4, 5 and 6. We can see that MOPSO yields better efficient frontiers than does the Monte Carlo method.

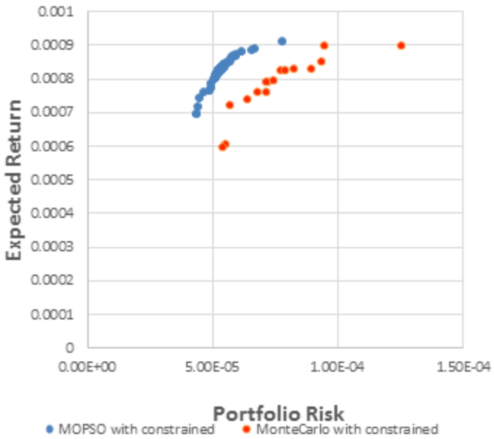

Figure 3: Efficient frontiers by MOPSO and Monte Carlo methods 2006 - 2010

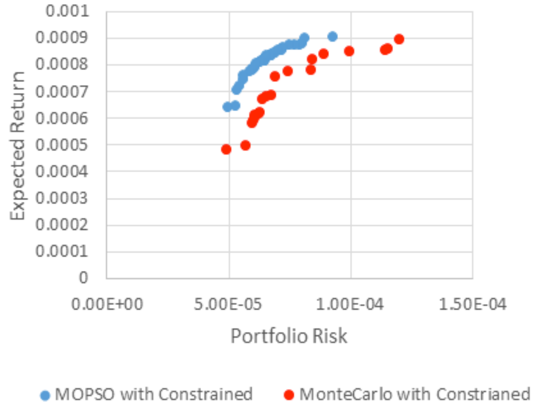

Figure 4: Efficient frontiers by MOPSO and Monte Carlo methods 2007 - 2011

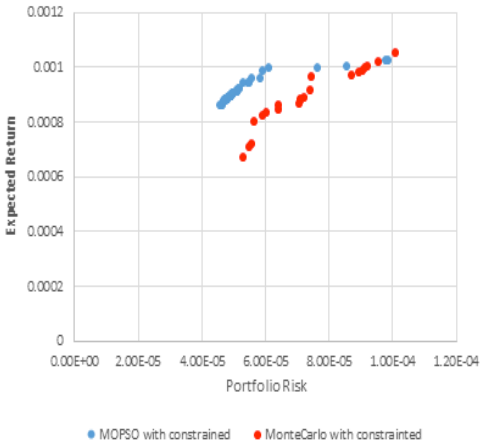

Figure 5: Efficient frontiers by MOPSO and Monte Carlo methods 2009 - 2012

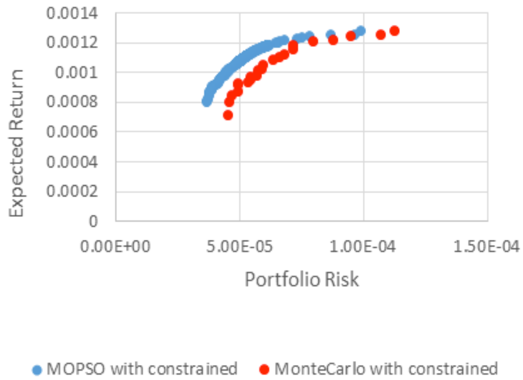

Figure 6: Efficient frontiers by MOPSO and Monte Carlo methods 2009 - 2013

Trading Results

In order to evaluate the trading performance of the proposed method, three types of portfolios by MOPSO are selected, and their returns are calculated which consist of:

1. Portfolio with the highest expected return,
2. Portfolio with the lowest portfolio risk, and
3. Portfolio with the minimum coefficient of variation.

We compare the returns of the 3 types of portfolios generated by MOPSO with the performance of SET, SET50, SET100 and SETHD indices. These indices are the market representatives and used as the standard comparative indices for investment.

Table 1: Total returns of MOPSO and benchmark indices

Total Return							
Portfolio Type / Investment Period	**Minimum Coeff. of Variation**	**Maximum Return**	**Minimum Risk**	**SET**	**SET50**	**SET100**	**SET HD**
2011	15.84%	15.65%	18.27%	3.69%	3.74%	3.23%	NA
2012	45.07%	59.70%	34.49%	40.53%	35.94%	37.69%	26.14%
2013	-1.99%	3.55%	0.04%	-3.63%	-3.53%	-4.07%	-8.57%
2014	15.44%	19.18%	12.25%	19.12%	16.98%	18.18%	8.91%

The results are shown in Table 1. The results show that the portfolio with the highest expected return outperforms all other portfolio types and all indices in every year. The portfolio with the minimum coefficient of variation performs better than all indices in almost every year. Only in 2014 that it generates the return which are 3.68%, 1.54%, and 2.74% less than SET, SET50, and SET100 indices, respectively. The portfolio with the lowest risk performs better than all indices in 2011 and 2013 while it performs worse than the indices (except for SETHD) in 2012 and 2014. Overall, we can see that the proposed technique generate portfolios that perform well in comparisons with standard investment indices.

Conclusion

Portfolio optimization aims to discover an efficient frontier which shows highest expected return on each level of portfolio variance. Due to large variations of variables and constraints, manual portfolio optimization is inefficient. Multi-objective particle swarm optimization is an optimization technique which is suitable for solving numeric optimization and yields high quality results. It is used in this research to construct efficient frontiers for portfolio optimization. The proposed method is evaluated using daily stock total return index gross dividends from Stock Exchange of Thailand between years 2006 and 2014. Its performance in actual trading is compared with the total returns net dividend from SET, SET50, SET100 and SETHD, widely used investment performance indices. The results indicate that the returns from the proposed method are better than the standard indices in most investment periods.

References

Clarke, R., De Silva, H., & Thorley, H. (2002). Portfolio constraints and the fundamental law of active management. *Financial Analysts Journal*. 58(3), 48-66.

Coello, C. A., Toscano, P.G., & Salazar, L.M. (2004). Handling multiple objectives with particleswarm optimization. *IEEE Transactions on Evolutionary Computation*, 8(3), 256–279.

Coello, C. A., Lamont, G. B., & Van Veldhuizen, D.A. (2002). *Evolutionary Algorithms for Solving Multi-Objective Problems*. Boston : Kluwer.

Corazza, M., & Komilov, N. (2009). *Praticle Swarm Intelligence: an alternative approach in Portfolio Optimization*. Facolta di Economia, Universita CA Foscari Venezia.

Deb, K., Agrawal, S., Pratab, A., & Meyarivan, T. (2000). A Fast and Fast and Elitist Multiobjective Genetic Algorithm: NSGA-II. *Proc. Parallel Problem Solving From Nature VI Conf*, 849 -858.

Eberhart, R. C., & Kennedy, J. (1995). Particle Swarm Optimization. *IEEE International Conference on Neural Networks*, 1942-1948.

Goldberg, D. E. (1989). *GeneticAlgorithm in Search, Optimization, and Machine Learning*. Addison Wesley.

Markowitz, H.M. (1952). Portfolio Selection. *The Journal of Finance*, 7 (1), 77-91.

Markowitz, H.M. (1959). *Portfolio Selection. Efficient Diversification of Investments*. NewYork: John Wiley.

Mishra S. K., Panda G., & Meher S. (2009). Multi-objective particle swarm optimization approach to portfolio optimization. In *Nature & Biologically Inspired Computing, NaBIC 2009, World Congress*, 1612-1615.

Moore, J., & Chapman, R. (1999). *Application of Particle Swarm to Multiobjective Optimization*. Department of Computer Science and Software Engineering, Auburn University.

Raquel, C.R., & Naval Jr, P.C. (2005). An Effective Use of Crowding Distance in Multiobjective Particle Swarm Optimization. In *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO-2005)*, 257–264.

Van den Bergh, F. (2006). *An analysis of particle swarm optimizers*. Doctoral dissertation, University of Pretoria.

Contact email: viriyay@sso.go.th