Applied Mathematical Sciences, Vol. 10, 2016, no. 42, 2087 - 2103
HIKARI Ltd, www.m-hikari.com
http://dx.doi.org/10.12988/ams.2016.64159

Multi-Objective Particle Swarm Optimization and Simulated Annealing in Practice

Ahmad Abubaker

School of Mathematical Sciences
University Sains Malaysia, Penang, Malaysia
&
Department of Mathematics & Statistics, Al Imam Mohammad Ibn Saud
Islamic University, Riyadh, Saudi Arabia

Adam Baharum

School of Mathematical Sciences
University Sains Malaysia, Penang, Malaysia

Mahmoud Alrefaei

Departments of Mathematics & Statistics
Jordan University of Science and
Technology, Irbid, Jordan

Abstract

Several automatic clustering multi-objective algorithms have been proposed in literature to solve the clustering problem. Recently, Multi-Objective Particle Swarm Optimization and Simulated Annealing Algorithm (MOPSOSA) has been proposed. The aim of (MOPSOSA) is to estimate the appropriate number of clusters and appropriately partition a data set into these clusters without the need to know the actual number of clusters. In this work, the efficiency of the MOPSOSA algorithm is studied, which is based on parameters of particles' velocity. Some of the artificial and real-life datasets are used to illustrate the impact of velocity parameters in the efficiency of MOPSOSA algorithm. The

results show that the suitable values of velocity parameters have almost the same range for the datasets used during the experiments.

Keywords: Automatic clustering, Particle swarm optimization, Simulated annealing, Multi-objective optimization

1 Introduction

Clustering [1] is a data mining technique in the field of the unsupervised datasets that is used to explore and understand large collections of data. Clustering unsupervised datasets implies that the structural characteristics of data are unknown and unlabeled. As a data processing technique, clustering is utilized in various fields by dividing and restructuring data to acquire significant and useful knowledge. The clustering process involves grouping the dataset of m objects into k meaningful clusters. Currently, clustering is open research problem. To solve a clustering problem, the number of clusters that fit a dataset must be determined. The objects for these clusters must then be assigned appropriately. The clustering problem can be defined as follows [2]: considering a dataset $P = \{p_1, p_2, \ldots, p_m\}$ with m objects, the clustering of dataset P is the distribution of objects that exits in P into k clusters $C = \{C_1, C_2, \ldots, C_k\}$. C is called the clustering solution, which has the following properties:

1. $\bigcup_{i=1}^{k} C_i = P$,
2. $C_i \bigcap C_j = \phi$, $i \neq j$, $i = 1, \ldots, k$, $j = 1, \ldots, k$,
3. $C_i \neq \phi$, $i = 1, \ldots, k$.

Several automatic clustering algorithms have been proposed in literature. These algorithms [3, 4, 5, 6, 7, 8, 9] can be used in many fields additionally to used in solving clustering problems. Although the clustering of a dataset is the main objective of this study, determining the appropriate number of clusters and conducting an accurate and proper partition of various datasets within these clusters is more important. Abubaker et al. [2] recently proposed a new automatic multi-objective clustering algorithm that can cluster various shapes, sizes, and overlapping datasets; this algorithm is called hybrid multi-objective particle swarm optimization with simulated annealing (MOPSOSA). In solving clustering problems, velocity parameters are the most important parameters in controlling the efficiency and accuracy of the MOPSOSA algorithm.

The ability to obtain highly accurate solutions is the most important objective for any clustering process. This study discusses the efficiency of the MOPSOSA algorithm on the basis of the change in velocity parameters of the

particles. Furthermore, this study discusses the suitable range of these parameters. These parameters control on the movement of particles in three directions, namely, current, particle, and swarm directions. Artificial and real-life datasets are used to illustrate the effect of velocity parameters in the efficiency of the MOPSOSA algorithm, while trying to determine the suitable choice of these parameters to obtain the best performance in clustering.

This paper is organized as follows; In Section 2, we present the steps of the MOPSOSA algorithm; Section 3 presents the probability velocity parameters; In Section 4, we discuss the efficiency of the new proposed algorithm based on the change in the three important probability velocity parameters; Finally, concluding remarks are given in Section 5.

2 MOPSOSA Algorithm

The MOPSOSA algorithm [2] integrates the advantages of fast calculation and convergence in particle swarm optimization with the capability to evade local solutions in simulated annealing. The MOPSOSA algorithm starts with the K-means technique [10] to create the initial particle positions (swarm). Then, the multi-objective particle swarm optimization (MPSO) is implemented, where all particles in the swarm are launched through the search space by following the current optimum particles to search for the best solution. During the search, the multi-objective simulated annealing (MOSA) is used if no change occurs in the position of a particle, or if the particle moves to a bad position. Each iteration, uses the idea of sharing fitness [11] to update the repository that contains Pareto optimal solutions. Figure 1 shows the conceptual diagram of the mechanism of the MOPSOSA algorithm with input and output datasets. The MOPSOSA algorithm performs automatic clustering according to three cluster validity indices (objective functions): the DaviesBouldin index (DB-index) [12], which is based on Euclidean distance; the symmetry-based index (Sym-index) [6], which is based on point symmetry distance; the connectivity-based index (Conn-index) [13], which is based on short distance.

In the MOPSOSA algorithm, each particle position is a clustering solution. The particle position X_i^t and velocity V_i^t are presented as vectors with m components $X_i^t = (X_{i1}^t, X_{i2}^t, \ldots, X_{im}^t)$ and $V_i^t = (V_{i1}^t, V_{i2}^t, \ldots, V_{im}^t)$ respectively at iteration t, $i = 1, \ldots, n$, where m is the number of data objects, and n is the number of particles (swarm size). The position particle X_i represents a clustering solution that is described using label-based integer encoding [14]. The position component $X_{ij}^t \in \{1, \ldots, k_i^t\}$ represents the cluster number of the j^{th} object in the i^{th} particle and $V_{ij}^t \in \{0, \ldots, k_i^t\}$ represents the motion of the j^{th} object in the i^{th} particle, where $k_i^t \in \{k_{min}, \ldots, k_{max}\}$ is the number of clusters related to particle i at iteration t, where k_{min} and k_{max} are the minimum and the maximum number of clusters respectively, the default value

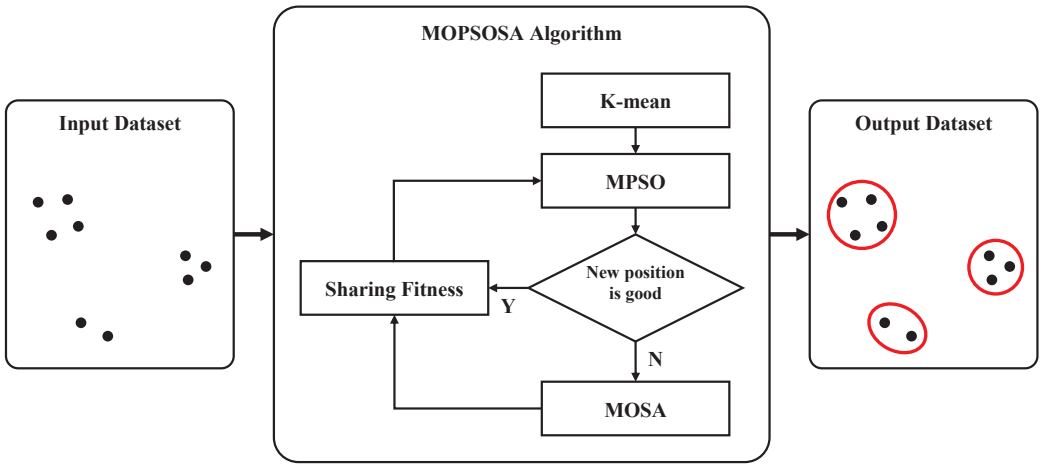


Figure 1: Diagram of the mechanism of MOPSOSA algorithm.

of k_{min} is 2, and k_{max} is $\sqrt{m}+1$ unless manually specified [15]. The best previous position of the i^{th} particle at iteration t is represented as $XP_i^t = (XP_{i1}^t, XP_{i2}^t, \ldots, XP_{im}^t)$. In each iteration t and for each particle i, XP_i^t is updated by non-dominant criteria. This update has three cases. If XP_i^t is dominated by X_i^{t+1}, then $XP_i^{t+1} = X_i^{t+1}$. If X_i^{t+1} is dominated by XP_i^t, then $XP_i^{t+1} = XP_i^t$. If neither XP_i^t nor X_i^{t+1} dominates the other, then one of them is chosen randomly as XP_i^{t+1}. The leader position that is chosen from the repository of Pareto set for the i^{th} particle at iteration t is represented by $XG_i^t = (XG_{i1}^t, XG_{i2}^t, \ldots, XG_{im}^t)$. A leader XG_i^t is selected from the repository according to the distance between XG_i^t and X_i^t, where the particle in the repository that are nearest to X_i^t is selected as the leader. For the particle i at iteration t, the new velocity V_i^{t+1} is calculated as follows:

$$V_i^{t+1} = \left(W \otimes V_i^t\right) \oplus \left[\left(R_1 \otimes \left(XP_i^t \ominus X_i^t\right)\right) \oplus \left(R_2 \otimes \left(XG_i^t \ominus X_i^t\right)\right)\right] \quad (1)$$

where W, R_1, and R_2 are vectors (velocity parameters) of $m-$dimension with values of either 0 or 1 that are randomly generated with probabilities of w, r_1, and r_2, respectively. The difference operation $\ominus$ is defined as follows; $(XP_i^t \ominus x_i^t) = (\lambda p_{i1}^t, \ldots, \lambda p_{im}^t)$ and $(XG_i^t \ominus X_i^t) = (\lambda g_{i1}^t, \ldots, \lambda g_{im}^t)$, λp_{ij}^t and λg_{ij}^t are defined as follows;

$$\lambda p_{ij}^t = \begin{cases} XP_{ij}^t & \text{if } X_{ij}^t \neq XP_{ij}^t \\ 0 & \text{otherwise} \end{cases} \quad (2)$$

and,

$$\lambda g_{ij}^t = \begin{cases} XG_{ij}^t & \text{if } X_{ij}^t \neq XG_{ij}^t \\ 0 & \text{otherwise} \end{cases} \quad (3)$$

The multiplication operator $\otimes$ of the two vectors $A = (a_1, \ldots, a_m)$ and $B = (b_1, \ldots, b_m)$ is defined as $A \otimes B = (a_1 b_1, \ldots, a_m b_m)$. The merging operator $\oplus$ of the two vectors A and B is defined as $A \oplus B = (c_1, \ldots, c_m)$, c_i is computed as follows:

$$c_i = \begin{cases} a_i & \text{if } a_i \neq 0 \text{ and } b_i = 0 \\ b_i & \text{if } a_i = 0 \text{ and } b_i \neq 0 \\ a_i \text{ or } b_i \text{ randomly} & \text{if } a_i \neq 0 \text{ and } b_i \neq 0 \\ 0 & \text{otherwise} \end{cases} \tag{4}$$

For particle i at iteration t, the new position X_i^{t+1} is generated from the new velocity as follows:

$$X_i^{t+1} = \begin{cases} V_i^{t+1} & \text{if } V_i^{t+1} \neq 0 \\ rand & \text{otherwise} \end{cases} \tag{5}$$

where $rand$ is an integer random number in $[1, k_i^t + 1]$, and $k_i^t + 1 \leq k_{max}$.

MOPSOSA Algorithm:

Step 1: Initialize swarm size n, number of iteration $iter$, k_{min}, k_{max}, T_0, and $t = 0$.

Step 2: Use the K-means method to generate the initial X_i^t, $V_i^t = 0$, and $XP_i^t = x_i^t$, $i = 1, \ldots, n$.

Step 3: Compute $f_1(X_i^t), \ldots, f_{\mathcal{S}}(X_i^t)$. Then fill the repository of non-dominated XP_i^t, $i = 1, \ldots, n$.

Step 4: Select the leader XG_i^t from the repository. Renumber XP_i^t and XG_i^t based on X_i^t, $i = 1, \ldots, n$.

Step 5: Compute X_i^{new} and V_i^{new}, $i = 1, \ldots, n$ by using Equations (1) and (5).

Step 6: The validity of X_i^{new}, $i = 1, \ldots, n$ is checked, and the position validation process is applied if it is not valid.

Step 7: Compute $f_1(X_i^{new}), \ldots, f_{\mathcal{S}}(X_i^{new})$ for the candidate next position X_i^{new}, and $f_1(X_i^t), \ldots, f_{\mathcal{S}}(X_i^t)$ for the current position X_i^t, $i = 1, \ldots, n$.

Step 8: Implement a dominance check for X_i^{new}, $i = 1, \ldots, n$. If X_i^{new} is non-dominated by X_i^t then $X_i^{t+1} = X_i^{new}$ and $V_i^{t+1} = V_i^{new}$. Otherwise $X_i^{t+1} = X_i^{MOSA}$ and $V_i^{t+1} = V_i^{MOSA}$, $i = 1, \ldots, n$, where X_i^{MOSA} is the position and V_i^{MOSA} the velocity which are returns from the MOSA technique.

Step 9: Find the new XP_i^{t+1}, $i = 1, \ldots, n$.

Step 10: Update the repository.

Step 11: Set $t = t + 1$, if $t > iter$ then the algorithm is stopped, and the repository contains the Pareto solutions, otherwise go to step 4.

3 Velocity Parameters

In solving clustering problems, the velocity parameters are the most important factors that control the efficiency and accuracy of the MOPSOSA algorithm. In the MOPSOSA algorithm, the velocity of each particle was modified by Equation (1). The velocity parameters W, R_1, and R_2 are generated randomly with probabilities w, r_1, and r_2, respectively. No clear formulation is available to define the probability velocity parameters w, r_1, and r_2. Moreover, w controls the motion of particles in the current direction. r_1 controls the motion of particles in the direction of the best position of the same particle, and r_2 controls the motion of particles in the direction of the best position obtained by swarm particles. As shown in Figure 2, these directions affect the movement of particles. Thus, these directions affect both the movement of particles and the performance of the MOPSOSA algorithm. Given these observations, this study investigates the effect of these parameters on one another and on the algorithm. For this purpose, the MOPSOSA algorithm was implemented with different values of the velocity parameters by using several datasets. The probability velocity parameters w, r_1, and r_2 are discussed below.

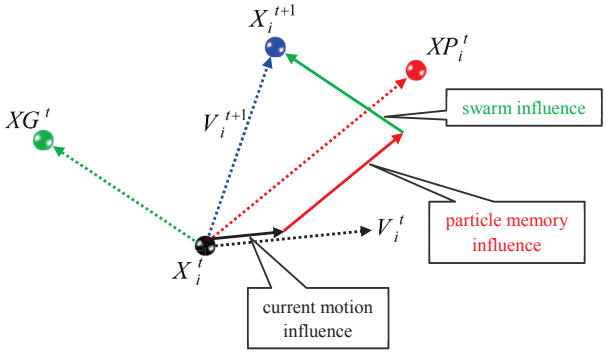


Figure 2: Influences on the motion of particle i at iteration $t + 1$

1. The probability velocity parameter w (inertia weight) generates the vector W, which contributes to the generation of the subsequent velocity of the particle on the basis of the current velocity. Therefore, this parameter

plays an important role in the performance of the MOPSOSA algorithm. A small w value facilitates searching in new areas, that is global exploration. By contrast, a large w value tends to facilitate searching in a current area, that is local exploration. The use of an appropriate w value ensures balance between global and local exploration capabilities. Thus, we can obtain convergence to the optimal solution with high accuracy in few iterations.

Shi and Eberhart [16] have introduced the concept of inertia weight in the original particle swarm optimization (PSO), they have also illustrated the impact of this parameter on the performance of PSO. There are some of the strategies in literature that can be utilized to adjust the inertia weight. Shi and Eberhart [16] have shown that the inertia weight which is linearly decreases during run time provides a good result. In another strategy, the inertia weight was randomly set between 0.5 and 1 [17]. Also increased the inertia weight during run time to improve the performance of the PSO algorithm [18]. However, the optimal value of inertia weight varies from case to case [19].

2. The probability velocity parameter r_1 generates the vector R_1, which controls the quantity of the cognitive component. The quantitative component is the individual memory of the best position of particles. Therefore, a large r_1 value affects the particle's movement, which is attracted to the best position of a particle. Thus, the particle tends to return to the best positions or situations. This finding indicates that a large r_1 increases the possibility of local exploration of particles, whereas a small r_1 value weakens the particle's ability for local exploration.

3. The Probability velocity parameter r_2 generates the vector R_2, which controls the quantity of the social component. Conceptually, the social component is a criterion that particles seek to attain. Therefore, a large r_2 value implies that each particle is attracted to the position that is best for all particles. This situation increases the particle's ability for global exploration, but weakens the efficiency for local exploration. By contrast, a small r_2 value weakens the particle's efficiency for global exploration.

4 Experimental Study

This experimental study presents the effects of changes in velocity parameters on the performance of the MOPSOSA algorithm. The effects of velocity parameters on the MOPSOSA algorithm were examined using the 6 artificial and 3 real-life datasets presented in Table 1 [2]. We used the MOPSOSA algorithm with one parameter of w, r_1 , and r_2 within the range $[0.01, 1]$ at 0.01

increment, whereas the other parameters are fixed. In addition, the swarm was formed from 50 particles, the number of iterations was 100, and the minimum and maximum numbers of clusters were 2 and $\sqrt{m}+1$, respectively, as shown in Table 2. To evaluate the clustering quality, we employed an external criterion called F-measure [20], which takes two clustering solutions i.e., one is the known true clustering and the other is the final solution obtained from the algorithm, and provides the value of the similarity (within interval [0,1]) between them. Let T and C be two clustering solutions, $T = \{T_1, \ldots, T_{k_T}\}$ be the true solution, and $C = \{C_1, \ldots, C_{k_C}\}$ be the solution to be measured, where k_T and k_C are the number of clusters for the solutions T and C, respectively. The F-measure of class T_i and cluster C_j are defined as:

$$F\left(T_i, C_j\right) = \frac{2 * P\left(T_i, C_j\right) * R\left(T_i, C_j\right)}{P\left(T_i, C_j\right) + R\left(T_i, C_j\right)} \tag{6}$$

where $P\left(T_i, C_j\right)$ is called a Precision, which is the ratio of the points in cluster C_j that exist in class T_i, and is defined as $P\left(T_i, C_j\right) = n_{ij}/|C_j|$, where n_{ij} is the number of points that exist in both T_i and C_j, and $|C_j|$ is the number of points in cluster C_j, $R\left(T_i, C_j\right)$ is called a Recall, which is the ratio of the points in class T_i that are in cluster C_j, denoted by $R\left(T_i, C_j\right) = n_{ij}/\left|T_i\right|$. Meanwhile, the F-measure of solutions T and C are constructed as follows:

$$F\left(T, C\right) = \sum_{i=1}^{k_T} \frac{|T_i|}{m} \max_{C_j \in C} \left\{F\left(T_i, C_j\right)\right\} \tag{7}$$

The higher values of $F\left(T, C\right)$ are the better values and the optimal value of $F\left(T, C\right)$ is 1.

Table 1: Description of the artificial and real-life data sets.

Dataset	**Points**	**Dimension**	**Clusters**
Sph_4_3	400	3	4
Sph_9_2	900	2	9
Pat2	417	2	2
Sizes5	1000	2	4
Spiral	1000	2	2
Fourty	1000	2	40
Iris	150	4	3
Cancer	683	9	2
LiverDisorder	345	6	2

Table 2: Parameter used in the MOPSOSA algorithm to obtain the best values for w, r_1, and r_2.

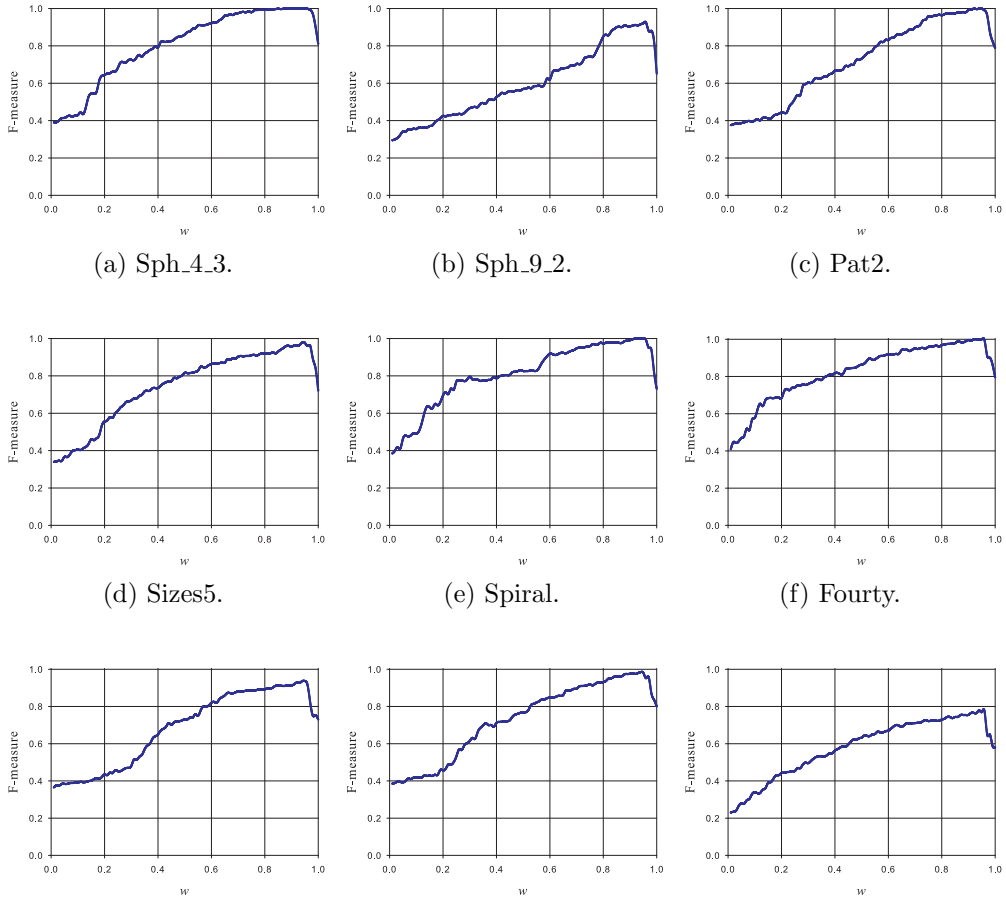
Description	**Parameters**	**Value**
Swarm size	n	50
Number of iteration	*iter*	100
Minimum number of clusters	k_{min}	2
Maximum number of clusters	k_{max}	$\sqrt{m}+1$

4.1 The Probability Velocity Parameter w

To study the effect of the probability velocity parameter w on the MOPSOSA algorithm, we selected 100 different values for w, that are $0.01, 0.02, 0.03, \ldots, 1$ when clustering 9 real life and artificial datasets by using the MOPSOSA algorithm. These values were used to obtain the appropriate number of clusters and appropriately partitioned the datasets into such clusters. Moreover, we determined the proper interval for w, which achieved high values for the F-measure criterion for each dataset used in this experiment. We also determined the best interval for w for all datasets. Assuming that the other velocity parameters are fixed, we considered $r_1 = 0.90$ and $r_2 = 0.90$.

Figures 3 show the relationship between w and the F-measure criterion by implementing the MOPSOSA algorithm on 6 artificial and 3 real life datasets. When w is increased to a certain value, the value of the F-measure criterion increases and then decreases thereafter, which means as w increases, the clustering accuracy of the MOPSOSA algorithm improves (although this phenomenon occurs up to a certain point only); then, an adverse relationship is formed between w and the F-measure. A small w value slightly maintains the momentum of the previous velocity. This phenomenon indicates a large chance for the particle to move into a new search area as well as a decrease in its chance of staying in the current search area. As w approaches to 0, there is no relation between previous velocity and next velocity of the particle over time, resulting in the movement of particle into a new search area; thus, the particle loses its ability to search in the current area. The particles cannot move back to the optimum, and the swarm diverges. In addition, when w approaches to 1, the particles' velocity evanesce in each step, and all particles move regardless of the previous velocity, leading to limited searching; thus, the MOPSOSA algorithm requires a large number of iterations to reach the optimal solutions or the particle becomes trapped in the local search area.

In this study, we have considered the best values of the F-measure (BVF) for clustering a dataset as the interval between the maximum value of the F-measure (MVF) minus 0.01 and the MVF, i.e., $\text{BVF} = [\text{MVF} - 0.01, \text{MVF}]$, or $\text{MVF} - 0.005 \pm 0.005$. MVF was obtained from clustering a dataset using the



(a) Sph_4_3. (b) Sph_9_2. (c) Pat2.

(d) Sizes5. (e) Spiral. (f) Fourty.

(g) Iris. (h) Cancer. (i) Liver Disorder.

Figure 3: The effect of w on the F-measure criterion of 9 artificial and real-life datasets, where $r_1 = 0.90$, $r_2 = 0.90$, and the 100 values for w, are $0.01, 0.02, \ldots, 0.99, 1$.

MOPSOSA algorithm at $w = 0.01, 0.02, \ldots, 1$. Table 3 shows the appropriate range for w to achieve BVF; this range also determines the correct number of clusters. The proper cluster number and BVF can be obtained from a certain interval of w values.

These results show that the probability velocity parameter w affects the accuracy of the MOPSOSA algorithm. Furthermore, the following conclusions can be drawn; First, as w increases, the F-measure criterion also increases up to a certain point; then, it decreases as w continues to increase. Second, a large w value (approaching 1) and a small w value (approaching 0) provide weak results when solving clustering problems. Third, the proper interval of w for all datasets used in this study is $[0.94, 0.95]$. Finally, w values may be

modified to improve the results for other datasets.

Table 3: A proper number of clusters, BVF, and an appropriate range of w obtained from MOPSOSA for 9 datasets, where $r_1 = 0.90$ and $r_2 = 0.90$.

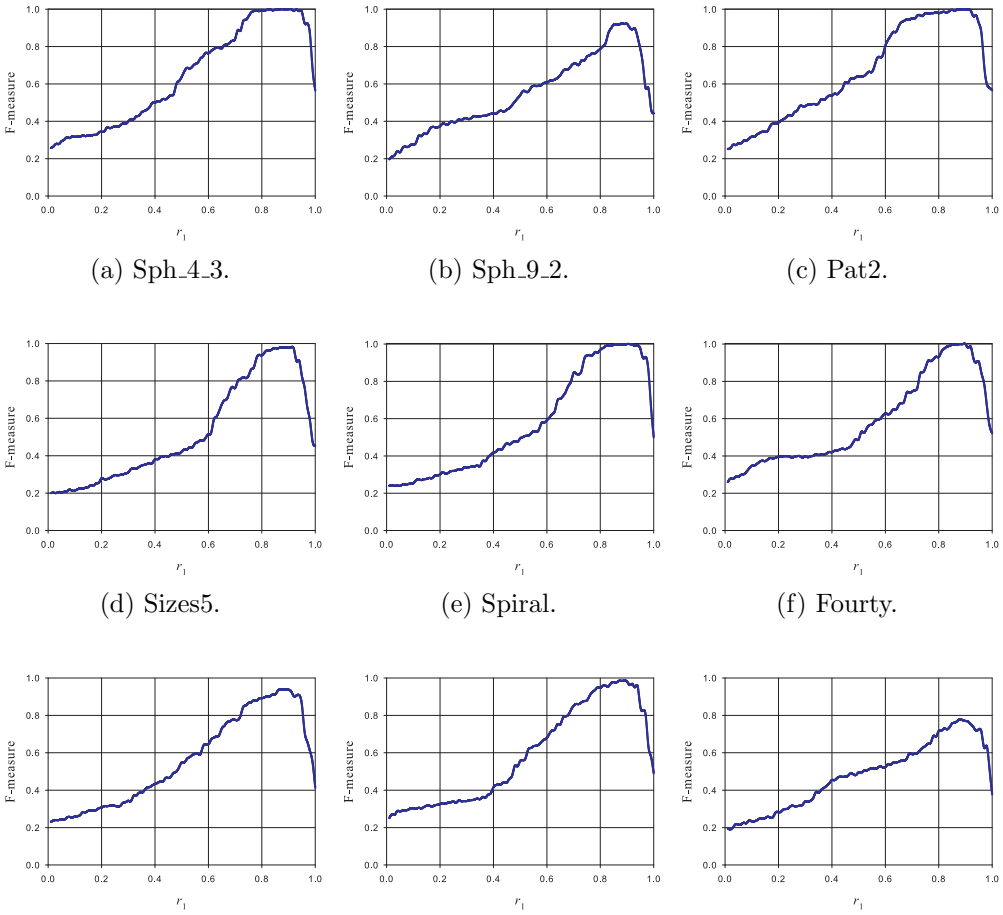
Dataset	**# Clusters**	**BVF**	$\boldsymbol{w}$
Sph_4_3	4	0.995 ± 0.005	[0.76,0.97]
Sph_9_2	9	0.920 ± 0.005	[0.94,0.96]
Pat2	2	0.995 ± 0.005	[0.90,0.96]
Sizes5	4	0.974 ± 0.005	[0.94,0.95]
Spiral	2	0.995 ± 0.005	[0.90,0.96]
Fourty	40	0.995 ± 0.005	[0.90,0.96]
Iris	3	0.936 ± 0.005	[0.92,0.96]
Cancer	2	0.980 ± 0.005	[0.90,0.95]
LiverDisorder	2	0.775 ± 0.005	[0.94,0.96]

4.2 The Probability Velocity Parameter r_1

To study the effect of the probability velocity parameter r_1 on the MOPSOSA algorithm, 100 different values for r_1, that are $0.01, 0.02, 0.03, \ldots, 1$ were selected to cluster 9 real-life and artificial datasets using the MOPSOSA algorithm. Each r_1 value was used to implement the MOPSOSA algorithm to appropriately divide the datasets into a proper number of clusters. The proper interval for r_1 was determined to achieve BVF in clustering each dataset used in this experiment, and determine the best interval for r_1 for all datasets. Assuming that the other velocity parameters are fixed, we considered $w = 0.95$ and $r_2 = 0.90$.

Figure 4 shows the relationship between r_1 and the F-measure criterion by implementing the MOPSOSA algorithm in the 6 artificial and 3 real life datasets. The value of the F-measure criterion increases as r_1 increases up to certain point, and then begin to decreases thereafter. Moreover, by increasing r_1 the accuracy of clustering by the MOPSOSA algorithm increases, although this phenomenon occurs to a certain point only; then, an adverse relationship is formed between r_1 and the F-measure. This phenomenon occurs because as r_1 approaches 1, the particle velocity increases over time toward the best previous position of a particle; thus, the particle cannot easily move into a new search area. The particles are possibly trapped in local solutions; therefore, the swarm diverges. In addition, a small r_1 value slightly maintains the momentum of the best previous position of a particle, indicating a higher chance for the particle to move into a new search area. The MOPSOSA algorithm thus loses its ability for an in-depth search in the local area of the particles. When r_1 approaches

0, then in each step, all particles move regardless of the previous condition, thereby making the movement of all particles depend on one particle.



(a) Sph_4_3. (b) Sph_9_2. (c) Pat2.

(d) Sizes5. (e) Spiral. (f) Fourty.

(g) Iris. (h) Cancer. (i) Liver Disorder.

Figure 4: The effect of r_1 on the F-measure criterion of 9 artificial and real-life datasets, where $w = 0.95$, $r_2 = 0.90$, and the 100 values for r_1, are $0.01, 0.02, \ldots, 0.99, 1$.

Table 4 shows the appropriate range for r_1 to achieve BVF; this range also determines the correct number of clusters. The proper cluster number and BVF are obtained from a certain interval of r_1 values.

This study shows that r_1 affects the accuracy of the MOPSOSA algorithm. Furthermore, we have the following observations; First, as r_1 increases the F-measure criterion also increases up to a certain point, and then decreases as r_1 continues to increase. Second, a large r_1 value (approaching 1) and a small r_1 value (approaching 0) provide weak results when solving clustering problems. Third, the proper r_1 interval for all datasets used in these study is $[0.87, 0.90]$.

Finally, r_1 may be modified to improve the results of the other datasets.

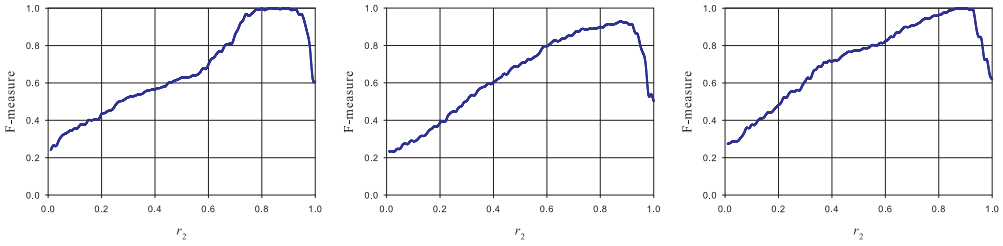
Table 4: The proper number of clusters, BVF, and an appropriate range of r_1 obtained from MOPSOSA of 9 datasets, where $w = 0.95$ and $r_2 = 0.90$.

Dataset	**# Clusters**	**BVF**	$\boldsymbol{r_1}$
Sph_4_3	4	0.995 ± 0.005	[0.77,0.94]
Sph_9_2	9	0.921 ± 0.005	[0.85,0.90]
Pat2	2	0.995 ± 0.005	[0.84,0.93]
Sizes5	4	0.975 ± 0.005	[0.84,0.92]
Spiral	2	0.995 ± 0.005	[0.83,0.94]
Fourty	40	0.995 ± 0.005	[0.84,0.91]
Iris	3	0.936 ± 0.005	[0.86,0.91]
Cancer	2	0.981 ± 0.005	[0.85,0.90]
LiverDisorder	2	0.774 ± 0.005	[0.87,0.91]

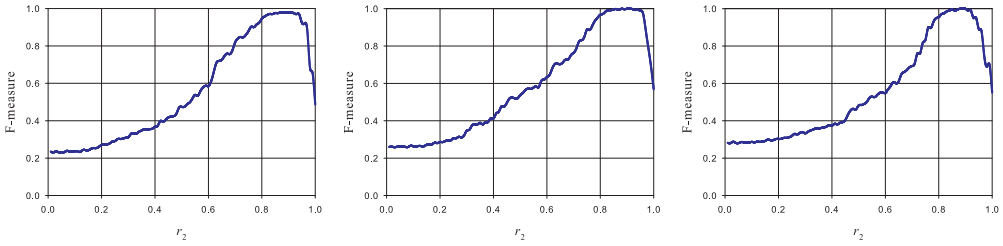
4.3 The Probability Velocity Parameter r_2

To study the effects of r_2 on the MOPSOSA algorithm, 100 different values for r_2 that are $0.01, 0.02, 0.03, \ldots, 1$ were selected to cluster 9 real-life and artificial datasets using the MOPSOSA algorithm. Each r_2 value was used to implement the MOPSOSA algorithm to appropriately divide the datasets into a proper number of clusters. The proper interval of r_2 was determined to achieve BVF for clustering each dataset used in this experiment, as well as to determine the best interval of r_2 for all datasets. The other velocity parameters are fixed at $w = 0.95$ and $r_1 = 0.90$.

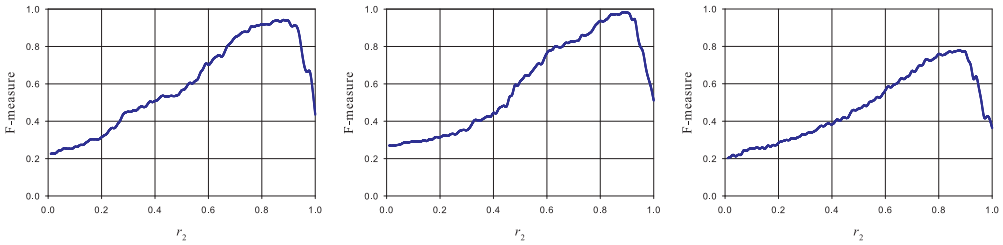
Figure 5 shows the relationship between r_2 and the F-measure criterion by implementing the MOPSOSA algorithm in the 6 artificial and 3 real life datasets. As r_2 increases up to a certain point, the value of the F-measure criterion increases, and then decreases thereafter. Moreover, by increasing r_2 the performance of the MOPSOSA algorithm improves, although this phenomenon occurs up to a certain point only; then, an adverse relationship is formed between r_2 and the F-measure. This phenomenon occurs when r_2 approaches 1, which causes the particle velocity to increases over time toward the best previous position of all particles, thereby making the movement of all particles depend on one particle. Therefore, the particles are possibly trapped in local solutions, and the swarm diverges. In addition, a small r_2 value slightly maintains the momentum of the best global position. The MOPSOSA algorithm thus loses its ability to search in the global area of the particles, indicating a high chance for the particle to become trapped in a local solution. Furthermore, if r_2 approaches 0, then in each step, all particles move regardless of previous swarm causing each particle to move within the local area.



(a) Sph_4_3. (b) Sph_9_2. (c) Pat2.



(d) Sizes5. (e) Spiral. (f) Fourty.



(g) Iris. (h) Cancer. (i) Liver Disorder.

Figure 5: The effect of r_2 on the F-measure criterion of 9 artificial and real-life datasets, where $w = 0.95$, $r_1 = 0.9$, and the 100 values for r_2, are $0.01, 0.02, \ldots, 0.99, 1$.

Table 5 shows the appropriate range for r_2 to achieve BVF; this range also determines the correct number of clusters. The proper cluster number and BVF are obtained from a certain interval of r_2 values.

This study shows that r_2 affects the accuracy of the MOPSOSA algorithm, where r_2 increases from 0.01 to 1 at 0.01 increment. Furthermore, we conclude the following. First, as r_2 increases, the F-measure criterion also increases up to a certain point and then decreases as r_2 continues to increase. Second, a large r_2 value (approaches 1) and a small r_2 value (approaches 0) provide weak results when solving the clustering problem. Third, the proper interval of r_2 for all datasets used in these study is $[0.87, 0.90]$. Finally, r_2 values may be modified to improve the results of other datasets.

Table 5: Proper number of clusters, BVF, and appropriate range of r_2 obtained from MOPSOSA of 9 datasets, where $w = 0.95$ and $r_1 = 0.9$.

Dataset	**# Clusters**	**BVF**	$\boldsymbol{r_2}$
Sph_4_3	4	0.995 ± 0.005	[0.78,0.93]
Sph_9_2	9	0.922 ± 0.005	[0.86,0.90]
Pat1	3	0.995 ± 0.005	[0.82,0.91]
Pat2	2	0.995 ± 0.005	[0.85,0.92]
Sizes5	4	0.974 ± 0.005	[0.83,0.94]
Spiral	2	0.995 ± 0.005	[0.81,0.95]
Fourty	40	0.995 ± 0.005	[0.86,0.92]
Iris	3	0.937 ± 0.005	[0.84,0.90]
Cancer	2	0.979 ± 0.005	[0.87,0.91]
LiverDisorder	2	0.773 ± 0.005	[0.84,0.90]

5 Conclusion

The effect of the particle's velocity parameters W, R_1, and R_2, on the ability of the MOPSOSA algorithm to solve the clustering problem was scrutinised. The velocity parameters W, R_1, and R_2 are the vectors of the $m-$dimension with 0 or 1 component are generated randomly by the probabilities w, r_1, and r_2, respectively. In this study, 100 distinct values for each probability velocity parameter, were chosen to cluster 9 real-life and artificial datasets using the MOPSOSA algorithm. It is shown that the performance of the MOPSOSA algorithm and the quality of clustering are affected by all the probability velocity parameters, where there exists a correlation between probability velocity parameters and the F-measure criterion. As the probability velocity parameters increase, the F-measure criterion increases correspondingly up to a specific value whereupon it would start to decrease. Hence, the efficiency of the MOPSOSA algorithm may be enhanced by raising the probability velocity parameters w, r_1, and r_2, though this is only true up to a specific value, after which, the positive effect of increasing the probability velocity parameters becomes a negative effect, instead. The suitable values for probability velocity parameters w, r_1, and r_2 to produce the vectors W, R_1, and R_2 are found in the ranges $[0.94, 0.95]$, $[0.87, 0.90]$, and $[0.87, 0.90]$, respectively.[]

References

[1] L. Kaufman and P.J. Rousseeuw, *Finding Groups in Data: An Introduction to Cluster Analysis,* John Wiley and Sons, 2009.

[2] A. Abubaker, A. Baharum and M. Alrefaei, Automatic clustering using multi-objective particle swarm and simulated annealing, *PloS One,* **10** (2015), e0130995. http://dx.doi.org/10.1371/journal.pone.0130995

[3] H. Masoud, S. Jalili and S. Hasheminejad, Dynamic clustering using combinatorial particle swarm optimization, *Applied Intelligence,* **38** (2013), 289 - 314. http://dx.doi.org/10.1007/s10489-012-0373-9

[4] S. Bandyopadhyay and U. Maulik, Genetic clustering for automatic evolution of clusters and application to image classification, *Pattern Recognition,* **35** (2002), 1197 - 1208. http://dx.doi.org/10.1016/s0031-3203(01)00108-x

[5] J. Handl and J. Knowles, An evolutionary approach to multiobjective clustering, *IEEE Transactions on Evolutionary Computation,* **11** (2007), 56 - 76. http://dx.doi.org/10.1109/tevc.2006.877146

[6] S. Bandyopadhyay and S. Saha, A point symmetry-based clustering technique for automatic evolution of clusters, *IEEE Transactions on Knowledge and Data Engineering,* **20** (2008), 1441 - 1457. http://dx.doi.org/10.1109/tkde.2008.79

[7] S. Saha and S. Bandyopadhyay, A symmetry based multiobjective clustering technique for automatic evolution of clusters, *Pattern Recognition,* **43** (2010), 738 - 751. http://dx.doi.org/10.1016/j.patcog.2009.07.004

[8] Y. Liu, X. Wu and Y. Shen, Automatic clustering using genetic algorithms, *Applied Mathematics and Computation,* **218** (2011), 1267 - 1279. http://dx.doi.org/10.1016/j.amc.2011.06.007

[9] S. Saha and S. Bandyopadhyay, A generalized automatic clustering algorithm in a multiobjective framework, *Applied Soft Computing,* **13** (2013), 89 - 108. http://dx.doi.org/10.1016/j.asoc.2012.08.005

[10] J. MacQueen, Some methods for classification and analysis of multivariate observations, *Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability,* **1** (1967), 281 - 297.

[11] D. Goldberg and J. Richardson, Genetic algorithms with sharing for multimodal function optimization, *Proceedings of the Second International Conference on Genetic Algorithms on Genetic Algorithms and their Applications,* (1987), 41 - 49.

[12] D.L. Davies and D.W. Bouldin, A cluster separation measure, *IEEE Transactions on Pattern Analysis and Machine Intelligence,* (1979), 224 - 227. http://dx.doi.org/10.1109/tpami.1979.4766909

[13] S. Saha and S. Bandyopadhyay, Some connectivity based cluster validity indices, *Applied Soft Computing,* **12** (2012), 1555 - 1565. http://dx.doi.org/10.1016/j.asoc.2011.12.013

[14] E.R. Hruschka, R.J.G.B. Campello, A.A. Freitas, A.C,P.L.F. De Carvalho, A survey of evolutionary algorithms for clustering, *IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews,* **39** (2009), 133 - 155. http://dx.doi.org/10.1109/tsmcc.2008.2007252

[15] S.L. Yang, Y.S. Li, X.X. Hu and R.Y. PAN, Optimization study on k value of K-means algorithm, *Systems Engineering-Theory & Practice,* **2** (2006), 97 - 101.

[16] Y. Shi and R. Eberhart, A modified particle swarm optimizer, *1998 IEEE International Conference on Evolutionary Computation Proceedings, IEEE World Congress on Computational Intelligence,* (1998), 69 - 73. http://dx.doi.org/10.1109/icec.1998.699146

[17] R.C. Eberhart and Y. Shi, Tracking and optimizing dynamic systems with particle swarms, *Proceedings of the 2001 Congress on Evolutionary Computation,* **1** (2001), 94 - 100. http://dx.doi.org/10.1109/cec.2001.934376

[18] Y.L. Zheng, L.H. Ma, L.Y. Zhang and J.X. Qian, On the convergence analysis and parameter selection in particle swarm optimization, *International Conference on Machine Learning and Cybernetics,* **3** (2003), 1802 - 1807. http://dx.doi.org/10.1109/icmlc.2003.1259789

[19] Y. Shi and R.C. Eberhart, Parameter selection in particle swarm optimization, Chapter in *Evolutionary Programming VII,* Springer Berlin Heidelberg, 1998, 591 - 600. http://dx.doi.org/10.1007/bfb0040810

[20] B.C.M. Fung, K. Wang and M. Ester, Hierarchical document clustering using frequent itemsets, Chapter in *Proceedings of the 2003 SIAM International Conference on Data Mining,* **3** (2003), 59 - 70. http://dx.doi.org/10.1137/1.9781611972733.6

Received: May 10, 2016; Published: June 24, 2016