Clustered stocks weighting with ant colony optimization in portfolio optimization

Cite as: AIP Conference Proceedings **2023**, 020204 (2018); https://doi.org/10.1063/1.5064201
Published Online: 23 October 2018

A. Steven, G. F. Hertono and B. D. Handari

Clustered Stocks Weighting with Ant Colony Optimization in Portfolio Optimization

A. Steven, G. F. Hertono [a)] and B. D. Handari

Department of Mathemtics, Faculty of Mathematics and Natural Sciences (FMIPA), Universitas Indonesia, Depok 16424, Indonesia

[a)] Corresponding author: gatot-f1@ui.ac.id

Abstract. Portfolio Optimization is an optimization problem in finance in which the aim is to maximize the return and minimize the risk of failure within the collection of assets, such as stocks, bonds, etc. This portfolio optimization problem has been of great concern to economic practitioners; since the risk cannot be completely eliminated, an appropriate risk management strategy for choosing and optimizing the portfolio, such that it will fulfill investors' expected risk criteria and returns, will be required. In this paper, we discuss stock portfolio optimization, where stock data based on financial parameters, such as P/E ratio and EPS ratio is converted to score parameters and then clustered by K-means clustering algorithm. After clustering, some stocks will be chosen for the portfolio. The weight of each stock portfolio will be determined such that the objective will be obtained. The Ant Colony Optimization algorithm is used to determine the weight of each stock. The portfolio performance will be evaluated based on some actual datasets. We show important facts that the value of a fitness function is key in choosing the corresponding weighted stock in a stock portfolio and the numerical results also suggest how to reduce the losses of the portfolio.

Keywords: portfolio optimization, stocks, K-means clustering, Ant Colony Optimization

INTRODUCTION

In the modern era, investment becomes an attractive prospect in case gain profit out of unused resources / money. However, investments are also very risky; expecting big profits will be followed by high risk of loss or failure. Investors usually will consider the risk and make their decision based on some interval of risk status. However, some investors will not make decisions with high risk, even if it is expected to offer a big profit. Due to the management of investments, researchers have proposed some approaches and analysis, but further research is needed to determine a better result. One of the more popular kinds of investment is portfolio optimization, which involves selection among the best pool of financial assets that leads to the highest return and the lowest risk (Raei, *et al.* [1]). A commonly utilized method of reducing the risk is diversification. The idea of diversification is to split the investment between various companies. So, if a few of the owned securities underwent a downturn, the others would not, thereby reducing losses. Diversification can be related to industries, countries, types of assets, and so on (Marvin [2]). One of the popular theories in portfolio optimization is Mean-Variance Markowitz by Harry Markowitz.

In this paper, we will concentrate on optimizing portfolios in the stock market. A diversified concept will be used to stabilize the constructing results. The diversified concept is based on various American stocks over last year and the data will be taken from the financial report of each stock. Those include accounting items and financial ratios, since it is assured that the information from quarterly reports or annual reports can influence the price of a stock, especially for unexpected earnings or losses (Magnusson *et al.* [3]). Stock market prediction is an appealing topic, not just for research, but also for commercial applications (Lee, *et al.* [4]).

We propose an approach for portfolio optimization based on clustering and Ant Colony Optimization (ACO). First, we do clustering of stocks with identical characteristics of financial ratios (P/E and EPS). Hence, we can create

Proceedings of the 3rd International Symposium on Current Progress in Mathematics and Sciences 2017 (ISCPMS2017)
AIP Conf. Proc. 2023, 020204-1–020204-6; https://doi.org/10.1063/1.5064201
Published by AIP Publishing. 978-0-7354-1741-0/$30.00

diversified stocks based on this ratio. In every cluster, we expect to have different stock movements and different management, so we can minimize the risk. Second, after clustering, we implement ACO algorithm for every cluster, in order to weight the stocks based on their expected return and variance (risk). Finally, we will compare the result to the previous year's data, and apply the weighting for the current stock market.

PROPOSED ALGORITHM FOR PORTFOLIO OPTIMIZATION

In this paper, we use the two most common attributes of financial ratio, P/E ratio and EPS ratio. P/E ratio is defined as ratio of stock price and net-earning for every stock sheets. Meanwhile, the EPS ratio is defined as net-earning for every stock sheet. There are two types of P/E and EPS ratios: P/E TTM and forward P/E, and basic EPS and diluted EPS. In this paper, we will use P/E TTM and diluted EPS as the parameter. We transform the ratio into the stock score using the following formula (White, et. al. [5]):

$$score = 100\left(\frac{current - min}{\max - min}\right) \tag{1}$$

The *current* value is determined as the present ratio, while *min* and *max* are the minimum value and the maximum value of the ratio, respectively.

The score will be used in the clustering algorithm. By using the score, every stock with the same characteristic can be grouped in a cluster. Hence, each cluster will have different characteristics and movement. We use *K-Means Clustering* algorithm, which is a form of centroid-based clustering. The algorithm begins with *K* starting points as a centroid. Each member will be assigned to the closest centroid, and join the centroid as a cluster member. After all members are in their corresponding clusters, the centroid will be updated as the middle point of the formed cluster, and the iteration is repeated until there are no changes in the cluster members (Tan *et al.* [6]). The reason we chose K-Means Clustering since this algorithm is the most useful clustering algorithm because of its simplicity and its time complexity of *O(n)* (Jain *et al.* [7].

After clustering, we implement Ant Colony Optimization algorithm (Haqiqi and Kazemi [8]) to determine the weight of each stock in the cluster. First, we set the constant of ACO algorithm, *Q*, as an evaporation constant, and construct the pheromone matrices that contain the same fixed non-zero number. We also assign the maximum weight of every stock (*m*), such that the stocks' weight will be in the set *{0,1,2,3,...,m}*. The probability of the weight of each stock is defined as:

$$prob_c = \frac{\tau_{cs}}{\sum_{\forall c} \tau_{cs}} \tag{2}$$

The index *c* represents the weight of the chosen stock *s*. Based on the probability, ants will choose the weight for every stock. Every weight will be divided by the sum of all weights, therefore the sum of the probability is one. After the weights have been constructed, we will be calculated the fitness function as:

$$fitness\ function = \frac{R_p}{\sigma_p} \tag{3}$$

R_p and σ_p are defined as:

$$R_p = \sum_{i=1}^{n} r_i x_i \tag{4}$$

$$\sigma_p = \sqrt{\sum_{i=1}^{n}\sum_{i=1}^{n} x_i x_j \sigma_{ij}} \tag{5}$$

where

R_p : Return of portfolio
r_i : expected return of stock *i*
x_i : weight of stock *i* in portfolio
σ_{ij} : covariance between stock *i* and *j*

In order to maximize the return of portfolio and minimize the covariance of the portfolio, we will maximize the fitness function. We start by updating the pheromone matrices with the equation:

$$\tau_{cs}(t+1) = \tau_{cs}(t)(1-\gamma) + \delta_{cs} \tag{6}$$

Here, $\tau_{cs}(t)$ is defined as pheromone density of stock s with weight c at time t, γ is defined as pheromone evaporation constants, and δ_{cs} is defined as how much the pheromone will be added if the ants choose the weight respective to the stocks. The δ_{cs} will be determined as:

$$\delta_{cs} = Q\,\frac{\frac{R_p}{\sigma_p}}{\frac{R_p^*}{\sigma_p^*}} \tag{7}$$

The value of $\frac{R_p^*}{\sigma_p^*}$ is the maximum fitness function for t iterations. The algorithm will be repeated up to the number of maximum iterations that have been set for ant colony optimization. We will select the optimal fitness function and the corresponding weight as the result of the algorithms. In order to get better solution, we create some weights for this algorithm. For further references, see Haqiqi [8] or Soufiane and Benbouzian [9]. In this paper we used the ACO algorithm since Haqiqi [8] not only shows that the proposed method is useful in finding the optimum portfolio return with the minimum of risk, but also provides better solutions than another meta-heuristic based on genetic algorithms with respect to convergence time and efficiency.

Since we obtain some weights from ACO algorithm, we must decide which is an optimal weight. For each stock weight, we fit them to the current year's stock market data. In order to do that, we first assume that we choose stocks from the current stock data, where the chosen stocks have related weights resulting from the ACO algorithm. Afterwards, we apply the stock weights to all chosen stock in the current year, and compare the monthly expected return to the expected return of the stocks in the current market for related month. Here, we set stock weight in the current market as one unit. Determine the total difference between two expected returns in a year. Then we choose the best stock weight that gives the maximum of total difference (The differences will be taken as the weighted stocks' return minus non-weighting stocks' returns). Finally, for prediction, we apply the best stock weight to the following year's data to check the expected return or the difference between the return and stock market.

CONSTRUCTING RESULTS

Data and Simulation

In this paper, we use the stock that is listed in Standard & Poor's 500 in 2016-2017. In selecting the stock, we choose 19 stocks that have non-negative expected returns in the last 3 years. Hence, the objective function will not be distracted by negative returns. The ratio data will be taken from ycharts.com, while stock prices will be taken from Yahoo Finances. For the test data, we will fit the best weight into stock market in first three month of 2017, and compare it to the stock market. We will use *Python* programming language version 2.7 to simulate the clustering and ACO weighting. We also use *sci-kit learn* module to do data clustering.

Clustering the data

Once we get the data, the first step would be clustering the data based on their financial ratios score. We compare the results of 2 and 4 clusters. The results of clustering data based on 2 and 4 clusters using K-Means algorithm are shown in Fig. 1. The number of colors shows the number of clusters in Fig. 1.

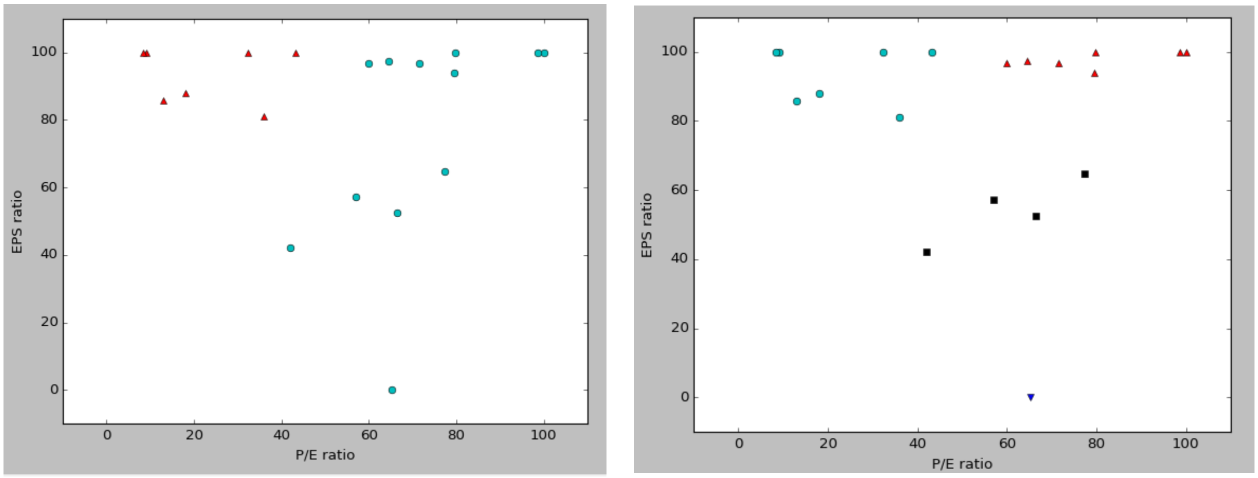

FIGURE 1. K-Means Clustering with (a) 2 clusters and (b) 4 clusters

TABLE 1. Difference of Expected Return Weighted Stock and Its Average Market Return for 2 Clusters and 4 Clusters with the corresponding Fitness Function in the brackets

	Simulation 1		**Simulation 2**	
2 Clusters	**Cluster 1**	**Cluster 2**	**Cluster 1**	**Cluster 2**
	61.26 (0.000168)	57.26 (0.000099)	65.90 (0.000169)	35.17 (0.000211)

	Simulation 1				**Simulation 2**			
4 Clusters	**Cluster 1**	**Cluster 2**	**Cluster 3**	**Cluster 4**	**Cluster 1**	**Cluster 2**	**Cluster 3**	**Cluster 4**
	66.53 (0.0000097)	18.00 (0.000126)	70.64 (0.000159)	0	31.60 (0.000211)	17.96 (0.000145)	56.51 (0.000153)	0

Ant Colony Optimization

In every cluster, we implement ACO algorithm to determine the weight. The constants that we use in this paper are:

$$Q = 1\,; Ants\ number = 100;\ evaporation\ rate = 0.01;\ maximum\ weight = 5;\ repeat\ count = 100;$$

The constant Q is based on Ghafurian and Javadian [10]. The number of ants is defined as how much the ants work in an ACO algorithm. Each ant will have its own weight and fitness function. As the last ants do the algorithm, we take the optimum of the fitness function and save the weight as the best weight for the algorithm. The value of *repeat count* determines how many attempts the ACO algorithm will be simulated. Thus, the ACO repeats as much as repeat count simulations; there are repeat count weights (in this case, we have 100 as the repeat count, so we have 100 best weights too) from the ACO that will be saved in the last simulation. Those weights will be applied to 2016 monthly data for the stock market, and we will find the optimum difference between the expected return of weighted stock and the average stocks' market price, in addition to its fitness function as in Equation 3. The simulation results of two implementations of ACO algorithm (stated as Simulation 1 and Simulation 2) are shown in the Table 1.

The difference between the expected return of weighted stock and its average market return for the 2-cluster case in the first implementation (simulation 1) is 61.26 with its fitness function value is 0.000168 in cluster 1 and 57.26 with its fitness function value is 0.000099 in cluster 2. The highest difference is in cluster 1 in the second implementation (simulation 2) which is 65.90 where the highest fitness function value is 0.000169. In the 4-cluster

TABLE 2. Simulation of maximum return of portfolio with the best stock weight

2 Clusters	January 2017		February 2017		March 2017		Total Difference
	Cluster 1	Cluster 2	Cluster 1	Cluster 2	Cluster 1	Cluster 2	
Simulation 1	-1.43	0.20	-1.05	0.43	-1.23	0.96	-2.12
Simulation 2	0.83	-0.90	0.60	-0.41	1.67	0.63	2.42

4 Clusters	January 2017				February 2017				March 2017				Total Difference
	Cluster Number												
	1	2	3	4	1	2	3	4	1	2	3	4	
Simulation 1	-3.15	0.91	-0.33	0	-4.46	0.80	-0.25	0	-4.09	0.71	0.25	0	-9.61
Simulation 2	-0.81	0.56	-0.27	0	-0.37	1.74	-0.2	0	0.56	1.52	0.2	0	2.93

case, in the first implementation, we get 66.53 and fitness function 0.0000097 in cluster 1, 18.00 and fitness function 0.000126 in cluster 2, 70.64 and fitness function 0.000159 in cluster 3, and 0 for cluster 4 (In case, it only has one member as its cluster, so the difference between its expected return and its average market return should be zero), and in the second implementation, we get 31.60 and fitness function 0.000211 in cluster 1, 17.96 and fitness function 0.000145 in cluster 2, 56.51 and fitness function 0.000153 in cluster 3, and 0 for cluster 4. For comparison, in 2-cluster case, we have mostly similar differences and fitness functions between two implementations in cluster 1, but in cluster 4, we have bigger differences and smaller fitness function in first implementation than in the second implementation. It should be similar in the 4-cluster case. So, we will investigate the impact of the difference of expected return and its average market return, and also its fitness function in the following section.

Apply Weight for Prediction

Finally, we apply the obtained weight related to the result in Table 1 to the first three months of 2017 data. We assume that an investor would select the optimal stock weight that we obtained in December 2016. Thus, the investor should apply the optimum weight for the next 3 months in 2017, and consider the result for the next iteration. In Table 2, we can see the results of the maximum return of portfolio.

As seen in Table 2, in January 2017, simulation 1 has a loss of-1.43 for cluster 1, and a gain of 0.20 for cluster 2. However, in simulation 2, we saw a gain of 0.83 in cluster 1, and a loss of 0.90 in cluster 2, and so on. In this case, total difference means sum of 1-year difference return, the algorithm is not guaranteed to exceed the return of the market price (as shown by the negative values, i.e. -1.43 in January 2017 for 2 cluster in return 1). Since every cluster is independent of the other clusters, we can take the weight that optimizes the results. As an example, in the case of 2 clusters, we can take the weight of Return 2 for Cluster 1 and the weight of Return 1 for Cluster 2. Hence, the total difference is now 4.69 which is obtained by adding 0.83, 0.20, 0.60, 0.43, 1.67 and 0.96. Thus, we have stocks that have positive returns compare to the market return.

However, since we do not know the future, we attempt to find out another strategy in choosing the right weight based on the last year return and its fitness function. First, we consider the case with 2 clusters in Table 1, in which the biggest difference in expected return is 65.60 and its large fitness function is 0.000169 (weight is based on Return 2 and Cluster 1). This case gives a better solution in Table 2 (0.83, 0.60 and 1.67). Unfortunately, this hypothesis does not hold if there are 4 clusters, since from Table 1 the highest expected return is based on weight of Return 1 and Cluster 3 (70.64) with its large fitness function (0.000159), but in Table 2 that weight only gives a maximum return of -0.33.

However, from this simulation, we can conclude that smaller fitness functions can determine a risky consequence. When the value of fitness function is very small, we see two different results: Table 2 with 2 clusters, simulation 1 Cluster 2 has a positive value (0.20), but with 4 clusters case, simulation 1 Cluster 1 has a large negative value (-3.15), i.e. well under the market value. So a small fitness function may be riskier. In the other case, a large fitness function is likely gives the maximum return that around the market price, although not always positive (the fitness function for 4 clusters in Return 2 and cluster 1 has the highest value but the corresponding return value is just -0.37). If we consider the fitness function in (3), we can see that covariance has a significant effect on the fitness function. Thus, if the fitness function is small, it is more likely to be risky in choosing the corresponding weight stock. Hence, we must consider a better fitness function over the difference of weighted return and its market return.

Furthermore, this algorithm also shows the stock weights that can minimize the loss. In our simulation, when there are stocks which have big losses in a particular month, the algorithm has a possibility to reduce the losses by 10%-50%. Thus, it can give choices to the investor who wants to invest in low-risk portfolios. On the other hand, the result can't reach the best return of the month, since the algorithm does not let the investor select the risky stocks which can give large earnings and also large losses.

CONCLUSIONS

In this paper, K-Means clustering algorithm and Ant Colony Optimization can be used to construct the solution of portfolio optimization. We use financial ratios to determine the characteristics of every stocks, and cluster it with K-Means clustering algorithm. Ant Colony Optimization is used to determine the stock weights of each cluster. However, since the numerical results in different clusters give different solutions in valuing the difference in expected returns, this paper cannot determine the best weight. We show important facts that the value of fitness function is more significant compared to the difference between weighted return and average market return, and also how to reduce the losses of the portfolio. In further works, more financial ratios or stock characterized-parameters are needed to improve the K-Means Clustering to picture the stock characteristics in more detail. Moreover, the improvement in ACO performance in choosing the better weight can also be examined, by using more specific fitness functions, or using parameter settings algorithm to get a better result.

ACKNOWLEDGMENTS

This research is fully supported by Hibah PITTA 2017 Number: 632/UN2.R3.1/HKP.05.00/2017.

REFERENCES

1. R. Raei, M. B. Jahromi, and S. Kamalzadeh, International Journal of Computer Applications (IJCA) **103 (11)**, 46 (2014).
2. K. Marvin, Creating Diversified Portfolios Using Cluster Analysis, Independent Work Report Fall, 2015. Available at https://www.cs.princeton.edu/sites/default/files/uploads/karina_marvin.pdf
3. C. Magnusson *et al.,* Inform. Manage. **42**, 561 (2005).
4. A. J. T. Lee, M. C. Lin, R. T. Kao, and K. T. Chen, An Effective Clustering Approach to Stock Market Prediction, in PACIS 2010 Proceedings 54 (Association for Information Systems, Atlanta, 2010). pp. 345-354.
5. G. I. White, A. C. Sondhi, and D. Fried, *The Analysis and Use of Financial Statements* (John Wiley and Sons Inc., Hoboken, 2003).
6. P. N. Tan, M. Steinbach, and V. Kumar, *Introduction to Data Mining*, 1st edition (Pearson Addison Wesley, Boston, 2005).
7. A. K. Jain, M. N. Murty, and P. J. Flynn, ACM Computing Surveys (CSUR) **31**, 264 (1999).
8. K. F. Haqiqi and T. Kazemi, International Journal of Trade, Economics, and Finance (IJTEF) **3**, 148 (2012).
9. S. Sefiane and M. Benbouzian, International Journal of Financial Management (IJFM) **3(4)**, 1 (2013).
10. S. Ghafurian, and N. Javadian, Appl. Soft Comput. **11**, 1256 (2011).