

1-1-2017

Data clustering using seed disperser ant algorithm

WEN LIANG CHANG

JEEVAN KANESAN

ANAND JAYANT KULKARNI

HARIKRISHNAN RAMIAH

Follow this and additional works at: <https://journals.tubitak.gov.tr/elektrik>



Part of the [Computer Engineering Commons](#), [Computer Sciences Commons](#), and the [Electrical and Computer Engineering Commons](#)

Recommended Citation

CHANG, WEN LIANG; KANESAN, JEEVAN; KULKARNI, ANAND JAYANT; and RAMIAH, HARIKRISHNAN (2017) "Data clustering using seed disperser ant algorithm," *Turkish Journal of Electrical Engineering and Computer Sciences*: Vol. 25: No. 6, Article 7. <https://doi.org/10.3906/elk-1512-231>
Available at: <https://journals.tubitak.gov.tr/elektrik/vol25/iss6/7>

This Article is brought to you for free and open access by TÜBİTAK Academic Journals. It has been accepted for inclusion in Turkish Journal of Electrical Engineering and Computer Sciences by an authorized editor of TÜBİTAK Academic Journals. For more information, please contact academic.publications@tubitak.gov.tr.

Data clustering using seed disperser ant algorithm

Wen Liang CHANG^{1,*}, Jeevan KANESAN¹, Anand Jayant KULKARNI²,
Harikrishnan RAMIAH¹

¹Nature Inspired Meta-heuristic Group, Department of Electrical Engineering, Faculty of Engineering,
University of Malaya, Kuala Lumpur, Malaysia

²Department of Mechanical Engineering, Symbiosis Institute of Technology, Symbiosis International University,
Lavale, Pune, Maharashtra, India

Received: 23.12.2015

Accepted/Published Online: 29.06.2017

Final Version: 03.12.2017

Abstract: Nature-inspired optimization algorithms have become popular in the past decade. They have been applied to solve various kinds of problems. Among these would be data clustering, which has become popular in data mining in recent times due to the data explosion. In the last decade, many metaheuristic algorithms have been used to obtain improved data clustering optimization for solving data mining problems. In this paper, we applied the seed disperser ant algorithm (SDAA), which mimics the evolution of an *Aphaenogaster senilis* ant colony, and we introduced a modified SDAA that is a hybrid of K-means and SDAA for solving data clustering problems. The solutions obtained for the data clustering are very promising in terms of quality of solutions and convergence speed of the algorithm.

Key words: Optimization, data clustering, seed disperser ant algorithm

1. Introduction

Over the last decade, swarm intelligence has emerged as an efficient search and problem-solving tool based on behavior modeling of social insects such as ants and bees. Swarm inspired metaheuristics are applied to many kind of optimization problems, including data clustering, constrained engineering problems, vehicle routing problems, assignment problems, scheduling, global optimization, control engineering, traffic system design, and many others.

The seed dispenser ant algorithm (SDAA) [1] was inspired by the *Aphaenogaster senilis* [2] evolution process. The colony is populated by sterile female workers and queen ants. Female ants have diploid genes, whereas male ants are haploid. The males perform nuptial flights [3], where they disperse from their own colony to mate with a queen inhabiting a different colony. After mating, the queens generate offspring/eggs. These offspring are sterile female workers that have diploid genes. These diploids are produced when haploid male genes copy alleles from the queen's diploid. This allows the offspring to be highly related to each other, useful in localized approaches for the purpose of search exploitation. Subsequently, in global searches, the colony produces young queens that migrate to form new colonies. This enhances search exploration. The young queens establish new colonies and the nuptial flight process resumes again. This cycle of nuptial flight and young queen production is repeated until the optimization converges to optimal solutions.

The SDAA has an advantage in its searching technique whereby it searches by a binary bit changing process to generate new solutions. The offspring generation advocates search exploitation where the young

*Correspondence: wen.liang@siswa.um.edu.my

queen generation explores for a better optimum. Hence, the SDAA aggressively searches for optimum solutions within the search domain and this helps to escape local optima. The SDAA [1] was developed and validated using classical unconstrained optimization benchmarks [4] and difficult unconstrained composite optimization benchmarks problems [5].

In the age of big data, data processing has become a fundamental and critical challenge for data analysts. Data clustering is the most common technique in data processing, where it is used as an unsupervised classification technique. The clustering technique groups the data with the same or similar characteristics or patterns and divides the different groups into clusters.

Many algorithms have been developed to enable the solving of data clustering problems, such as several swarm intelligence (SI) [6] methods and evolutionary algorithms (EAs) [7]. Particle swarm optimization (PSO) [8–10] and the genetic algorithm (GA) [11,12] are the most familiar SI and EA methods respectively used for data clustering optimization. There are also other SI methods developed for solving data clustering problems, such as honeybee-mating optimization [13] and ant colony optimization (ACO) [14]. On the other hand, there are also learning-based algorithms developed for optimization in clustering such as cohort intelligence and modified K-means cohort intelligence (K-MCI) [15]. The motivation for improvement of the SDAA for data clustering problems focuses on reducing function evaluations by simplifying the SDAA and then combining it with K-means.

In this paper, we are proposing a modified SDAA (MSDAA) that hybridizes the advantages of both SDAA and K-means for solving data clustering problems. The purpose of developing the MSDAA is to enhance the original SDAA to produce improved accuracy and lower standard deviations with lower function evaluations with the aid of K-means in the algorithm. This paper is organized as follows. Section 1 gives an overview of the research and acts as an introduction to the study. Section 2 describes the original SDAA and the MSDAA for clustering. Section 3 presents the results of the experiment followed by an in-depth discussion. Finally, Section 4 presents our conclusions and recommendations for future study.

2. Seed disperser ant algorithm (SDAA) and modified seed disperser ant algorithm (MSDAA)

2.1. Seed disperser ant algorithm (SDAA)

Consider a general minimization problem as follows:

$$\text{Minimize } f(L_h) = f(l_1, \dots, l_i, \dots, l_N), \tag{1}$$

where

$$l_i = \text{ith variable or dimension} \tag{2}$$

$$\Psi_i^{lower} \leq l_i \leq \Psi_i^{upper}, \quad i = 1, \dots, N. \tag{3}$$

For the SDAA, the objective function $f(L_h)$ is treated as the fitness of the male ant in the colony. The male ants' gene code is represented by $L_h = (l_1, \dots, l_i, \dots, l_N)$. The queen of each colony produces many virgin queens with the reference gene $[R_h]_{C_i}$ as the identity gene of the colony. The mated queen $[Q_d]_{C_i}$ is represented by pairing the binary gene code of the male ant and its complement, reference gene $[R_h]_{C_i} = [\bar{L}_h]_{C_i}$, as shown in Eq. (4):

$$[Q_d]_{C_i} = [L_h]_{C_i} [R_h]_{C_i}, \tag{4}$$

where

$$C_i = \text{ith colony.}$$

The SDAA procedure is initialized with parameters such as decimal points of precisions for input variables, maximum number of iterations n_{\max} , number of colonies C , shrinking factor r , convergence parameter ε , and saturation number S . Initially, upper boundary UB_i and lower boundary LB_i are fixed as given in the equations below. Decimal points of precisions for input variables are used to determine the number of bits The researchers used 5 decimal points of precisions for input variables (L_h) in this research.

$$UB_i = \Psi_i^{upper} \tag{5}$$

$$LB_i = \Psi_i^{lower}, \quad i = 1, \dots, N \tag{6}$$

Step 1: Male ant L_h is generated randomly as shown in Eq. (7) below:

$$[L_h]_{C_i} = [l_1, \dots, l_i, \dots, l_N]_{C_i}, \tag{7}$$

where

$$l_i = LB_i + rand \times (UB_i - LB_i), \quad rand \sim \cup([0, 1]).$$

Step 2: Every l_i is converted to binary form for the male gene $[L_h]_{C_i}$ in every colony. The binary coded gene is complemented to form colonies' identity reference gene $[R_h]_{C_i}$ as shown in Eq. (8) below:

$$[R_h]_{C_i} = [\bar{L}_h]_{C_i} = [l'_1, \dots, l'_i, \dots, l'_N]_{C_i}, \tag{8}$$

and every mated queen $[Q_d]_{C_i}$ is formed as given in Eq. (4).

Step 3: Offspring generated. The mated queens then generate offspring. This is carried out when the gene (L_h) copies binary bits from the reference gene (R_h), commencing from the least significant bit (LSB) to the most significant bit (MSB) and vice versa. For example, given a three-dimensional gene L_h of colony 1, $[L_h]_{C_1} = [l_1, l_2, l_3]_{C_1}$ with the reference gene $[R_h]_{C_1} = [l'_1, l'_2, l'_3]_{C_1}$, dimension $l_1 = 1110010$ and the complement of it in the reference gene is $l'_1 = 0001101$. The dimensions l_1 of the offspring are generated as shown below:

- Generated by LSB to MSB and MSB to LSB example copying process:

$l_1 = 1110010,$	$l'_1 = 0001101$
LSB to MSB:	MSB to LSB:
11100101110010	
111001 <u>1</u>	<u>0</u> 110010
111000 <u>1</u>	<u>00</u> 10010
\vdots	\vdots
<u>0001101</u>	<u>0001101</u>

Step 4: By evaluating the fitness based on the objective function, the best offspring of each colony was selected $[L_{h(fit)}]_{C_i}$ and the associated best gene denoted as $L_{h(best)}$ was stored. It is accepted as the current best solution if it does not improve for a considerable number of iterations. In such a situation, progress to Step 6; otherwise, continue to Step 5.

Step 5: Nuptial flight. In nuptial flights, every male flies out from its colony to mate with a queen inhabiting its colony. This process is mimicked by pairing the best offspring $[L_{h(fit)}]_{C_i}$ of each colony with the next colony reference gene $[R_h]_{C_{i+1}}$. The last colony's best offspring $[L_{h(fit)}]_{C_N}$ will be paired with the first colony's reference gene $[R_h]_{C_1}$. This process is represented in Eq. (9) as shown below and continues to Step 3.

In colony C_{i+1} ,

$$[L_h]_{C_{i+1}} [R_h]_{C_{i+1}} = [L_{h(fit)}]_{C_i} [R_h]_{C_{i+1}} \quad (9)$$

where $i = 1, 2, \dots, N$

and for the 1st colony C_1 :

$$[L_h]_{C_1} [R_h]_{C_1} = [L_{h(fit)}]_{C_N} [R_h]_{C_1}.$$

Step 6: The search boundary is reduced by shrinking factor r based on $L_{h(best)}$ as the center of the boundary. The shrinking process is given in Eq. (10) as below:

$$\text{Boundary size, } B_i = r \times (UB_i - LB_i) \quad (10)$$

where

$$i = 1, \dots, N.$$

Then UB_i and LB_i are generated as shown below:

$$UB_i = [l_i]_{L_{h(best)}} + \frac{1}{2} B_i, \quad (11)$$

$$LB_i = [l_i]_{L_{h(best)}} - \frac{1}{2} B_i. \quad (12)$$

Step 7: After several nuptial flights in a colony, new queens are produced that will lead new colonies. This is carried out by regenerating all the $[L_h]_{C_i}$ and $[R_h]_{C_i}$, similar to Step 1 and Step 2. However, the current generation's best gene $L_{h(best)}$ will replace one of the randomly chosen queen's reference gene $[R_h]_{C_i}$. This ensures that the best gene found in the current generation is retained in the next generation, i.e. elitism. This is represented as follows:

$$[R_h]_{C_{random}} = L_{h(best)}. \quad (13)$$

If there is no significant improvement in $f(L_{h(best)})$ for a significant number of iterations, the evolution process is considered converged. The SDAA flowchart is shown in Figure 1. The best gene $L_{h(best)}$ is accepted as the final solution when either of the criteria listed below is valid or else continue to Step 2.

- If the SDAA converged by satisfying the conditions shown below:

$$\left\| \max (f(L_{h(best)}))^n - \max (f(L_{h(best)}))^{n-1} \right\| \leq \varepsilon \text{ and}$$

$$\left\| \min (f(L_{h(best)}))^n - \min (f(L_{h(best)}))^{n-1} \right\| \leq \varepsilon \text{ and}$$

$$\left\| \max (f(L_{h(best)}))^n - \min (f(L_{h(best)}))^n \right\| \leq \varepsilon.$$

- If maximum number of iterations n_{max} is exceeded.

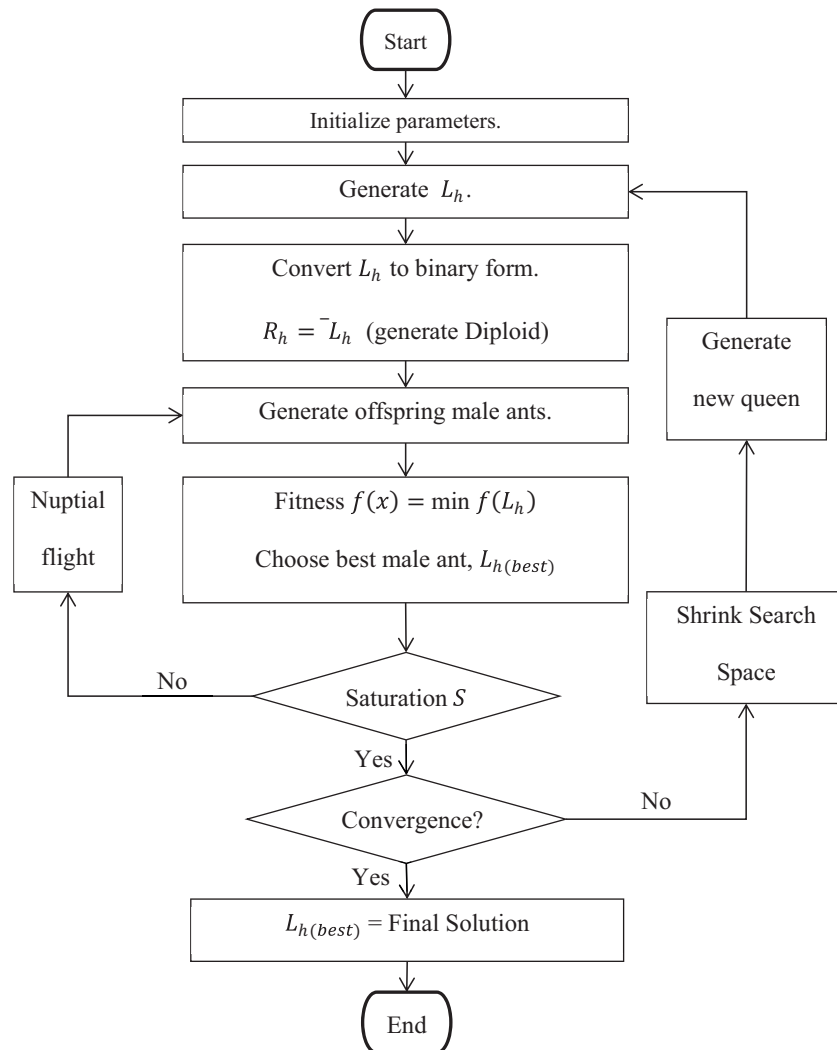


Figure 1. Flowchart of SDAA.

2.2. Modified seed disperser ant algorithm (MSDAA)

The SDAA was modified to perform better in terms of accuracy as well as to help function evaluations reduction. This was carried out by making changes in Step 1 and removing Step 6 and Step 7 of the SDAA. In the initialization, the control parameters such as decimal points of precisions for input variables, number of colony C , and saturation number S were set. The upper boundary UB_i and lower boundary LB_i were generated in the same way as in the original SDAA.

In Step 1, K-means search was used instead of random generating solutions. The solution or centroid was generated using K-means optimization and became the reference point to generate male ant gene L_h . By using the same concept of Step 6, the boundary size, B_i was fixed to be $\frac{2}{3}$ of the original boundary size. Male ant gene L_h was generated randomly around the reference point. Step 2 to Step 5 was the same as in the original SDAA. Once the saturation was reached in Step 5, $L_{h(best)}$ was accepted as the final solution. Step 6 and Step 7 of the original SDAA were removed completely as they consume large numbers of function evaluations. The modification implemented on SDAA resulted in significant reductions of function evaluations for data clustering

in comparison with the original SDAA as well as other optimization algorithms. The MSDAA was developed by adopting the K-means search potential solution that speeds up the overall search process and removing the excess steps of new queen spawn iteration from the original SDAA. This enabled the MSDAA to obtain better results in a shorter period of time in conjunction to the SDAA as well as other algorithms. The main difference between the MSDAA and SDAA is that the MSDAA has only 1 randomization compared to the SDAA's 2. In the MSDAA, the initial randomization (Step 1) of the SDAA is replaced with K-means. This means that the new queen's colonies establishment is replaced by K-means, and this simplifies the evolution process by K-means, which directly produces a fit generation and consequently the nuptial flight search ensures the best solution. This nuptial flight convergence stops when the solution saturates after several iterations. Existing offspring generation (Step 3) and nuptial flight (Step 5) combined with initial K-means produces sufficient randomness to lead the solution to convergence; hence, removing Steps 6 and 7 helps reduce the converging speed.

In clustering, for a set of data, D with N data objects is clustered to K sets of clusters:

$$D = [Y_1, Y_2, \dots, Y_N], \text{ where } Y_i \in R^D,$$

$$\text{Clusters, } S = [X_1, X_2, \dots, X_K].$$

Each data point in set D was allocated to one of the K clusters in such way that it minimized the objective function. The objective function is the sum of squared Euclidean distance between each object Y_i and the center of the cluster X_j to which it belongs [15]. This objective function is given by Eq. (14):

$$F(X, Y) = \sum_{i=1}^N \text{Min} \|Y_i - X_j\|^2, \tag{14}$$

$$j = 1, 2, \dots, K.$$

Also,

$$X_j \neq \emptyset, \forall j \{1, 2, \dots, K\}, \quad X_i \cap X_j \neq \emptyset, \forall i \neq j \text{ and } \forall ij \{1, 2, \dots, K\},$$

$$\bigcup_{j=1}^K X_j = D.$$

For data clustering problems, the objective function $f(x) = F(XY)$ is represented by the sum of squared Euclidean distance between each object Y_i and the center of the cluster X_j as shown in Eq. (15):

$$\text{Minimize } f(L_h) = f(l_1, \dots, l_i, \dots, l_N), \tag{15}$$

Subject to $\Psi_i^{lower} \leq l_i \leq \Psi_i^{upper}$, $i = 1, \dots, N$.

In the MSDAA, the male ant gene $L_h = [l_1, \dots, l_j, \dots, l_K]$ represents the center of the cluster, where dimension X_j is represented by l_j ,

where

$$L_h = [X_1, \dots, X_j, \dots, X_K] = [l_1, \dots, l_j, \dots, l_K]. \tag{16}$$

Thus, the objective is shown as:

$$f(x) = F(L_h Y). \tag{17}$$

The MSDAA flowchart is shown in Figure 2.

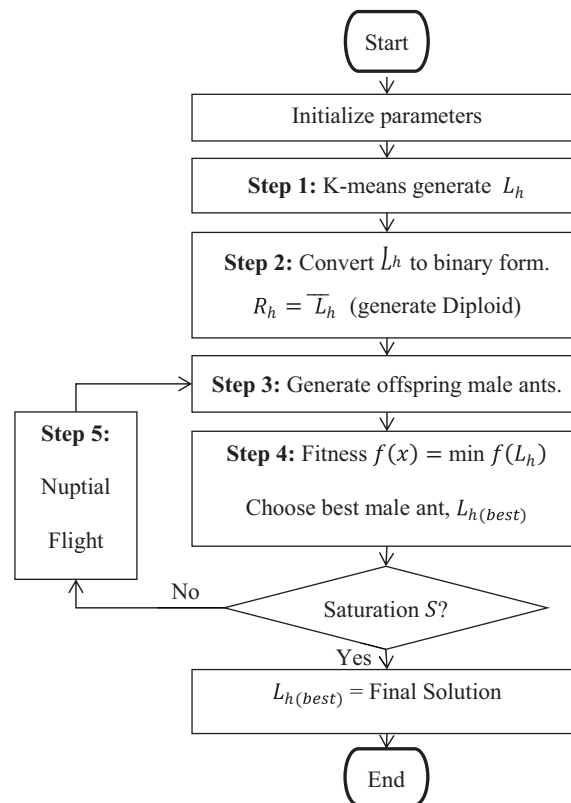


Figure 2. Flowchart of MSDAA.

3. Numerical experiments and results

The SDAA was coded in MATLAB (R2012a) using a computer with the Windows 7 platform, Intel i7-4770, 3.40 GHz processor, and 16 GB RAM. The simulations were carried out 30 times for every data clustering problem. To validate the SDAA, six real data sets from the UCI Machine Learning Repository were used. Each data set had different numbers of clusters, data objects, and features, as described in Table 1. *Iris setosa*, *Iris versicolour*, and *Iris virginica* are the *Iris* flowers species that constitute the Iris data set. Fifty samples of each species with four features, which are sepal length and width and petal length and width, were collected. The Wine data set is the result of a chemical analysis of wines from three different cultivators in the same region in Italy. The three different types of cultivators are derived into class 1 (59 instances), class 2 (71 instances), and class 3 (48 instances). Thirteen features including alcohol, malic acid, ash, alkalinity of ash, magnesium, total phenols, flavanoids, nonflavanoid phenols, proanthocyanins, color intensity, hue, OD280/OD315 of diluted wines, and proline are analyzed to determine the quantities found in each type of wine. Six types of glass in the Glass data set consist of building windows float processed (70 objects), building windows not float processed (76 objects), vehicle windows float processed (17 objects), containers (13 objects), tableware (9 objects), and headlamps (29 objects). The nine features for each type of glass are refractive index, sodium, magnesium, aluminum, silicon, potassium, calcium, barium, and iron. The Contraceptive Method Choice (CMC) data set is a subset of the 1987 National Indonesia Contraceptive Prevalence Survey. The samples are married women who were either not pregnant or did not know if they were at the time of interview. The problem is to predict the current contraceptive method choice (no use, long-term methods, or short-term methods) of a woman based on

her demographic and socioeconomic characteristics. The Breast Cancer Wisconsin data set contains 2 categories and 683 objects, which are malignant (444 objects) and benign (239 objects). The nine features in each category are clump thickness, cell size uniformity, cell shape uniformity, marginal adhesion, single epithelial cell size, bare nuclei, bland chromatin, normal nucleoli, and mitoses. The Vowel data set consists of 871 Indian Telugu vowels sounds with six overlapping vowel classes. The vowel classes included d (72 instances), a (89 instances), i (172 instances), u (151 instances), e (207 instances), and o (180 instances). There are three input features corresponding to the first, second, and third vowel frequencies for each class.

Table 1. UCI machine learning repository data set information.

Data set	Number of data sets, N	Dimensions, D	Number of clusters, K
Iris	150	4	3
Wine	178	13	3
Breast Cancer Wisconsin	683	9	2
Contraceptive Method Choice (CMC)	1473	10	3
Glass	214	9	6
Vowel	871	3	6

Table 2a shows the clustering results of 6 different data sets solved by 6 different algorithms. The results of the 6 different data sets solved included the Iris data set, Wine data set, Cancer data set, CMC data set, Glass data set, and Vowel data set. Comparing the mean results, the MSDAA performed very well whereby it enabled minimum results for all the cases to be achieved. The MSDAA also shows a low standard deviation (SD), which means it is able to obtain minimum results frequently. On the other hand, the MSDAA has the lowest function evaluation (FEs) and computation time compared to all other algorithms in Table 2b. It showed significant improvement in comparison to the FEs of the MSDAA and SDAA. Based on the overall average FEs in Table 2b, the MSDAA can be seen to perform well with 2900 FEs. This shows that the MSDAA is not affected much by cluster size and dimension. The enhancement from the SDAA to the MSDAA resulted in improvement to the overall average FEs from 9928 to 2900, which is about 70% reduction of FEs. Besides, the MSDAA was able to solve these problems in the shortest time compared to the other algorithms based on the computation times of Table 2b. The higher number of dimensions increases the complexities that require more computation time. The MSDAA was able to perform well on the Cancer data set and Glass data set problems. Table 3 shows the parameters used by the MSDAA compared to the SDAA. The parameters used for the MSDAA were reduced compared to the SDAA. The overall average FEs of all data clustering algorithms were compared and the MSDAA used the least FEs for solving all these data clustering problems.

The objective function of intercluster variance is defined as the sum of squared Euclidean distance, which means that a different objective function was produced by a different found centroid. The different locations of the centroid create chances for different data to be grouped into the cluster. Thus, a minor improvement of the results should take into consideration the boundaries of the data set given. For example, the Iris data set has smaller values compared to other data sets. Thus, more significant figures were taken into consideration.

Figures 3 and 4 show convergence plots of the Wine and Cancer clustering problems. The convergence plots show the fitness/objective function $f(x)$ of all 10 L_h mated with the queen in every colony in every nuptial flight. The fitness/solution of male ants in every nuptial flight was improved compared to previous nuptial flights, clearly showing that the nuptial flight helps to improve the fitness of male ants. This necessarily demonstrated the ability of the K-SSDAA to jump out of local minima and further reach the global minimum.

Table 2a. Simulation results of data sets for different algorithms.

Data set		GA	ACO	PSO	K-MCI	SDAA	MSDAA
Iris	Mean	125.19	97.17	97.23	96.65	96.65	96.65
	SD	14.563	0.367	0.347	0	0	0
	FEs	38128	10998	4953	3500	7080	2800
Wine	Mean	16,530.53	16,530.53	16,417.47	16,292.70	16,292.24	16,292.24
	SD	0	0	85.497	0.130	8.24E-02	0.1782
	FEs	33,551	15,473	16,532	6250	10,150	2900
Cancer	Mean	3249.46	3046.06	3050.04	2964.38	2964.38	2964.38
	SD	229.734	90.500	110.801	0	0	0
	FEs	20,221	15,983	16290	5000	9860	2900
CMC	Mean	5756.59	5819.13	5820.96	5693.75	5694.01	5693.75
	SD	50.369	45.634	46.959	0.014	2.47E-01	0.01366
	FEs	29483	20,436	21,456	15,000	11,020	2900
Glass	Mean	282.32	273.46	275.71	212.57	220.05	210.50
	SD	4.138	3.584	4.550	0.135	9.453	0.0142
	FEs	199,892	196,581	198,765	25,000	11310	3000
Vowel	Mean	159,153.49	159,458.14	151,999.82	148,967.55	148,969.63	148,967.45
	SD	3105.54	3485.38	2881.34	36.08	1.79	1.16
	FEs	10,548	8046	9635	7500	10,150	2900

Table 2b. Function evaluation and computation time for different algorithms.

Data set		GA	ACO	PSO	K-MCI	SDAA	MSDAA
Iris	FEs	38,128	10,998	4953	3500	7080	2800
	Time(s)	83.56	87.04	17.34	15.89	22.85	12.37
Wine	FEs	33,551	15,473	16,532	6250	10,150	2900
	Time(s)	257.73	378.06	151.95	81.38	103.89	36.51
Cancer	FEs	20,221	15,983	16,290	5000	9860	2900
	Time(s)	953.58	1269.59	754.37	251.57	446.84	139.64
CMC	FEs	29,483	20,436	21,456	15,000	11,020	2900
	Time(s)	926.78	1523.82	731.84	532.46	348.74	105.57
Glass	FEs	199,892	196,581	198,765	25,000	11310	3000
	Time(s)	3385.57	4575.76	2917.70	693.85	297.86	123.72
Vowel	FEs	10,548	table 8046	9635	7500	10150	2900
	Time(s)	314.74	492.48	328.96	247.61	312.11	107.57
Avg. FEs		55,303	44,586	43,634	10,375	9928	2900

Table 3. Parameters used for the SDAA and MSDAA for data clustering.

SDAA					MSDAA	
<i>C</i>	<i>S</i>	<i>G</i>	<i>r</i>	ϵ	<i>C</i>	<i>S</i>
10	10	200	0.75	1E-05	10	10

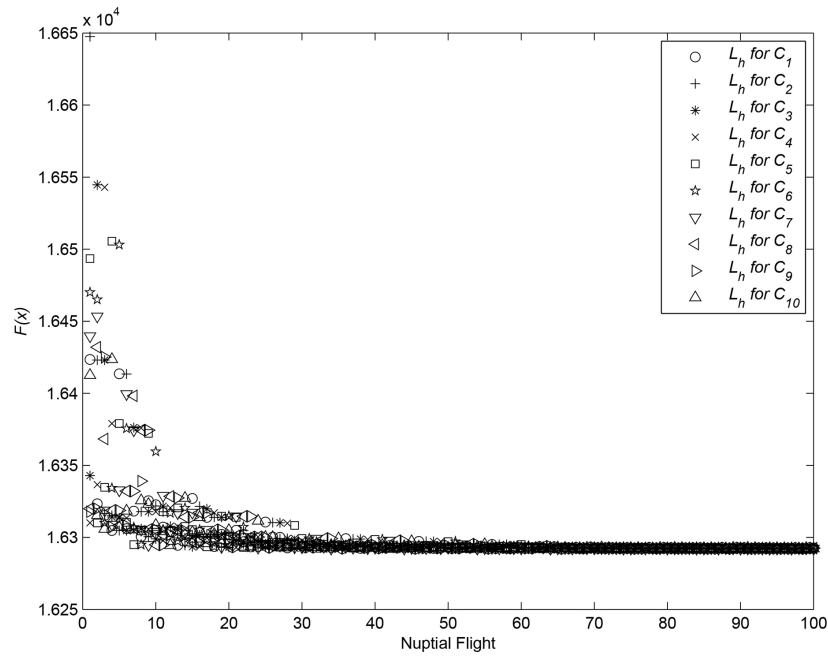


Figure 3. Convergence of Wine clustering problem.

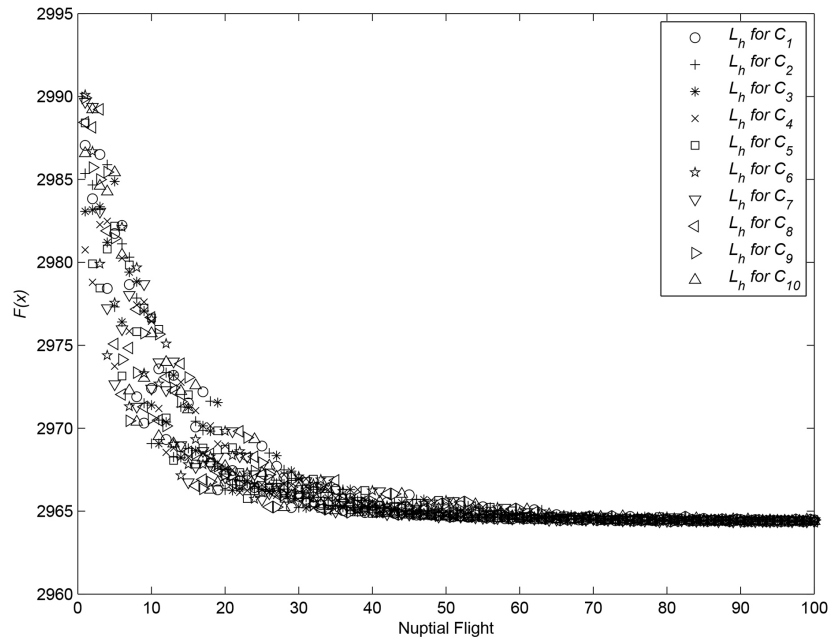


Figure 4. Convergence of Cancer clustering problem.

4. Conclusions and future directions

The SDAA has the potential to solve optimization problems such as data clustering. This technique is successfully developed using the concept of male ants performing nuptial flights to generate new superior colonies. The nuptial flights and production of young queens ensures that the optimization continuously improves to find the global minimum. The production of young queens helps the optimization to escape local optima. The

SDAA shows comparable results for solving data clustering. However, the SDAA has potential for improvement by reducing FEs. With this purpose, the researchers proposed a hybrid algorithm of K-means and the SDAA, namely the MSDAA. K-means solves data clustering problems with low FEs. K-means in the MSDAA is used to expedite the process of searching to provide a near optimum followed by the SDAA search. The results show that the MSDAA is able to solve a variety of clustering problems with low FEs. Moreover, no significant variations in FEs were observed despite differences in dimensions and cluster size among the problems. Taken as a whole, the MSDAA performs better than all other metaheuristics shown in this work in terms of best result, mean result, and FEs. The researchers would like to improve the MSDAA for solving real-life clustering problems in the future.

Acknowledgment

The researchers are thankful for the financial support from the Universiti Malaya Research Grant (Grant No.: UMRG333-15AFR).

References

- [1] Chang WL, Kanesan J, Kulkarni AJ. Seed disperser ant algorithm: an evolutionary approach for optimization. In: *Evostar 2015: Applications of Evolutionary Computation*; 8–10 April 2015; Copenhagen, Denmark. Berlin, Germany: Springer. pp. 643-654.
- [2] Chéron B, Doums C, Fédérici P, Monnin T. Queen replacement in the monogynous ant *Aphaenogaster senilis*: supernumerary queens as life insurance. *Anim Behav* 2009; 78: 1317-1325.
- [3] Kenne M, Dejean A. Nuptial flight of *Myrmicaria opaciventris* (Hymenoptera: Formicidae, Myrmicinae). *Sociobiology* 1998; 31: 41-50.
- [4] Shang YW, Qiu YH. A note on the extended Rosenbrock function. *Evol Comput* 2006; 14: 119-126.
- [5] Liang J, Suganthan P, Deb K. Novel composition test functions for numerical global optimization. In: *Proceedings of the 2005 Swarm Intelligence Symposium*; 8–10 June 2005; Pasadena, California. New York, NY, USA: IEEE. pp. 68-75.
- [6] Blum C, Li X. Swarm intelligence in optimization. In: Rozenberg G, Bäck T, Eiben AE, Kok JN, Spaink HP, editors. *Natural Computing Series*. Berlin, Germany: Springer, 2008. pp. 43-85.
- [7] Back T. *Evolutionary Algorithms in Theory and Practice*. New York, NY, USA: Oxford University Press, 1996.
- [8] Kennedy J, Eberhart R. Particle swarm optimization. In: *Proceedings of IEEE International Conference on Neural Networks 1995*; 27 November–1 December 1995; Perth, Australia. New York, NY, USA: IEEE. pp. 1942-1948.
- [9] Tsai CY, Kao IW. Particle swarm optimization with selective particle regeneration for data clustering. *Expert Syst Appl* 2011; 38: 6565-6576.
- [10] Kennedy J. Particle swarm optimization. In: Sammut C, Webb GI, editors. *Encyclopedia of Machine Learning*. Berlin, Germany: Springer, 2010. pp. 760-766.
- [11] Mitchell M. *An Introduction to Genetic Algorithms*. Cambridge, MA, USA: MIT Press, 1998.
- [12] Maulik U, Bandyopadhyay S. Genetic algorithm-based clustering technique. *Pattern Recogn* 2000; 33: 1455-1465.
- [13] Fathian M, Amiri B. A honeybee-mating approach for cluster analysis. *Int J Adv Manuf Tech* 2008; 38: 809-821.
- [14] Dorigo M, Birattari M. Ant colony optimization. In: Sammut C, Webb GI, editors. *Encyclopedia of Machine Learning*. Berlin, Germany: Springer, 2010. pp. 36-39.
- [15] Krishnasamy G, Kulkarni AJ, Paramesran R. A hybrid approach for data clustering based on modified cohort intelligence and K-means. *Expert Syst Appl* 2014; 41: 6009-6016.