

# Implementation of Density Based Spatial Clustering of Applications with Noise (DBSCAN) and Multi-objective Co-variance Based Artificial Bee Colony (M-CABC) Algorithm on Portfolio Optimization Problem

M. R. Ilham<sup>1</sup>, G. F. Hertono<sup>2</sup>, B. D. Handari<sup>3</sup>  
{mreza.ilham@sci.ui.ac.id<sup>1</sup>, gatot-fl@ui.ac.id<sup>2</sup>, bevina@sci.ui.ac.id<sup>3</sup>}

Department of Mathematics, Universitas Indonesia<sup>1,2,3</sup>

**Abstract.** In order to prepare planned and unplanned needs in the future, investments must be made. In investing, an investor is faced with the problem of determining the number of assets and the proportion of optimal capital in each asset in building an investment portfolio. This problem is called the portfolio optimization problem. In building a portfolio, diversification is required, i.e. combining different characteristics of assets with the aim of reducing investment risk. Clustering can be used as a diversification strategy. The aim of this study is to investigate asset diversification strategy on portfolio using DBSCAN and to select assets and determine the optimal capital proportion of each asset that builds the portfolio using the metaheuristic algorithm M-CABC. DBSCAN is a density-based clustering algorithm designed to form clusters and find noise in arbitrarily shaped data. M-CABC algorithm is a development of the Artificial Bee Colony (ABC) algorithm by adding statistical covariance concepts to accelerate convergence. The assets used in the study are stocks. The implementation is carried out with three different methods: optimization without DBSCAN, optimization with DBSCAN but without noise, and optimization with DBSCAN including noise. The result shows that the size of the proportions of stocks with negative mean returns had an effect on the selection of methods used to obtain the portfolio which carries the least risk.

**Keywords:** Diversification, Portfolio Optimization, DBSCAN, M-CABC.

## 1 Introduction

One of the choices for preparing planned and unplanned needs in the future is through investments. An investment is the employment of funds with the aim of achieving additional income or growth [1]. The benefits of investing are to improve welfare in the future, help reduce inflation pressure, and create profits from sustainable investment instruments [2]. In investing, an investor not only invests in one asset, but also invests in a variety of assets. A collection of assets where the composition is the result of a series of planned or unplanned decisions is called a portfolio [3].

In investing, an investor is faced with the problem of determining the number of assets and the proportion of optimal capital in each asset in building an investment portfolio. This problem is called the portfolio optimization problem [3]. In building a portfolio, diversification is required, i.e. combining different characteristics of assets with the aim of reducing investment risk [4]. One strategy for diversification is clustering [5].

There are various types of clustering methods: K-means Clustering [6], Mean Shift Clustering [7], Expectation-Maximization (EM) Clustering using Gaussian Mixture Models (GMM) [8], Agglomerative Hierarchical Clustering [9], Density Based Spatial Clustering of Applications with Noise (DBSCAN) [10], and so on. DBSCAN is used as a diversification strategy when assets with similar characteristics to each other are in the same cluster so that the selection of assets from different clusters will form a diversified portfolio. DBSCAN is a density-based type clustering method which is designed to discover clusters and noises in an arbitrarily shaped database.

To solve the portfolio optimization problem, a metaheuristic algorithm is used. The following are various metaheuristic algorithms that have been used to solve the portfolio optimization problem: Pareto-Ant Colony Optimization (P-ACO) with Integer Linear Programming (ILP) [11], Multi-objective Genetic Algorithm [12], Particle Swarm Optimization (PSO) [13], Artificial Bee Colony (ABC) [14], and Multi-objective Co-variance Based Artificial Bee Colony (M-CABC) [15]. The M-CABC algorithm is a development of the Artificial Bee Colony algorithm [18], where the M-CABC algorithm is used to select assets and determine the optimal capital proportion of each asset that builds the portfolio.

The portfolio optimization problem in this paper uses portfolio optimization model proposed by Deb, Steuer, Tewari, & Tewari in [16], [17] with DBSCAN as a diversification strategy and the M-CABC algorithm to solve portfolio optimization problems. The portfolio optimization is carried out with three different cases of methods: optimization without DBSCAN, optimization with DBSCAN but without noise, and optimization with DBSCAN including noise. The assets used in this paper are stocks. The results obtained are focused on the portfolio which carries the least amount of risk.

## 2 Literature Review

In this section, we will discuss the Portfolio Optimization Model, Density Based Spatial Clustering of Applications with Noise, and Multi-objective Co-variance Based Artificial Bee Colony Algorithm.

### 2.1 Portfolio Optimization Model

The portfolio optimization model in this paper is by Deb, Steuer, Tewari, & Tewari in [17]. If there are  $n$  available assets and  $x_i$  is the fraction of the capital dedicated to  $i^{\text{th}}$  asset then the portfolio optimization model is as on Eqs. (1) – (6):

$$\text{minimize } f_1(x) = \sum_{i=1}^n \sum_{j=1}^n x_i \sigma_{ij} x_j, \quad (1)$$

$$\text{maximize } f_2(x) = \sum_{i=1}^n r_i x_i, \quad (2)$$

subject to

$$\sum_{i=1}^n x_i = 1, \quad (3)$$

$$\forall i \in \{1, \dots, n\} \quad x_i \geq 0, \quad (4)$$

$$\forall i \in \{1, \dots, n\} \quad x_i = 0 \text{ or } \alpha \leq x_i \leq \beta, \quad (5)$$

$$d_{nz} = \sum_{i=1}^n y_i, \quad y_i = \begin{cases} 1, & \text{if } (x_i > 0) \\ 0, & \text{if } (x_i = 0) \end{cases} \quad (6)$$

$$d_{\min} \leq d_{nz} \leq d_{\max},$$

where  $f_1$  is the variance of the return of the portfolio,  $\sigma_{ij}$  is the correlation between the  $i^{\text{th}}$  and  $j^{\text{th}}$  assets,  $f_2$  is the expected return of the portfolio,  $r_i$  is the mean return of  $i^{\text{th}}$  asset,  $\alpha$  is the lower bound of the capital proportion for every asset,  $\beta$  is the upper bound of the capital proportion for every asset,  $d_{nz}$  is the number of assets invested in a portfolio,  $d_{\min}$  is the minimum number of assets invested in a portfolio, and  $d_{\max}$  is the maximum number of assets invested in a portfolio. Eq. (3) ensures the total proportion of capital in a portfolio is equal to 1, Eq. (4) ensures there is no negative investment, Eq. (5) limits the proportion of capital on each asset, and Eq. (6) limits the number of assets invested in a portfolio.

## 2.2 Density Based Spatial Clustering of Applications with Noise (DBSCAN)

DBSCAN is a density-based type clustering method which is designed to discover clusters and noises in an arbitrarily shaped database. The main idea is for each point of a cluster, the neighborhood around that point with a certain radius (Eps) must at least contain a minimum number of points (MinPts) [10]. DBSCAN requires two parameter values in order to work:

1. Epsilon (Eps): The maximum distance between a point and another point to form a cluster. If the distance between the two points is less than or equal to Eps, the two points are said to be neighbors.
2. Minimum Points (MinPts): The minimum number of points that must exist in the area around a point to form a cluster.

To determine the threshold value of parameter Eps and MinPts, use the sorted k-dist plot discussed in [10]. The DBSCAN algorithm is as follows:

---

### DBSCAN Algorithm.

---

Input: Data, Eps, MinPts

1. Repeat step 2 - 4 until all points already a member of a cluster or noise
2. Select a point in the data set
3. Find Eps-neighborhood of the selected point
4. If the number of members of the selected point Eps-neighborhood less than MinPts then label the point as noise else expand the cluster

Output: Clustered data and noises

---

## 2.3 Multi-objective Co-variance Based Artificial Bee Colony (M-CABC) Algorithm

The M-CABC algorithm uses statistical covariance concept with the aim of accelerating the convergence of solutions and obtaining more precise solutions. It is designed by Kumar & Mishra in [15] and is used to solve the portfolio optimization problem. There are four phases in M-CABC algorithm: the initialization phase, the employed bee phase, the onlooker bee phase, and the scout bee phase.

Every evolutionary multi-objective optimizer has two phases: (i) determining the non-dominated rank of one solution over another and (ii) generating a population from the best (non-dominated) parent solution [15]. In the M-CABC algorithm, NSGA-2 [18] is used for calculating the non-dominated rank of any solution and for generating the non-dominated fronts  $F_1, F_2, \dots, F_n$ . The general M-CABC algorithm is as follows:

---

**M-CABC Algorithm.**

---

Input:  $d$  (number of stocks),  $mx$  (maximum number of iterations)

1. Initialization Phase
2. Perform Non-dominated Sorting on  $P_g$  using NSGA-2 and build non-dominated fronts  $F_1, F_2, \dots, F_n$
3. Repeat step 3.1 - 3.7 until  $g < mx$ 
  - 3.1. Employed bee phase
  - 3.2. Onlooker bee phase
  - 3.3. Scout bee phase
  - 3.4.  $g = g + 1$
  - 3.5.  $hive = \{solution \text{ from employed bee phase } \cup \text{ solution from onlooker bee phase}\}$
  - 3.6. Perform Non-dominated Sorting on  $hive$  using NSGA-2 and build non-dominated fronts  $F_1, F_2, \dots, F_n$
  - 3.7.  $P_g$  is best  $e$  solutions from  $hive$
4. Return  $F_1$

Output:  $P'$  (Pareto-optimal set)

---

Further discussion of each phase on M-CABC algorithm can be seen in the following sub sections (2.4.1 – 2.4.4).

#### 2.4.1 Initialization Phase

Set the parameters needed in the employed bee, onlooker bee, and scout bee phases with Eqs. (7) – (12) as follows:

$$\text{bee hive size} = d * 10, \quad (7)$$

$$e = \text{bee hive size} / 2, \quad (8)$$

$$o = \text{bee hive size} / 2, \quad (9)$$

$$\text{limit} = d * 10, \quad (10)$$

$$g = 0, \quad (11)$$

$$\sigma = 0,5, \quad (12)$$

where bee hive size is the number of bees in the hive, e is the number of employed bees, o is the number of onlooker bees, limit is the iteration limit for scout bees where if there is a bee that does not improve a solution for  $d * 10$  iterations then the solution is re-initialized, g shows the generation of bees, and  $\sigma$  is a magnitude control which is used for controlling the nearness and farness of a child solution from its parent. The initialization phase algorithm is as follows:

---

#### **Initialization Phase Algorithm.**

---

1. Set the parameters with Eqs. (7) - (12)
  2. Generate initial population  $P_g$  of size e
  3. Clean the initial population  $P_g$
- 

---

#### **Cleaning Algorithm.**

---

Input: Unconstrained portfolio  $P = (p_1, p_2, \dots, p_n)$ ,  $d_{\min}$ ,  $d_{\max}$ ,  $\alpha$ ,  $\beta$

1. Calculate nz % number of stocks invested in the portfolio
2. If  $nz < d_{\min}$ , make an investment on other  $nz < d_{\min}$  stocks where no investment has been made with the proportion of capital determined based on random values  $[0,1]$  and set  $nz = d_{\min}$
3. If  $nz > d_{\max}$ , the investment in  $nz - d_{\max}$  number of stocks is eliminated and set  $nz = d_{\max}$
4. For each stock  $i \in \{1, \dots, n\}$ , if  $p_i < \alpha$ , then  $p_i = \alpha$
5. For each stock  $i \in \{1, \dots, n\}$ , if  $p_i > \beta$ , then  $p_i = \beta$
6. Ensure the total proportion of capital in a portfolio is equal to 1
  - 6.1.  $v_{ti} = 0$
  - 6.2. For each stock  $i \in \{1, \dots, n\}$ ,  $v_{ti} = v_{ti} + p_i$
  - 6.3.  $raf = 1 - (nz \cdot \alpha)$
  - 6.4. For each stock  $i \in \{1, \dots, n\}$ , if  $p_i \neq 0$ ,  $p_i = \alpha + \left(\frac{p_i}{v_{ti}} \times raf\right)$

Output: Cleaned portfolio which satisfies all constraints

---

### **2.4.2 Employed Bee Phase**

In the employed bee phase, each solution of the population is updated. The employed bee phase algorithm is as follows:

---

**Employed Bee Phase Algorithm.**

---

1. Set the parameters
    - 1.1.  $\gamma$  is random number,  $\gamma \in [-1, 1]$
    - 1.2.  $\omega$  is random number,  $\omega \in [0, 1]$
  2. Repeat step 2.1 - 2.7 for each  $i^{\text{th}}$  employed bee  $X_i = (x_1^i, x_2^i, \dots, x_d^i)$ ,  $i=1, 2, \dots, e$ 
    - 2.1. Choose any random solution  $X_j$  and any solution  $X_k = (x_{k_1}, x_{k_2}, \dots, x_{k_d})$  from front  $F_1$
    - 2.2. Determine a random dimension (dm) to be updated
    - 2.3.  $X_i^{\text{old}} = X_i$
    - 2.4.  $\text{dfn} = x_{\text{dm}}^i - x_{\text{dm}}^j$  % distance from any random employed bee in the search space
    - 2.5.  $\text{dfb} = b_{\text{dm}} - x_{\text{dm}}^i$  % distance of employed bee from any solution  $X_k$  from  $F_1$
    - 2.6.  $x_{\text{dm}}^i = x_{\text{dm}}^i + \gamma \cdot \text{dfn} + \omega \cdot \text{dfb}$
    - 2.7.  $X_i =$  Pareto optimal selection ( $X_i^{\text{old}}, X_i$ )
  3. Clean the updated solutions
- 

---

**Pareto Optimal Selection Algorithm.**

---

Input: Two solutions  $X_i^{\text{old}} = (x_1^{\text{old}}, x_2^{\text{old}}, \dots, x_d^{\text{old}})$  and  $X_i = (x_1^i, x_2^i, \dots, x_d^i)$

1. If  $(\forall j \in \{1, \dots, k\} f_j(X_i^{\text{old}}) \leq f_j(X_i))$  and  $(\exists j \in \{1, \dots, k\} f_j(X_i^{\text{old}}) < f_j(X_i))$   
%solution  $X_i^{\text{old}}$  is as good as  $X_i$  for all  $k$  objectives function and  
solution  $X_i^{\text{old}}$  is strictly better than  $X_i$  for at least one  
objective  
then return  $X_i^{\text{old}}$   
else return  $X_i$

Output:  $O^*$

---

### 2.4.3 Onlooker Bee Phase

In the onlooker bee phase, new solutions were formed from previously formed fronts  $F_1, F_2, \dots, F_n$ . The onlooker bee phase algorithm is as follows.

---

**Onlooker Bee Phase Algorithm.**

---

1. Repeat step 1.1 - 1.5 for each front  $F_j$ ,  $1 \leq j \leq \lceil \log_2 o \rceil$ 
    - 1.1. Calculate C, B, and D for the front  $F_j$
    - 1.2. Calculate m, the mean of solutions on front  $F_j$
    - 1.3.  $\text{nof} = \lfloor e/2 \rfloor$  % front  $F_j$  contains nof onlooker bees
    - 1.4.  $e = e - \text{nof}$  % remaining onlooker bees for the next Fronts
    - 1.5. Repeat step 1.5.1 - 1.5.2 for each ith onlooker bee  $S_i$ ,  $i \in \{1, \dots, \text{nof}\}$ 
      - 1.5.1. Obtain a random vector  $r_{d \times 1}$ ,  $r_i \in (0, 1)$
      - 1.5.2.  $S_i = m + \sigma B D r$  % the new solution
  2. Clean the formed solutions
- 

To calculate  $C_{d \times d}$  which is a covariance matrix from the selected front ( $F_j$ ) and  $m = (\bar{x}_1, \bar{x}_2, \dots, \bar{x}_d)$ , use Eqs. (13) and (14) respectively.

$$\forall i, j \in \{1, 2, \dots, d\}, C_{ij} = \frac{1}{N-1} \sum_{k=1}^N (x_{ik} - \bar{x}_i)(x_{jk} - \bar{x}_j), \quad (13)$$

$$\forall i \in \{1, 2, \dots, d\}, \bar{x}_i = \frac{1}{N} \sum_{j=1}^N x_{ij}. \quad (14)$$

To obtain  $B_{d \times d}$  and  $D_{d \times d}$ , decompose  $C_{d \times d}$  into  $B_{d \times d}$  and  $D_{d \times d}$  where  $B_{d \times d}$  is a diagonal matrix which contains the eigenvalue of matrix  $C_{d \times d}$  and  $D_{d \times d}$  is a matrix whose columns are eigenvectors corresponding to eigenvalues of matrix  $C_{d \times d}$ .

#### 2.4.4 Scout Bee Phase

In the scout bee phase, if there are solutions that cannot be updated after the number of iterations reaches limit, reinitialize the solution.

### 3 The Proposed Method

This paper proposed the implementation of Density Based Spatial Clustering of Applications with Noise (DBSCAN) and the Multi-objective Co-variance Based Artificial Bee Colony (M-CABC) algorithm on solving the portfolio optimization problem. DBSCAN is used as the diversification strategy to reduce investment risk. The clustering results from DBSCAN are used as input in M-CABC algorithm. The M-CABC algorithm itself is used to select assets and determine the optimal capital proportion of each asset that builds the portfolio. The results obtained are focused on the portfolio which carries the least amount of risk. The portfolio optimization is carried out with three different case of methods which will be discussed on section 3.2.

#### 3.1 Data Sets

The data used in this paper is taken from OR-Library [19]. There are five stock portfolio data used: port1 consists of 31 stocks, port2 consists of 85 stocks, port3 consists of 89 stocks, port4 consists of 98 stocks, and port5 consists of 225 stocks. Each data consists of the mean return and standard deviation of each stock and the correlation value between the stocks.

Each data used in this paper has its own characteristics which are distinguished by the negative mean return proportions/percentages of stocks. The proportions of stocks with negative mean return for each data is shown on Table 1.

**Table 1.** Proportion of Stocks with Negative Mean Return

	port1	port2	port3	port4	port5
	0%	23.5%	4.49%	8.16%	78.6%
Proportion	(0 of 31 stocks)	(20 of 85 stocks)	(3 of 89 stocks)	(8 of 98 stocks)	(177 of 225 stocks)

### 3.2 Case of Methods

In this paper, the portfolio optimization is carried out with three different case of methods to see the effect of clustering with DBSCAN on stock portfolio optimization results with the M-CABC algorithm. The three case of methods are as follows:

1. Stock portfolio optimization without DBSCAN.
2. Stock portfolio optimization with DBSCAN but without noise. The noise resulting from the clustering process is not used in the optimization process.
3. Stock portfolio optimization with DBSCAN including noise (noise which has mean return value above the average of mean return of related data) which added into the cluster with the largest members. This is because cluster with the largest members has the largest cluster weight. So, by adding noise in that cluster, it is expected that the impact of noise can provide a positive influence on the performance of the resulting portfolio.

### 3.3 Parameter Values

The threshold values of the parameter MinPts and Eps used for DBSCAN along with the number of clusters produced for each data in this paper are shown on Table 2.

**Table 2.** The values MinPts, Eps, and Number of Clusters

Data	MinPts	Eps	Number of Clusters
port1	3	0.0026	2
port2	3	0.0035	2
port3	4	0.0020	2
port4	4	0.0019	2
port5	3	0.0015	2

In Table 2, the DBSCAN result for data port1 with MinPts = 3 and Eps = 0.0026 gives 2 clusters, data port2 with MinPts = 3 and Eps = 0.0035 gives 2 clusters, etc.

The parameter values for M-CABC algorithm are different depend on the used data and the method. The parameter values for stock portfolio optimization without DBSCAN are shown in Table 3.

**Table 3.** The Parameter Values for Stock Portfolio Optimization without DBSCAN

No.	Parameter	Value	Information
1.	$[d_{\min}, d_{\max}]$	[10,20]	For port1
2.	$[d_{\min}, d_{\max}]$	[25,35]	For port2, port3, dan port4
3.	$[d_{\min}, d_{\max}]$	[55,65]	For port5

In Table 3, parameter  $[d_{\min} = 10, d_{\max} = 20]$  is used for data port1 in stock portfolio optimization without DBSCAN, etc.

The parameter values for stock portfolio optimization with DBSCAN but without noise and stock portfolio optimization with DBSCAN including noise are different for each cluster of each data. The calculation of parameter values is in the following formula on Eqs. (15) and (16).

$$d_{\min_i} = \left\lfloor \frac{1}{3} \times dc_i \right\rfloor, \quad (15)$$

$$d_{\max_i} = \left\lfloor \frac{2}{3} \times dc_i \right\rfloor, \quad (16)$$

where  $d_{\min_i}$  is the minimum number of stocks taken from cluster-i,  $d_{\max_i}$  is the maximum number of stocks taken from cluster-i, and  $dc_i$  is the number of stocks in cluster-i.

## 4 Experimental Result

Three experiments were conducted on port1, port2, port3, port4, and port5 for each case of methods. The selection of portfolio optimization methods from three different case of methods for each data is based on the least risk resulted in the worst-case scenario, that is the average risk value of three portfolio with the largest risk from all three experiments. The average results of stock portfolio optimization of the three different cases of methods for data port1, port2, port3, port4, and port5 are shown on Tables 4 – 8 respectively.

**Table 4.** The Average Results of Stock Portfolio Optimization from 3 Experiments on port1

Optimization Case of Method	Worst		Best	
	Risk	Return	Risk	Return
Without DBSCAN	0.004722971	0.010813541	0.000822717	0.00414575
With DBSCAN But Without Noise	0.001674622	0.005388993	0.001371028	0.004976016
With DBSCAN Including Noise	0.003822064	0.009524985	0.001029114	0.00464483

In Table 4, the optimization results of data port1, the average results of stock portfolio optimization with the least risk in the worst-case scenario is obtained using stock portfolio optimization with DBSCAN but without noise, whereas in the best-case scenario, it is obtained using stock portfolio optimization without DBSCAN. However, this case of method yields the most risk on the worst-case scenario.

**Table 5.** The Average Results of Stock Portfolio Optimization from 3 Experiments on port2

Optimization Case of Method	Worst		Best	
	Risk	Return	Risk	Return
Without DBSCAN	0.002796855	0.009734295	0.000210545	0.002708512
With DBSCAN But Without Noise	0.001548451	0.006862043	0.000267542	0.002265645
With DBSCAN Including Noise	0.001473559	0.008453718	0.000251448	0.002925431

**Table 6.** The Average Results of Stock Portfolio Optimization from 3 Experiments on port3

Optimization Case of Method	Worst		Best	
	Risk	Return	Risk	Return
Without DBSCAN	0.001511064	0.008195912	0.000264476	0.003567766
With DBSCAN But Without Noise	0.000962329	0.005357208	0.000328674	0.003609678
With DBSCAN Including Noise	0.001373189	0.007650561	0.000331136	0.003887761

**Table 7.** The Average Results of Stock Portfolio Optimization from 3 Experiments on port4

Optimization Case of Method	Worst		Best	
	Risk	Return	Risk	Return
Without DBSCAN	0.002972171	0.008665696	0.000192577	0.00304456
With DBSCAN But Without Noise	0.000591093	0.004887653	0.000230718	0.003090582
With DBSCAN Including Noise	0.001382946	0.007726000	0.000232384	0.003388728

**Table 8.** The Average Results of Stock Portfolio Optimization from 3 Experiments on port5

Optimization Case of Method	Worst		Best	
	Risk	Return	Risk	Return
Without DBSCAN	0.001210947	0.00372993	0.000418549	0.001835981
With DBSCAN But Without Noise	0.001660886	0.003037778	0.000577917	-0.000281151
With DBSCAN Including Noise	0.001231857	0.003234307	0.000468296	0.000621267

From Tables 4 - 8, we have three distinct results. First, on data port1, port3, and port4, which have small percentages of stocks with negative mean returns (port1 has 0%, port3 has 4.49%, and port4 has 8.16%), the average results of stock portfolio optimization with the least risk in the worst-case scenario are obtained using stock portfolio optimization with DBSCAN but without noise. Second, on port5, which has a large percentage of stocks with negative mean returns (port5 has 78.6%), the average results of stock portfolio optimization with the least risk in the worst-case scenario is obtained using stock portfolio optimization without DBSCAN. Third, on port2, which has a percentage of stocks with negative mean returns between the first and second results (port2 has 23.5%), the average results of stock portfolio optimization with the least risk in the worst-case scenario is obtained using stock portfolio optimization with DBSCAN including noise.

## 5 Conclusion

This paper has discussed the implementation of DBSCAN as a diversification strategy and the M-CABC algorithm to solve the portfolio optimization problem. The results obtained based on parameter values and the data used in this paper are as follows: First, if the data has a small percentage of stocks with negative mean returns (3 of 5 used data have 0%, 4.49%, and 8.16% respectively), risk-averse investors are recommended to use stock portfolio optimization with DBSCAN but without noise method, where the worst-case scenario yields the least risk. Second, if the data has a large percentage of stocks with negative mean returns (1 of 5 data used has 78.6%), risk-averse investors are recommended to use stock portfolio optimization without DBSCAN method, where the worst-case scenario yields the least risk. Third, if the data has a percentage of stocks with negative mean returns between the first and second results (1 of 5 data used has 23.5%), risk-averse investors are recommended to use stock portfolio optimization with DBSCAN including noise method, where the worst-case scenario yields the least risk.

**Acknowledgments.** We appreciate the PITTA B grant 2019 No.: NKB-0618/UN2.R3.1/HKP.05.00 /2019 in supporting this research.

## References

- [1] S. S. Kaptan, *Investment management*. Sarup & Sons, 2001.
- [2] Eduardus Tendelilin, "Analisis Investasi dan Manajemen Portofolio, Edisi I, cet. I," *Yogyakarta, BPFE*, 2001, doi: 10.21082/jae.v25n1.2007.55-83.
- [3] E. J. Elton, M. J. Gruber, S. J. Brown, and W. N. Goetzmann, *Modern Portfolio Theory and Investment Analysis, 9th ed*, 9th ed. Hoboken, New Jersey: John Wiley & Sons, Inc., 2014.
- [4] D. C. Francis and D. Kim, *Modern Portfolio Theory*. Hoboken, New Jersey: John Wiley & Sons, Inc, 2013.
- [5] D. León, A. Aragón, J. Sandoval, G. Hernández, A. Arévalo, and J. Niño, "Clustering algoritihms for Risk-Adjusted Portfolio Construction," *Procedia Comput. Sci.*, pp. 1334–1343, 2017.
- [6] J. MacQueen, "Some methods for classification and analysis of multivariate observations," in *Proceedings of the fifth Berkeley Symposium on Mathematical Statistics and Probability*, 1967.
- [7] K. Fukunaga and L. D. Hostetler, "The Estimation of the Gradient of a Density Function, with Applications in Pattern Recognition," *IEEE Trans. Inf. Theory*, pp. 32–40, 1975, doi: 10.1109/TIT.1975.1055330.
- [8] A. P. Dempster, N. M. Laird, and D. B. Rubin, "Maximum Likelihood from Incomplete Data via the EM Algorithm.," *J. R. Stat. Soc. Ser. B*, pp. 1–38, 1977.
- [9] W. H. E. Day and H. Edelsbrunner, "Efficient algorithms for agglomerative hierarchical clustering methods," *J. Classif.*, pp. 7–24, 1984.
- [10] M. Ester, H.-P. Kriegel, J. Sander, and X. Xu, "A density-based algorithm for discovering clusters in large spatial databases with noise," *Knowl. Discov. Data Min.*, pp. 226–231, 1996.
- [11] K. F. Doerner, W. J. Gutjahr, R. F. Hartl, C. Strauss, and C. Stummer, "Pareto ant colony optimization with ILP preprocessing in multiobjective project portfolio selection," *Eur. J. Oper. Res.*, vol. 171, pp. 830–841, 2006.
- [12] P. Skolpadungket, K. Dahal, and N. Harnpornchai, "Portfolio Optimization using Multi-objective Genetic Algorithms," *IEEE Congr. Evol. Comput. (CEC 2007)*, pp. 516–523, 2007.
- [13] T. Cura, "Particle swarm optimization approach to portfolio optimization," *Nonlinear Anal. Real World Appl.*, pp. 2396–2406, 2009.
- [14] W. Chen, "Artificial bee colony algorithm for constrained possibilistic portfolio optimization problem," *Physica A*, pp. 125–139, 2015.
- [15] D. Kumar and K. Mishra, "Portfolio optimization using novel co-variance guided Artificial Bee

- Colony algorithm,” *Swarm Evol. Comput.*, vol. 33, pp. 119–130, 2017.
- [16] K. Deb, *Multi-Objective Optimization using Evolutionary Algorithms*. Chichester: John Wiley & Sons, Ltd, 2001.
- [17] K. Deb, R. E. Steuer, and R. Tewari, “Bi-objective portfolio optimization using a customized hybrid nsga-ii procedure. Evolutionary Multi-criterion Optimization,” *Evol. Multi-criterion Optim.*, pp. 358–373, 2011.
- [18] K. Deb, S. Agrawal, A. Pratap, and T. Meyarivan, “A fast and elitist multiobjective genetic algorithm: NSGA-II,” *IEEE Trans. Evol. Comput.* 6, pp. 182–197, 2002.
- [19] J. E. Beasley, “OR-Library: distributing test problems by electronic mail,” *J. Oper. Res. Soc.*, pp. 1069–1072, 1990.