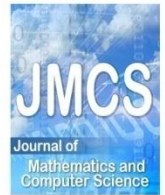


Contents list available at JMCS

Journal of Mathematics and Computer Science

Journal Homepage: www.tjmcs.com



Portfolio Optimization using Particle Swarm Optimization and Genetic Algorithm

Samira Kamali¹

¹*Abadan branch, Islamic Azad University, Abadan, Iran*

¹samirakamali@yahoo.com

Article history:

Received January 2014

Accepted March 2014

Available online March 2014

Abstract

This study basically employs the Markowitz mean–variance model for portfolio selection problem. Since this model is classified as a quadratic programming model there is not any efficient algorithm to solve it. The goal of this study is to find a feasible portfolio with a minimum risk through the application of heuristic algorithm. The two PSO and GA algorithm has been used. The results show that PSO approach is suitable in portfolio optimization.

Keywords: Portfolio optimization, Particle Swarm Optimization, Generic Algorithm.

1 Introduction

Portfolio is conduct to the problem of how to allocate wealth among several assets. The portfolio optimization problems have been one of the important research fields in modern risk management. In generally, an investor always prefers to have the return on their portfolio as large as possible and the risk as small as possible. Markowitz proposed the mean–variance model, which has been regarded as a quadratic programming problem [7, 9]. There are a lot of efforts that have been performed by experts in order to solve and expand Markowitz’s model. In 1956, Markowitz extended the critical line method to solve this quadratic model [8]. Wolfe tried to solve Markowitz’s model by Simplex algorithm [14]. Some heuristic methods based on Genetic Algorithm [1, 2, 3, 7]. Loraschi presented Tabu Search (TS) in the literatures for the standard Markowitz portfolio selection problem [6]. Xu, Chen, and Yang (2007) were the first who proposed a method based on Particle Swarm Optimization (PSO) algorithm to solve the standard Markowitz model [15]. The remainder of this paper is organized as follows. A portfolio selection model which is based on Markowitz’s portfolio selection is described in section 2. In Section

3.4 the characteristics of proposed PSO and GA is explained respectively. Finally Empirical experiments which research the performance of PSO algorithm and also the GA are discussed in Section 5.

2 Portfolio selection problem

In this study we use the famous Markowitz standard model, that model is formulated as follows [7]:

$$\begin{aligned} \min \quad & \sum_{i=1}^N \sum_{j=1}^N x_i x_j \sigma_{ij} \\ \text{s.t.} \quad & \sum_{i=1}^N x_i \mu_i = R^* \\ & \sum_{i=1}^N x_i = 1 \\ & 0 \leq x_i \leq 1, \quad i = 1, \dots, N \end{aligned}$$

Where N is the number of different assets, σ_{ij} is the covariance between returns of assets i and j , x_i is the proportion of asset i in the portfolio, μ_i is the mean return of asset i and R^* is the desired mean return of the portfolio. The objective function minimizes the total variance (risk) associated with the portfolio whilst educational constrained ensure that the portfolio has an expected return, R^* , and the proportions sum up to one. The non- negativity constraint simply means that no short selling is allowed in this problem. Unfortunately, the underlying assumptions of the M-V model are subject to serious criticisms:

- 1) With an increasing number of assets, size of Covariance matrix becomes too big.
- 2) There is no upper limit to the contribution of each asset in the assets set.
- 3) A measure of variance is adopted as a measure of risk. If the assets have normal distributions, this criterion is suitable. But in experimental may this condition not established.

This model is a quadratic programming problem, for solving this model does not exit effective algorithms [1, 2], because of this sake we use the PSO algorithm.

2.1 PSO Algorithm

A particle swarm is a population of particles, where each particle is a moving object that is through the search space and is attracted to previously visited locations with high fitness. In contrast to the individuals in evolutionary computation, particles neither reproduce nor get replaced by other particles. Suppose particle swarm include N particles, $X_i = (x_{i,1}, \dots, x_{i,n})$ and $V_i = (v_{i,1}, \dots, v_{i,n})$ denote the location and velocity of particle i respectively in n -dimensional search space, $P_l = (p_{l,1}, \dots, p_{l,n})$ denote local optimal location of each particle (lbest), $P_{gl} = (p_{g,1}, \dots, p_{g,n})$ denote global optimal location of all particles (gbest).

Velocity update formula of each particle is:

$$v_{i,j}(t+1) = wv_{i,j}(t) + c1rand1(p_{l,j}(t) - x_{i,j}(t)) + c2rand2(p_{g,j}(t) - x_{i,j}(t)) \quad (2.1)$$

for $i = 1, \dots, N$ and $j = 1, \dots, n$.

Where, t is the number of iteration, $c1, c2$ are speeding figure, regulating the length when flying to the most particle of the whole swarm and to the most optimist individual particle, w is inertia weight, $rand1$ and $rand2$ are two independent random number ranged from 0 to 1.

After updating particle velocity, use the following formula to update the location of each particle:

$$x_{i,j}(t+1) = x_{i,j}(t) + v_{i,j}(t) \quad (2.2)$$

for $i = 1, \dots, N$ And $j = 1, \dots, n$.

Then compare new location of each particle with objective value of local optimal location, If the new location is better than the local optimal location, ($lbest$) is updated for the new location, else maintain the original value of ($lbest$) unchanged. Update ($gbest$) according to the new global optimum solution of particle swarm, and continue the evolution of next generation. After a certain number of evolutionary computations by PSO, the particle swarm converges to the global optimum solution, so obtain a global optimal objective value [11].

Here we explain how the PSO algorithm works:

1. Set the parameters.
2. Generate S particles according to the particle formulation
 - 2.1 Initial positions of the particle throughout the search space are defined by initializing each coordinate to a value drawn from the uniform distribution on the interval $[x_{min}, x_{max}]$.
 - 2.2 Generate velocities randomly from $[-1.0, 1.0]$.
3. Repeat for a given maximal number of iterations.
 - 3.1 Evaluate the fitness of each particle based on the pre-computed moments.
 - 3.2 Determine the personal best position $lbest$ visited so far by each particle.
 - 3.3 Determine the local best position $gbesti$ visited so far by the local neighboring particles of each particle.
 - 3.4 Update particles' velocities using (21).
 - 2.5 Update particles' positions using (2.2).

3 GA Algorithm for portfolio optimization

Genetic Algorithms are discussed originally by Goldberg [3]. They are stochastic search techniques based on the principles and policies of natural genetics and selection. The policy of genetic algorithms is that they start with a population of randomly generated particles and evolve towards better solutions by applying genetic operators, such as crossover and mutation, modeled on natural genetic inheritance and Darwinian survival of the fittest [1, 2].

As in [3] the proposed GA for solving portfolio optimization is stated below:

3.1. Population initialization

We initialize with an initial population containing constant number of chromosomes is generated randomly. With regard to portfolio optimization problems, each chromosome represents the weight of individual stock of portfolio and is optimized to reach a possible solution. Parents were chosen by binary contest selection which works by forming two pools of individuals, randomly, each consisting of two individuals drawn from the population randomly. The individuals with the best fitness that are taken from two contest pools are selected to be parents.

3.2. Fitness objective function evaluation

We used mean–variance objective function

$$f = \sum_{i=1}^N \sum_{j=1}^N x_i x_j \sigma_{ij}$$

An evaluation function is formed to evaluate the fitness for each chromosome.

3.3. Reproduction, crossover, and mutation

Children in this GA algorithm are generated by uniform crossover. In uniform crossover two parents have a single child.

3.4. Replacement

In this paper the steady-state population replacement strategy is used. With this strategy each new child is placed in the population as soon as it is generated with wrost objective function value.

4 Computational results

The proposed PSO and the GA algorithm using two sets of data of including 9, 50 stocks are considered. Due to evaluate the time each algorithm requires obtaining the solutions, both algorithms are coded in MATLAB 7.12 and run on a PC with Intel core i5 GHz with 4GB RAM. The data set used in this study are evaluated from contemporaneous stock price for period December 31, 2007, through December 31, 2012 that they were obtained from <http://rdis.ir/MonthReport.asp> and <http://lib.seo.ir/>. For the data set the two algorithms are run 50 times. For GA, the number of iterations is set equal to 1000., the population size equal to 50 and the mutation rate equal to 0.25. For PSO algorithm, the decreasing mutation rate is set from 0.5 down to a value 0.01 and the population size is set equal to 50. Result from solving PSO and GA algorithms with the data set with 9 and 50 stocks are tabulated in tables 1, 2, 3 and 4.

According to result given in these tables, for Best and mean variance, when the number of stocks is 50, we have the better result than the number of stokes is 9 in both algorithms. Based on the results, obviously for the data set of 9 stocks and 50 stocks, best variances found by PSO are better than those found by GA. In addition, PSO can reach better solutions in less amount of time. The application of PSO in solving problems could be very promising. There is a great distinction between PSO and genetic algorithms. In genetic algorithms, chromosomes share the information, which causes the whole community moves gradually into a better area, while in PSO the information is sent out by the best particle which is followed by other individuals to quickly converge to a point. In view of the mentioned reasons, in this paper the PSO algorithm for portfolio optimization by focused on reducing the computation time and better solutions is proposed.

Table 1: result of applying PSO algorithm to 9 stokes data set

Size of portfolio	Expected rate of return	5	7	9
2	Best variance	0.00E+00	0.00E+00	0.00E+00
	Mean variance	3.234E-12	2.124E-12	0.0002
	Standard deviation of variance	5.765E-12	4.453E-12	2.990E-11
	Mean time(s)	10.85469	12.65112	14.93360
4	Best variance	2.623E-8	6.112E-07	2.54E-09
	Mean variance	8.432E-05	3.098E-05	0.00032
	Standard deviation of variance	0.000129	0.001234	0.000345
	Mean time(s)	9.98721	13.87641	18.98765

Table 2: result of applying GA algorithm to 9 stokes data set

Size of portfolio	Expected rate of return	5	7	9
-------------------	-------------------------	---	---	---

2	Best variance	2.65E-08	3.659E-07	1.00E-09
	Mean variance	5.734E-08	4.654E-08	1.887E-11
	Standard deviation of variance	0.009654	0.043256	0.000543
	Mean time(s)	10.89769	12.65432	14.98760
4	Best variance	1.549E-06	2.883E-05	7.213E-04
	Mean variance	0.008765	0.000121	0.000231
	Standard deviation of variance	0.009265	0.000043	0.005674
	Mean time(s)	12.98765	15.98432	23.9732

Table 3: result of applying PSO algorithm to 50 stokes data set

Size of portfolio	Expected rate of return	5	7	9
2	Best variance	0.00E+00	0.00E+00	0.00E+00
	Mean variance	1.6544E-13	2.6543E-13	1.325E-13
	Standard deviation of variance	2.3241E-11	4.3670E-11	2.9876E-11
	Mean time(s)	16.90000	19.65432	15.43669
4	Best variance	0.1000E-13	0.1245E-13	0.1288E-13
	Mean variance	1.54300E-12	2.3266E-12	2.2145E-12
	Standard deviation of variance	3.32670E-10	3.6555E-10	4.21467E-11
	Mean time(s)	19.88432	28.54908	32.96652

Table 4: result of applying GA algorithm to 50 stokes data set

Size of portfolio	Expected rate of return	5	7	9
2	Best variance	0.01E-06	0.00E-06	0.0001E-06
	Mean variance	5.22E-03	2.76E-03	1.876E-03
	Standard deviation of variance	1.2235E-03	2.33421E-03	4.662E-03
	Mean time(s)	10.89769	12.65432	14.98760
4	Best variance	1.8765E-08	2.6543E-08	7.87221E08
	Mean variance	0.00007E-07	0.5431E-06	0.1234E06
	Standard deviation of variance	0.91222E-06	1.0987E-05	3.66541E-05
	Mean time(s)	23.98334	29.08764	36.96650

5 Conclusion

A heuristic approach for the solving for portfolio optimization problem is considered. In order to evaluate the performance of the PSO approach, it is compared to GA approach with Two different (small and large) data sets, i.e. PSO and GA are employed under different portfolio sizes, number of stocks, investor's expected returns. The results clearly show that the PSO approach can achieve better solutions rather than GA approach in a fewer of time.

6 Acknowledgment

The authors would like to thank the council of Abadan Islamic Azad University for finance supporting of this study.

Reference

- [1] T.J.Chang , S.Yang , K.Chang, Portfolio optimization problems in different risk measures using genetic algorithm, *Expert Systems with Applications* ,vol.36, pp.10529–10537(**2009**).
- [2] T.Cura, Particle swarm optimization approach to portfolio optimization, *Nonlinear Analysis Real World Applications*, vol.10, pp. 2396–2406(**2009**).
- [3] D.E.Goldberg, *Genetic Algorithms in Search, Optimization, and Machine Learning*, Addison-Wesley, Reading, MA, (**1989**).
- [4] H.R.Golmakani, M.Fazel, Constrained Portfolio Selection using Particle Swarm Optimization, *Expert Systems with Applications*, vol.38, pp.8327–8335(**2011**).
- [5] R.Khalesi, H.Maleki, A new method for solving fuzzy MCDM problems, *Journal of mathematics and computer Science*, Volume1, Issue4, pp.238-438(**2010**).
- [6] A.Loraschi, A.Tettamanzi, M.Tomassini, C.Svizzero, C.Scientifico, P.Verda, Distributed genetic algorithms with an application to portfolio selection, *Proceedings of the international conference on artificial neural networks and genetic algorithms, ICANNGA95*, pp.384–387(**1995**).
- [7] H.Markowitz, *Portfolio selection, efficient diversification of investments*. New York: Wiley, (**1959**).
- [8] H. Markowitz, The optimization of a quadratic function subject to linear constraints. *Naval Research Logistics Quarterly*, vol. 3, p.111–133(**1956**).
- [9] H.Markowitz, *Portfolio selection*, *Journal of Finance*, vol.7, pp.77–91(**1952**).
- [10] F.Matroud , H.Sadeghi, Solving bi-level programming with multiple linear objectives at lower level using particle swarm optimization, *Journal of mathematics and computer science*, vol.7, pp.221-229(**2013**).
- [11] M.Rostami, M.Kianpour, E.bashardoust, A numerical algorithm for solving nonlinear fuzzy differential equations, *Journal of mathematics and computer Science*, vol. 177, no. 16, pp. 3397-3410(**2007**).
- [12] H.Soleimani, *Portfolio selection using genetic algorithm*, MS Degree Thesis, Amirkabir University of Technology, Industrial Engineering Department, Tehran (**2007**).
- [13] H.Soleimani, H.R.Golmakani, M.H.Salimi, Markowitz-based portfolio selection with minimum transaction lots, cardinality constraints and regarding sector capitalization using genetic algorithm, *Expert Systems with Applications*, 36 pp.5058–5063 (**2009**).
- [14] P.Wolfe, The simplex method for quadratic programming, *Econometrica*, vol.27, pp.382 -398(**1959**).
- [15] F.Xu, W.Chen, L.Yang, Improved Particle Swarm Optimization for realistic portfolio selection, In Eighth ACIS international conference on software engineering, artificial intelligence, networking, and parallel/distributed computing, IEEE Computer Society, pp. 185–190(**2007**).