EMITTER International Journal of Engineering Technology
ISSN: 2443-1168, Vol. 8, No. 1, June 2020, pp. 86~101
DOI : 10.24003/emitter.v7i2.498

An Improved Crow Search Algorithm for Data Clustering

Vivi Nur Wijayaningrum[1], Novi Nur Putriwijaya[2]

[1]Department of Information Technology, Politeknik Negeri Malang, Indonesia
vivinurw@polinema.ac.id
[2]Faculty of Information and Communication Technology,
Institut Teknologi Sepuluh Nopember, Indonesia
novinputriw@gmail.com

Received March 27, 2020; Revised April 19, 2020; Accepted May 11, 2020

Abstract

Metaheuristic algorithms are often trapped in local optimum solutions when searching for solutions. This problem often occurs in optimization cases involving high dimensions such as data clustering. Imbalance of the exploration and exploitation process is the cause of this condition because search agents are not able to reach the best solution in the search space. In this study, the problem is overcome by modifying the solution update mechanism so that a search agent not only follows another randomly chosen search agent, but also has the opportunity to follow the best search agent. In addition, the balance of exploration and exploitation is also enhanced by the mechanism of updating the awareness probability of each search agent in accordance with their respective abilities in searching for solutions. The improve mechanism makes the proposed algorithm obtain pretty good solutions with smaller computational time compared to Genetic Algorithm and Particle Swarm Optimization. In large datasets, it is proven that the proposed algorithm is able to provide the best solution among the other algorithms.

Keywords: awareness probability, clustering, crow search algorithm, metaheuristic algorithm

1. INTRODUCTION

[illegible] several [illegible] used to define groups of a given data set so that each group consists of entities that are similar, otherwise entities between the groups is different. These algorithms typically used for data analysis without the learning process of the test data [1]. Research on clustering has been widely applied to solve problems in various fields, such as stock trend prediction [2], the disease identification [3][4], and the identification of disaster types [5]. Clustering belongs to the NP-hard problem because the principle of the groups division is based on the minimization of the dissimilarity in the group and maximizing the distance between groups [6][7]. Therefore, various approaches are used to solve the clustering problems. Such approaches could be a result of the modification of an algorithm or the result of hybridization of two or more certain algorithms. Some researchers also have proposed the use of mathematical models to solve the clustering problems [8].

Lately, evolutionary algorithm, that is algorithm that works by mimicking the biological principle of evolution and natural selection, considered as an alternative that can be used to solve the clustering problems because it is suitable for handling global optimization problems. Evolutionary algorithm is effective, reliable, and adaptive, and can produce solutions that approach the optimal solution through evolution. Several studies proved that the algorithm has capability to address complex problems [9]. Advantages of the use of evolutionary algorithms to solve the clustering problems is the ability to handle a local optimum by recombination and comparison of candidate solutions simultaneously, and allows for perfecting the final solution [7]. While the weakness of evolutionary algorithms is the computational time required is relatively high, the parameter is difficult to determine for example the population size, the type of selection and crossover operators, and the fitness function [10]. Similar to the evolutionary algorithm, swarm intelligence is also a field of research that has gained enormous popularity in these days [11]. Swarm intelligence is inspired by the collective intelligence that arises from the behavior of a group of social insects, such as bees, ants, birds, and others. Some researchers have implemented swarm intelligence to solve various NP hard problems, one of which is clustering.

In this study, an Improved Crow Search Algorithm (ICSA) is used for data clustering with the aim of balancing exploration and exploitation during the process of finding solutions so that the results of clustering provided are global optimum solutions. The results of data clustering using ICSA will be compared with other metaheuristic algorithms to find out how well the solutions obtained from ICSA.

2. RELATED WORKS

Evolutionary algorithms and swarm intelligence have been widely used by several researchers for clustering. This is due to its ability to solve complex problems by finding the best solution through an iterative process during its development [12][13]. Clustering technique using genetic algorithm (GA) was proposed by Maulik and Bandyopadhyay [14] to find cluster centers in the feature space so that the metric similarity of the cluster formed is optimal. The proposed algorithm gives better results compared to k-Means on the seven datasets used at the time of testing.

Zhao, et al [15] solved the clustering problem using k-Means based on Particle Swarm Optimization (PSO). The use of PSO is based on the weakness of k-Means in determining a good cluster center so that the performance of the algorithm can be affected. The working principle of the proposed algorithm is that each particle forms a clustering solution that contains several cluster centers. Then, the cluster center is used to divide the data using the k-Means process. The proposed algorithm execution time is less than the k-Means algorithm.

Nasiri and Khiyabani [16] proposed a metaheuristic algorithm for clustering, namely Whale Optimization Algorithm (WOA) which works based on the principle of humpback whales in foraging. Algorithm performance has been tested using various datasets and compared with several other clustering algorithms. The results of the evaluation using the intra-cluster distance function and standard deviation show that WOA can be applied well to solve the clustering problem.

Another metaheuristic algorithm that has the ability to produce good solutions is Crow Search Algorithm (CSA), which works by mimicking the behavior of crows in finding food. Based on the results of experiments using six constrained engineering design problems, CSA provides a better solution compared to PSO and GA [17]. In addition, CSA provides a more accurate solution compared to other search methods [18].

3. ORIGINALITY

CSA has the ability to promise in solving complex optimization problems by providing better solutions compared to other search algorithms. However, in the search process, sometimes CSA is trapped in the local optimum area and fails to reach the global optimum solution [19]. This is due to the low efficiency of global exploration and the lack of balance between the exploration and exploitation phases [14]. Modifications to the solution update mechanism can be made to improve the global exploration process in the solution search space [20].

In this study that uses ICSA for data clustering, there is a renewal of the solution update mechanism proposed to improve the solutions provided by conventional CSA in order to provide a global optimum solution. In conventional CSA, the solution update mechanism is performed by directing a search agent to follow another search agent that is chosen randomly so that the search agent will approach the other search agent in the search space to explore new areas. A search agent chosen randomly to be followed by another search agent does not always have a good solution. If the search agent that is followed has a solution that is far from the optimum solution, then the other search agent who follow it will also tend to move away from the optimum solution that should have been obtained. This makes the algorithm difficult to obtain a global optimum solution. Therefore, this study proposes the probability of choosing the agent to follow. A search agent can follow another search agent with the best fitness value or randomly chosen search agent. The probability of choosing the search agent is used to maintain the quality of the solution of the exploration process. If too many search agents follow the best search agents, then this may lead to premature convergence due to lack of diversity of solutions. However, if too many search agents follow a randomly chosen search agent, then the process of finding a solution takes longer to reach the global optimum solution. Therefore, the probability of choosing a search agent for a solution update mechanism proposed in this study is expected to make search agents more quickly reach the global optimum solution in the search space.

4. SYSTEM DESIGN

The application of ICSA to data clustering is illustrated as a group of crows that states several possible solutions to solve the problem. Every possible solution is called a search agent. The purpose of ICSA is to get the best search agent position that produces the best solution based on the objective functions that have been given.

4.1 Encoding

In the search area, there are a number of *N* search agents (flock size) so that they can be stated in Equation 1.

$$S = (s_1, s_2, \ldots, s_N) \tag{1}$$

In Equation 1, *S* is a vector that contains a set of solutions stated by several search agents. To resolve data clustering using ICSA, each search agent represents the center of the cluster by the number of *K*. *K* is the number of pre-determined clusters. Each cluster has an attribute with a number of *Y*. Thus, each search agent s_i is arranged as in Figure 1.

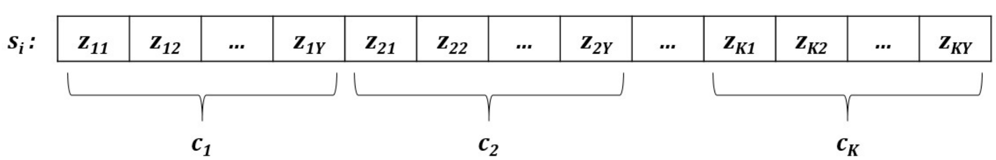

Figure 1. Encoding of a search agent

In Figure 1, there are a number of *K* cluster for the *i*-th search agent, i.e. $c_1, c_2, \ldots, c_K$. Each cluster has an attribute with the number of *Y*, so the center of the cluster is defined as $z_{11}, z_{12}, \ldots, z_{KY}$. The possible solution of a search agent arranged as in Figure 1 is the position of the crow that will be updated every iteration until it reaches maximum interaction ($iter_{max}$). Each search agent has a memory of the position of its hiding place, symbolized by m_i.

4.2 Fitness Functions

The quality of partitions from the result of grouping can be measured using a cluster validity index. The measurement used is intra-distance of clusters, which is calculating the distance between cluster centers and vector data from the same cluster [21]. A clustering result is considered good if the resulting intra-distance of clusters value is small.

The fitness function is used to measure the quality of solutions provided by a search agent. A high fitness value indicates that the solution is good to be used as the final solution [22]. Thus, in the case of finding the minimum intra-distance of clusters value, the fitness function can be expressed in Equation 2.

$$fitness = \frac{1}{\sum_{j=1}^{Y}\sum_{i=1}^{N}\sum_{h=1}^{K} w_{ih}\left|x_{ij} - z_{hj}\right|} \tag{2}$$

In Equation 2, x_{ij} is the i-th data vector and the j-th attribute, while z_{hj} is the h-th cluster center and the j-th attribute. The value of w_{ih} is obtained through the following conditions:

$$w_{ih} = \begin{cases} 1 & \text{if } x_{ij} \text{ is assigned to cluster } h \\ 0 & \text{otherwise} \end{cases}$$

4.3 Proposed Algorithm

Pseudocode of ICSA for data clustering proposed in this study is described in Algorithm 1. Algorithm 1 is the workflow of the Crow Search Algorithm with an improvised mechanism highlighted in gray.

To maintain the quality of the solution of the exploration process, the novelty proposed in this study is the probability of choosing a search agent that is followed which consists of the search agent with the best fitness value or randomly chosen search agent, in this case shown in Algorithm 1, lines 9 to 11. The conventional CSA only uses the solution update mechanism using a randomly selected search agent to be followed by other search agents. The mechanism used in conventional CSA causes the algorithm to have difficulty in achieving the global optimum solution because the search agent will tend to move away from the optimal solution if the search agent that is followed has a poor solution. Therefore, the probability of selecting a search agent proposed in this study is used to accelerate the search agents to achieve the global optimum solution, while maintaining the diversity of solutions provided by each search agent.

An update of the awareness probability value is also performed on each search agent by considering the most recent fitness value achieved shown in Algorithm 1, lines 18 to 21. If the best fitness value of a search agent is no better than the best fitness value previously obtained during several iterations, then the value of awareness probability will be updated. Thus, each search agent has different awareness probability values according to their respective abilities. Equation 5 is used in the mechanism of awareness probability update which adopts the concept of Dynamic Awareness Probability [23].

Algorithm 1. An Improved Crow Search Algorithm

Load data samples
Initialize parameters of ICSA (flock size N, $iter_{max}$, flight lenth fl, and awareness probability AP)
Initialize the threshold value thr to update the AP value
Initialize the position of N search agents, each containing k randomly cluster centers
Define an awareness probability AP for all search agent
While $iter < iter_{max}$ **do**
  For each search agent i **do**
    Calculate the fitness function using Equation 2
    If a random value $rand > 0.5$
      Randomly choose one of the search agent to follow j
    Else if $rand \leq 0.5$
      Choose the best search agent to follow j
    End if
    If a random value $r_j \geq AP$, update the position using Equation 3

$$x_{i\_new} = x_i + r_i \times fl_i \times \left(m_j - x_i\right) \quad (3)$$

    Else if $r_j < AP$, update the position using Equation 4

$$x_{i\_new} = \text{a random position of search space} \quad (4)$$

    End if
    Check the feasibility of the new positions x_{i_new}
    Calculate the fitness function using Equation 2
    Update the memory of search agent
    If the current best fitness value ≤ previous best fitness value
      Increment counter ct
    End if
    If $ct == thr$, update the AP value using Equation 5

$$AP = 0.9 \times \frac{\text{best fitness value}}{\text{worst fitness value}} + 0.1 \quad (5)$$

    End if
  End for
End while

4.4 Validity Indices

Cluster validation method is used to evaluate the results of clustering. This evaluation aims to determine the quality of partitions from the results of clustering produced by the proposed algorithm. This study uses the Root-mean-square Standard Deviation (RMSSTD) and R-squared (RS) as an index of the validity of the partition quality from clustering results.

RMSSTD is an index measuring the homogeneity of clusters formed, by calculating the square root of the sample variance of all attributes [24]. This measurement index only measures the compactness of the cluster produced. Compactness measures the proximity of each data point that is clustered in the same cluster. Each data point in a cluster should be interconnected by sharing features that describe a certain pattern [25]. The RMSSTD calculation is shown in Equation 6.

$$RMSSTD = \sqrt{\frac{\sum_{\substack{i=1..k \\ j=1..p}} \sum_{c=1}^{n_{ij}} \left(x_c - \bar{x}_{ij}\right)^2}{\sum_{\substack{i=1..k \\ j=1..p}} \left(n_{ij} - 1\right)}} \tag{6}$$

RS is used to determine whether there are significant differences between objects in different clusters and high similarity between objects in the same cluster. RS is calculated using Equation 7, which is the complement of the ratio of the number of squares between objects in different clusters with the total number of squares. This type of measurement only considers the separation between clusters. Separation measures how different a cluster is from another [25].

$$RS = \frac{\left(\sum_{j=1..p} \left(\sum_{c=1}^{n_j} \left(x_c - \bar{x}_j\right)^2\right)\right) - \left(\sum_{\substack{i=1..k \\ j=1..p}} \left(\sum_{c=1}^{n_{ij}} \left(x_c - \bar{x}_{ij}\right)^2\right)\right)}{\sum_{j=1..p} \left(\sum_{c=1}^{n_j} \left(x_c - \bar{x}_j\right)^2\right)} \tag{7}$$

In Equation 6 and Equation 7, k is the number of clusters, p is the number of attributes, n_{ij} is the amount of data in the p-th attribute and k-th cluster, x_c is the value of the c-th data, $\bar{x}_j$ is the average value of data that is in the j-th attribute, and $\bar{x}_{ij}$ is the average value of data that is in the j-th attribute and the i-th cluster.

5. EXPERIMENT AND ANALYSIS

Parameter testing is performed to get the optimal parameters so that the solution produced by ICSA is global optimum. By using these optimal parameters, the performance of ICSA is compared with several other algorithms through several datasets so that the performance of the proposed algorithm can be known.

5.1 Parameter Testing of ICSA

To get optimal results, ICSA has several parameters that must be determined properly through the testing process. Parameter testing conducted in this study consisted of flock size, number of iterations, flight length, and awareness probability. These tests are performed using the Iris dataset.

The flock size is tested using several test scenarios with a range of values from 100 to 600. Each test scenario is run 20 times. Other parameters that are set to a constant value in this test are the number of iterations is 500, a flight length is 2, and the awareness probability is 0.1. The flock size test results are shown in Figure 2. The value on the graph shown is the average fitness value of 20 independent experiments.

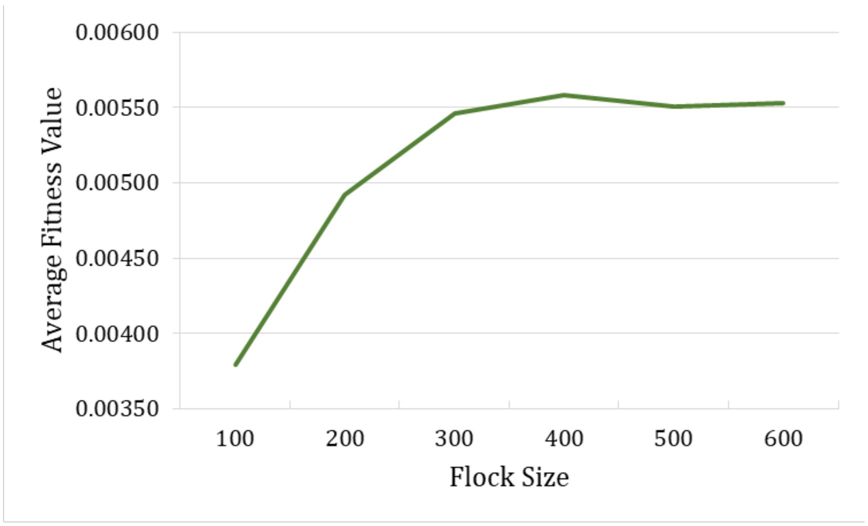

Figure 2. The testing result of flock size

In Figure 2, it is known that the larger flock size causes an increase in the average fitness value. A large flock size indicates that there are many agents who are looking for solutions scattered in the search space so that the algorithm can get the optimum global solution. However, the large flock size also causes more computational time needed. Figure 2 also shows that the average fitness value tends to be stable when the flock size is above 300. This means that no matter how many the flock size used if it is above 300, then it will produce fitness values that are almost the same as the flock size of 300. Thus, the optimal flock size is 300 because with this number the algorithm can provide a high fitness value and not much computational time is used.

The number of iterations is tested using several test scenarios with a range of values from 50 to 300. Each test scenario is run 20 times. The flock size used in this test is 300, which is the optimal value obtained from the previous flock size test. Other parameters that are set to a constant value are a flight length is 2 and the awareness probability is 0.1. The number of iterations test results are shown in Figure 3.

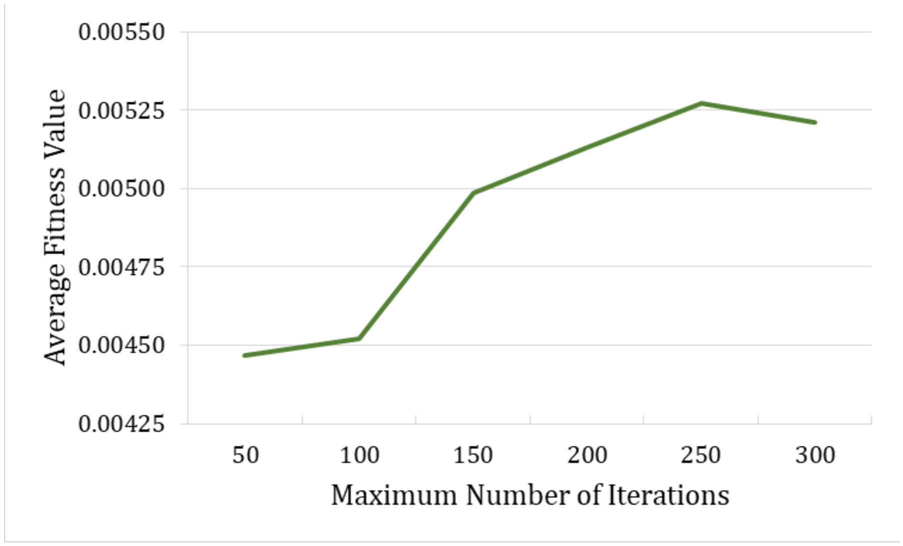

Figure 3. The testing result of maximum number of iterations

In Figure 3, it is known that the larger number of iterations causes an increase in the average fitness value. The large number of iterations indicates that search agents have a longer chance to improve the solution that has been obtained so that the solution becomes better than before. However, the large number of iterations also causes more computational time needed. Figure 3 also shows that the average fitness value tends to be stable when the number of iterations is above 250. This means that no matter how many the number of iterations used if it is above 250, then it will produce fitness values that are almost the same as the number of iterations of 250. Thus, the optimal number of iterations is 250.

The flight length is tested using several test scenarios with a range of values from 0.5 to 3. Each test scenario is run 20 times. This test is performed using the optimal parameters that have been obtained from the two previous tests, which consist of a flock size is 300 and the number of iterations is 250. Another parameter that is set to a constant value is the awareness probability is 0.1. The flight length test results are shown in Figure 4.

Flight length with a small value directs the algorithm to perform local searches, which means the search process is carried out around the current area of agent. Conversely, flight length with large value directs the algorithm to conduct global searches, which means the search process is carried out far from the current area of agent [17]. The value of flight length should not be too big or too small to balance the exploration and exploitation process. In Figure 4, it is known that a flight length of 2.5 produces the highest average fitness value, which means that the algorithm can produce a good solution using this value. Thus, the optimal flight length is 2.5.

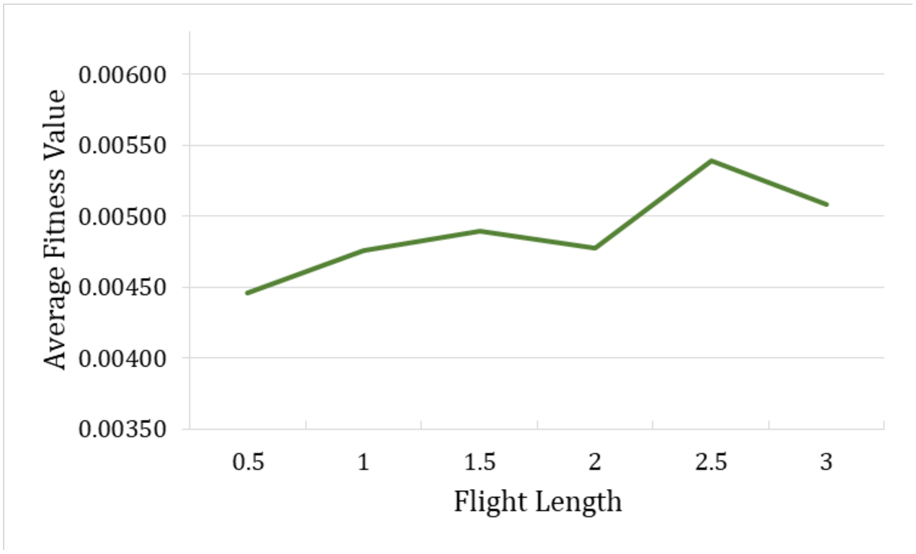

Figure 4. The testing result of flight length

The awareness probability is tested using several test scenarios with a range of values from 0.1 to 0.5. Each test scenario is run 20 times. This test is performed using the optimal parameters that have been obtained from the previous tests, which consist of a flock size is 300, the number of iterations is 250, and a flight length is 2.5. The awareness probability test results are shown in Figure 5.

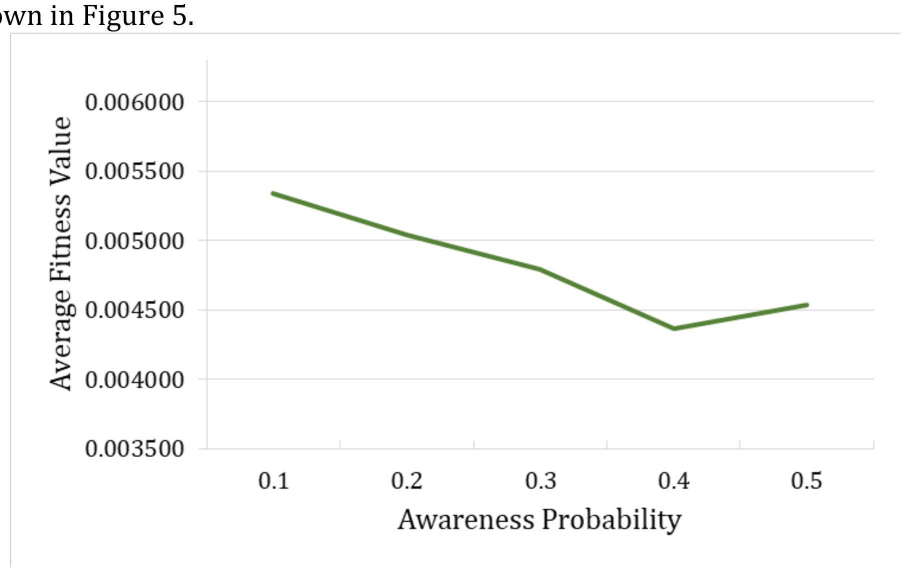

Figure 5. The testing result of awareness probability

The small awareness probability causes the algorithm to tend to search locally around the area of current good solution. Conversely, a large awareness probability causes the algorithm to explore the search space on a global scale [17]. The awareness probability should also not be too big or too small to balance the exploration and exploitation process. In Figure 5, it is known that the awareness probaility of 0.1 produces the highest average fitness value, which means that the algorithm can produce a good solution using this value. Thus, the optimal awareness probability is 0.1.

Based on the test results shown in Figure 2 to Figure 5, the optimal value for each parameter of ICSA is the flock size is 300, the maximum number of iterations is 250, the flight length is 2.5, and the probability of awareness is 0.1.

5.2 Comparison of ICSA with Other Algorithms

The performance of ICSA for data clustering is compared against the performance of K-Means, K-Medoids, genetic algorithm (GA), and Particle Swarm Optimization (PSO). K-Means and K-Medoids are used as comparison algorithms because they are popular non-metaheuristic algorithms which are quite simple and can work well for clustering problems [26]. While the metaheuristic algorithms used as comparison algorithms are GA and PSO because both of these algorithms can work well on high-dimensional data [27]. The experiment was carried out on four datasets. The dataset used is a classification dataset from the UCI dataset, which consists of Iris, Wine, Seeds, and Glass Identification datasets with details as shown in Table 1.

Table 1. Dataset details

Dataset	Number of Instances	Number of Attributes	Number of cluster
Iris	150	4	3
Wine	178	13	3
Seeds	210	7	3
Glass Identification	214	9	6

The performance of ICSA for data clustering is compared with other algorithms such as K-Means, K-Medoids, GA, and PSO. The tests are carried out using parameters with the same value to be fair. The maximum number of iterations used is 250 for all algorithms, while the number of search agents is 300 for the three metaheuristics algorithms (GA, PSO, and ICSA). Other parameters used for each metaheuristic algorithm are presented in Table 2.

These five algorithms are executed for 20 independent runs. Table 3 shows a comparison of the results obtained from the five clustering algorithms for the four datasets used. Values displayed are the average of RMSSTD, RS, and computational time obtained from 20 experiments.

Table 2. Parameter settings used for GA, PSO, and ICSA

Algorithm	Parameter Settings
GA	crossover probability = 0.6 mutation probability = 0.4 crossover method = heuristic crossover mutation method = random mutation selection method = elitism
PSO	inertia weight = 0.7 individual learning factor = 2 social learning factor = 2
ICSA	flight length = 2.5 awareness probability = 0.1

Table 3. Comparison of algorithm performance

Comparator	Algorithm	Dataset			
		Iris	Wine	Seeds	Glass
RMSSTD	K-Means	0.46451	37.10335	0.75597	0.65693
	K-Medoids	0.40773	34.96072	0.71357	0.57015
	GA	0.39059	32.58319	0.67680	0.54857
	PSO	*0.36418	*32.09808	*0.63804	0.50996
	ICSA	0.37488	32.11533	0.64187	*0.50792
RS	K-Means	0.80865	0.81924	0.68997	0.38387
	K-Medoids	0.85371	0.84003	0.72571	0.53442
	GA	0.86625	0.86110	0.75319	0.56937
	PSO	*0.88390	*0.86524	*0.78102	0.62868
	ICSA	0.87634	0.86510	0.77839	*0.63155
Time (ms)	K-Means	66	90	77	96
	K-Medoids	*62	*71	*58	*87
	GA	2605	3862	3939	6818
	PSO	1433	2152	2101	3653
	ICSA	1319	1951	1874	3479

* indicate the best value for each comparison

Based on the results of the comparison of the five algorithms with the four datasets shown in Table 3, it is known that the three metaheuristic algorithms provide a better solution compared to K-Means and K-Medoids on all datasets. This is due to the two algorithms working with a single solution on each iteration, while GA, PSO, and ICSA use multiple solutions. The selection of the initial centroids will greatly affect the final solution so failing to select the initial centroids will cause K-Means and K-Medoids to produce poor results because the solution provided is a local optimum. In the metaheuristic algorithm, the search process is carried out using multiple solutions and each solution will be updated continuously in each iteration to improve the quality of the solution so that it helps find the global optimum solution and increase the speed of convergence.

Among the three metaheuristic algorithms, PSO gives the best results on Iris, Wine, and Seeds dataset with the lowest RMSSTD value and the highest RS value among other algorithms. RMSSTD is the opposite of RS so that a low RMSSTD value causes the RS value to be high which means the solution is good, and vice versa. RMSSTD measures compactness while RS measures separation in each cluster. However, ICSA provides the best results compared to other algorithms in the Glass Identification dataset. Glass Identification dataset has quite a lot number of combinations of attribute and cluster compared to the other three datasets. This number determines the length of the dimensions of a search solution as shown in Figure 1. The longer the dimension of a solution, the more values are sought during the search process. This proves that ICSA can work better than the two other metaheuristic algorithms at high dimensions.

In computational time comparisons, K-Medoids has the smallest time compared to all algorithms, then followed by K-Means in second. Both of these algorithms require a fairly small computational time because both algorithms work with a single solution so the process of finding a solution is faster, but unfortunately the solution provided is not good enough compared to metaheuristic algorithms.

Metaheuristic algorithms require more computing time compared to K-Means and K-Medoids because it works with multiple solutions so that the solution obtained is a global optimum. Among the three metaheuristic algorithms, ICSA requires the smallest time to obtain an optimal solution compared to the other two algorithms on all datasets. The improve mechanism in ICSA is useful for balancing the process of exploration and exploitation when searching for solutions so that ICSA can get the optimal solution without spending a lot of computational time. With this mechanism, search agents can explore by visiting the search space that have not yet been visited, thereby increasing the chances of finding the best solution, as well as being able to exploit to find the best solution around the search space that have been previously found.

The use of ICSA to solve this clustering problem only needs to use four control parameters, namely flock size, maximum number of iterations, flight length, and awareness probability while the PSO requires five parameters and GA requires seven parameters to be determined. Surely this is an advantage of ICSA because there are not too many test parameters required. Testing parameters is one of the weaknesses of the metaheuristic algorithm because it is a time consuming task.

6. CONCLUSION

The test results show that GA, PSO, and ICSA require more computational time compared to K-Means and K-Medoids, but the three metaheuristic algorithms provide better solutions compared to K-Means and K-Medoids. By using four datasets, it is known that PSO provides the best solution among the other algorithms on the Iris, Wine, and Seeds dataset. However, for larger datasets such as Glass Identification, ICSA is able to provide the best solution. ICSA also requires the smallest computational time among the three metaheuristic algorithms in all datasets. It indicates that to get the optimal solution, ICSA does not require too much computational time like the other two metaheuristic algorithms. This is due to an improvement mechanism in updating the solution so that it can balance the exploration and exploitation processes when searching for solutions.

REFERENCES

[1] Renato Cordeiro de Amorim, Vladimir Makarenkov, **Applying Subclustering and Lp Distance in Weighted K-Means with Distributed Centroids**, *Neurocomputing*, Vol. 173, pp. 700–707, 2016.

[2] Kuo Ping Wu, Yung Piao Wu, Hahn Ming Lee, **Stock Trend Prediction by Using K-Means and AprioriAll Algorithm for Sequential Chart Pattern Mining**, *Journal of Information Science and Engineering*, Vol. 30, pp. 653–667, 2014.

[3] S. Vijayarani, S. Sudha, **An Efficient Clustering Algorithm for Predicting Diseases from Hemogram Blood Test Samples**, *Indian Journal of Science and Technology,* Vol. 8, No. 17, pp. 1–8, 2015.

[4] Pawan P. Warne, S. R. Ganorkar, **Detection of Diseases on Cotton Leaves Using K-Mean Clustering Method**, *International Research Journal of Engineering and Technology (IRJET)*, Vol. 2, No. 4, pp. 425–431, 2015.

[5] Fabiana Santos Lima, Daniel de Oliveira, Mirian Buss Gonçalves, Márcia Marcondes Altimari Samed, **Humanitarian Logistics: A Clustering Methodology for Assisting Humanitarian Operations**, *Journal of Technology Management and Innovation*, Vol. 9, No. 2, pp. 86–97, 2014.

[6] Tapas Kanungo, David M. Mount, Nathan S. Netanyahu, Christine D. Piatko, Ruth Silverman, Angela Y. Wu, **An Efficient K-Means Clustering Algorithm: Analysis and Implementation**, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 24, No. 7, pp. 881–892, 2002.

[7] Adriane B. S. Serapião, Guilherme S. Corrêa, Felipe B. Gonçalves, Veronica O. Carvalho, **Combining K-Means and K-Harmonic with Fish School Search Algorithm for Data Clustering Task on Graphics Processing Units**, *Applied Soft Computing*, Vol. 41, pp. 290–304, 2016.

[8] Naohiko Kinoshita, Yasunori Endo, Akira Sugawara, **On Hierarchical Linguistic-Based Clustering**, *Journal of Advanced Computational Intelligence and Intelligent Informatics*, Vol. 19, No. 6, pp. 900–906, 2015.

[9] Wayan Firdaus Mahmudy, Romeo M. Marian, Lee H. S. Luong, **Real Coded Genetic Algorithms for Solving Flexible Job-Shop Scheduling Problem - Part I: Modelling**, *Advanced Materials Research*, Vol. 701, pp. 359–363, 2013.

[10] G. Phanendra Babu, M. Narasimha Murty, **Clustering with Evolution Strategies**, *Pattern Recognition*, Vol. 27, No. 2, pp. 321–329, 1994.

[11] Ajith Abraham, Swagatam Das, Sandip Roy, **Swarm Intelligence Algorithms for Data Clustering**, Soft Computing for Knowledge Discovery and Data Mining, *Springer,* pp. 279–313 pp. 2008.
[12] Anna Maria Sri Asih, Bertha Maya Sopha, Gilang Kriptaniadewa, **Comparison Study of Metaheuristics: Empirical Application of Delivery Problems**, *International Journal of Engineering Business Management*, Vol. 9, pp. 1–12, 2017.
[13] Gamal Abd El-Nasser A. Said, Abeer M. Mahmoud, El Sayed M. El-Horbaty, **A Comparative Study of Meta-Heuristic Algorithms for Solving Quadratic Assignment Problem**, *International Journal of Advanced Computer Science and Applications*, Vol. 5, No. 1, pp. 1–6, 2014.
[14] Mohit Jain, Asha Rani, Vijander Singh, **An Improved Crow Search Algorithm for High-Dimensional Problems**, *Journal of Intelligent and Fuzzy Systems*, Vol. 33, No. 6, pp. 3597–3614, 2017.
[15] Mingru Zhao, Hengliang Tang, Jian Guo, Yuan Sun, **Data Clustering Using Particle Swarm Optimization**, In: Lecture Notes in Electrical Engineering, Future Information Technology, Springer, Berlin, Heidelberg, pp. 391–396, 2014.
[16] Jhila Nasiri, Farzin Modarres Khiyabani, **A Whale Optimization Algorithm (WOA) Approach for Clustering**, *Cogent Mathematics & Statistics*, Vol. 5, No. 1, pp. 1–13, 2018.
[17] Alireza Askarzadeh, **A Novel Metaheuristic Method for Solving Constrained Engineering Optimization Problems: Crow Search Algorithm**, *Computers and Structures*, Vol. 169, pp. 1–12, 2016.
[18] Alireza Askarzadeh, **Capacitor Placement in Distribution Systems for Power Loss Reduction and Voltage Improvement: A New Methodology**, *IET Generation, Transmission and Distribution*, Vol. 10, No. 14, pp. 3631–3638, 2016.
[19] Chiwen Qu, Yanming Fu, **Crow Search Algorithm Based on Neighborhood Search of Non-Inferior Solution Set**, *IEEE Access*, Vol. 7, pp. 52871–52895, 2019.
[20] Vivi Nur Wijayaningrum, Wayan Firdaus Mahmudy, Muhammad Halim Natsir, **Improved Simulated Annealing for Poultry Feed Formulation**, In: IEEE International Conference on Sustainable Information Engineering and Technology (SIET), IEEE, pp. 33–37, 2018.
[21] Stephen Merendino, M. Emre Celebi, **A Simulated Annealing Clustering Algorithm Based on Center Perturbation Using Gaussian Mutation**, In: FLAIRS 2013 - Proceedings of the 26th International Florida Artificial Intelligence Research Society Conference, pp. 456–461, 2013.
[22] Vivi Nur Wijayaningrum, Wayan Firdaus Mahmudy, Muhammad Halim Natsir, **Optimization of Poultry Feed Composition Using Hybrid Adaptive Genetic Algorithm and Simulated Annealing**, *Journal of*

Telecommunication, Electronic and Computer Engineering, Vol. 9, No. 2–8, pp. 183–187, 2017.

[23] Primitivo Díaz, Marco Pérez-Cisneros, Erik Cuevas, Omar Avalos, Jorge Gálvez, Salvador Hinojosa, Daniel Zaldivar, **An Improved Crow Search Algorithm Applied to Energy Problems**, *Energies*, Vol. 11, No. 3, pp. 571, 2018.

[24] Maria Halkidi, Yannis Batistakis, Michalis Vazirgiannis, **On Clustering Validation Techniques**, *Journal of Intelligent Information Systems*,. Vol. 17, No. 2, pp. 107–145, 2001.

[25] Marwan Hassani, Thomas Seidl, **Using Internal Evaluation Measures to Validate the Quality of Diverse Stream Clustering Algorithms**, *Vietnam Journal of Computer Science*, Vol. 4, No. 3, pp. 171–183, 2017.

[26] T. Velmurugan, **Efficiency of K-Means and K-Medoids Algorithms for Clustering Arbitrary Data Points**. *International Journal of Computer Technology & Applications*, Vol. 3, No. 5, pp. 1758–1764, 2012.

[27] Syarif Iwan, **Feature Selection of Network Intrusion Data Using Genetic Algorithm and Particle Swarm Optimization**, *EMITTER International Journal of Engineering Technology*, Vol. 4, No. 2, pp. 277–290, 2016.