Fast Artificial Bee Colony for Clustering

Abba Suganda Girsang
Computer Science Department, BINUS Graduate Program - Master of Computer Science
Bina Nusantara University, Jakarta, Indonesia, 11480
E-mail: agirsang@binus.edu

Yohan Muliono and Fanny Fanny
Computer Science Department, School of Computer Science, Bina Nusantara University
Jakarta, Indonesia, 11480
E-mail: ymuliono@binus.edu, fanny.sa@binus.edu

Keywords: artificial bee colony, clustering, fast ABC, redundant process

Received: February 14, 2017

Artificial Bee Colony (ABC) is one of good heuristic intelligent algorithm to solve optimization problem including clustering. Generally, the heuristic algorithm will take the high computation time to solve optimization problem. Likewise, ABC also consumes too much time to solve clustering problem. This paper intends solving clustering problem using ABC with focusing reduction computation time called FABCC. This idea proposes detecting the pattern of redundant process then compacting it to effective process to diminish the computation process. There are five data sets to be used to prove the performance of FABCC. The results shows that FABCC is effective to prune the duration process up to 46.58 %.

Povzetek: Predstavljena je izboljšava v algoritmu Artificial Bee Colony za gručenje, ki dosega na merjenih domenah skoraj 50% pohitritev.

1 Introduction

Clustering is the unsupervised classification of patterns (observations, data items, or feature vectors) into groups (clusters). The clustering problem has been addressed in many contexts in many disciplines [1]. Yang and Kamel state as a machine learning perspective that clustering is unsupervised learning because no category labels denoting appropriate partition of the objects are used [2]. Generally, there are two types of data to be clustered; metric as a numerical data and non-metric as not a numerical data [3].

Clustering problem has a broad appeal as one of the steps in exploratory data analysis. Jain et al describe some important applications of clustering algorithms such as image segmentation, object recognition, and information retrieval [1].

Recently, population-based optimization based on behaviour of animal swarm often used as a solution to find an optimization problem such as travelling salesman problem [4]–[6] and clustering [7]–[11]. Artificial bee colony (ABC) algorithm is one of the most recently introduced swarm-based algorithms. ABC simulates the intelligent foraging behaviour of a honey bee swarm [8] to get the optimal solution. ABC has been proved as a good algorithm for solving clustering [8]. Their proposed method can cluster perfectly accurate for Cancer-Int, Iris and wine dataset. This algorithm also shows the good performance comparing with ten algorithms (PSO, BayesNet, MlpANN, RBF, KStar, Bagging, MultiBoost, NBTree, Ridor and VFI). However, compared to other evolutionary algorithms, ABC has a challenging problem. For example, the convergence speed of ABC is slower than the other representative of population-based algorithm [12]. Also, like the general evolutionary algorithm, ABC has many repetition computations before converging solution. However, some of researchers commonly depend on their algorithm to find a best solution only, nevertheless they forget about time needed to fully operate their algorithm. In that case, this research is conducted to focus on the time and try not to stray excess average best solution. Some researchers have attempt solving this problem. Girsang et al proposed BCOPR that is inspired bee colony optimization to gain the fast process to solve travelling salesman problem [13]. Lu et al also used bee swarm for fast clustering [14].This research conducts a fast algorithm using ABC algorithm to solve the cluster problem. The reason why this research focuses on artificial bee colony besides of recent success of clustering data in ABC as stated in [4] [7] [8], because ABC also have many slots to be modified as a fast algorithm. Karaboga used greedy algorithm to be applied in ABC [8] and Zhang [7] used Deb's Algorithm [15] instead of greedy because Zhang believe that deb's algorithm is much more simple.

The remainder of this paper is organized as follows: Section 2 gives a brief introduction to the clustering problem and artificial bee colony. Section 3 provides a detailed description of the FABCC algorithm, while the performance evaluation of the proposed algorithm is presented in Section 4. Finally, conclusions are offered in Section 5.

2 Related work

2.1 *K*-means Clustering

Clustering algorithm generally is classified into two big parts, first one known as hierarchical clustering and second one is partition clustering [16]. Hierarchical clustering group data objects with a sequence of partitions. Hierarchical procedures divided into two segments which are agglomerative and divisive. Where agglomerative approach begins with each pattern in distinct cluster (single cluster) and then will be merged later. Divisive pattern is vice versa, begin with a single cluster and will be divided later [1]. Partitional clustering algorithm obtains a single partition of data instead of a clustering structure. This technique usually produce clusters by optimizing a criterion function defined either locally or globally. This research will be using partition, because we already know the total of the clusters. And we start with the number of clusters without reducing or adding it.

Among such a varies clustering formulations that are based on minimizing a formal objective function, the most widely used and studied is k-means clustering [17]. Clustering based on k-means is closely related to a number of other clustering and location problems. These include the Euclidean k-medians in which the objective is to minimize the distance to the nearest centre of the centroid. The aim of the K-means algorithm is to divide M points in N dimensions into K clusters so that the within-cluster sum of squares is minimized. It is not practical to require that the solution has minimal sum of squares against all partitions, except when M, N are small and K = 2. We seek instead "local" optima, solutions such that no movement of a point from one cluster to another will reduce the within-cluster sum of squares[18]. This research will do a similar algorithm to k-means, modified the algorithm a little by adding greedy algorithm to swap the route of the clusters.

2.2 Artificial Bee Colony

Artificial Bee Colony (ABC) Algorithm is one of the swarm intelligence, whereas its copy the mechanism of honey bee swarm's intelligence to find the food source [19]. Originally, ABC optimization was proposed for solving numerical problems [20]. Therefore, the first studies aimed to evaluate the performance of ABC on the widely used set of numerical benchmark test functions and to compare it with that of well-known evolutionary algorithms such as Genetic Algorithm, Particle Swarm Optimization and Ant Colony Optimization. [19] There are 3 types of bee. The first one is employed bees which search for a food source. The food source value depends on many factors, such as its proximity to the nest, richness or concentration of energy, and the ease of extracting this energy [21]. Employed Bee will do a waggle dance later. The more the food source, the longer the waggle dance will last. The waggle dance represent the fitness value. Second is onlooker bees which wait employed bee to do waggle dance and choose randomly according to how much fitness value of the employed bees. The last one is scout bees which looking for the food source without pattern. The position of the food source represents a solution that can be made by the bees. And the amount of the nectar represents a better solution that can be found by the bees. Whereas in [19] Karaboga has proved that ABC can be used to optimize multivariable functions and ABC outperforms the other swarm intelligence algorithm such as Genetic Algorithm, Particle Swarm Algorithm and Particle Swarm Inspired Evolutionary Algorithm (PS-EA) [19]. The main steps of Artificial Bee Colony algorithm are:

The main steps of artificial bee colony algorithm are:

Step 1: Initialize the population of solutions randomly and evaluate them.

Step 2: Produce new solutions for each employed bees, evaluate them, and apply the greedy solutions for them and the greedy selection process.

Step 3: Calculate the probabilities of current sources (employed bees) with which they are preferred by the onlookers.

Step 4: Assign onlooker bees to employed bees according to probabilities.

Step 5: Produce new solutions for each onlooker bees, evaluate them, and apply the greedy selection process.

Step 6: Stop the exploitation process of the sources abandoned by bees and send the scouts in the search area for discovering new food sources, randomly.

Step 7: Memorize the best food source found so far.

Step 8: If the termination condition is still not met, repeat the algorithm process from **Step 2**, otherwise stop the algorithm.

As the development of Karaboga's artificial bee algorithm, Zhang contributes some additional step in Karaboga's algorithm, so it can be used for solve clustering problem. Zhang appends control parameter to his algorithm and he also have some different step with Karaboga's algorithm. Zhang adds the control parameter to scout phase which called upper bound that uses as the limit of scout number and in his algorithm, the scout phase will be different with employed bees phase. In Karaboga's algorithm, the scout only finds once the random food source and act like employed bees as well. However in Zhang's algorithm, the scout will act as scout that always find the new food source randomly as long as it is scout. The scout will only change into employed bees if the limit of scout number is reached, where the worst bee will still be scout, and the others will be employed bees.

The steps of Zhang's artificial bee colony algorithm for clustering are:

Step 1: Initialize the population of solutions and its control parameter. Order the first half of colony consists of the employed bees and the second half includes onlooker bees. Generate random position for each employed bees and evaluate it. Set scout number to zero.

Step 2: If the number of scouts is more than its upper bound, order the first half of colony, make the bees with worst solution quality as scouts and

others as employed bees. Update the scout number.

Step 3: Produce new solutions for each employed bees, evaluate them, and apply the Deb's selection process. If the limit for abandonment is reached, the employed bee forgets its memory and become a scout. The scout number is adding by

Step 4: Send each scout into the search area for discovering new food sources randomly. When a new food is found, evaluate it, and apply the Deb's selection process.

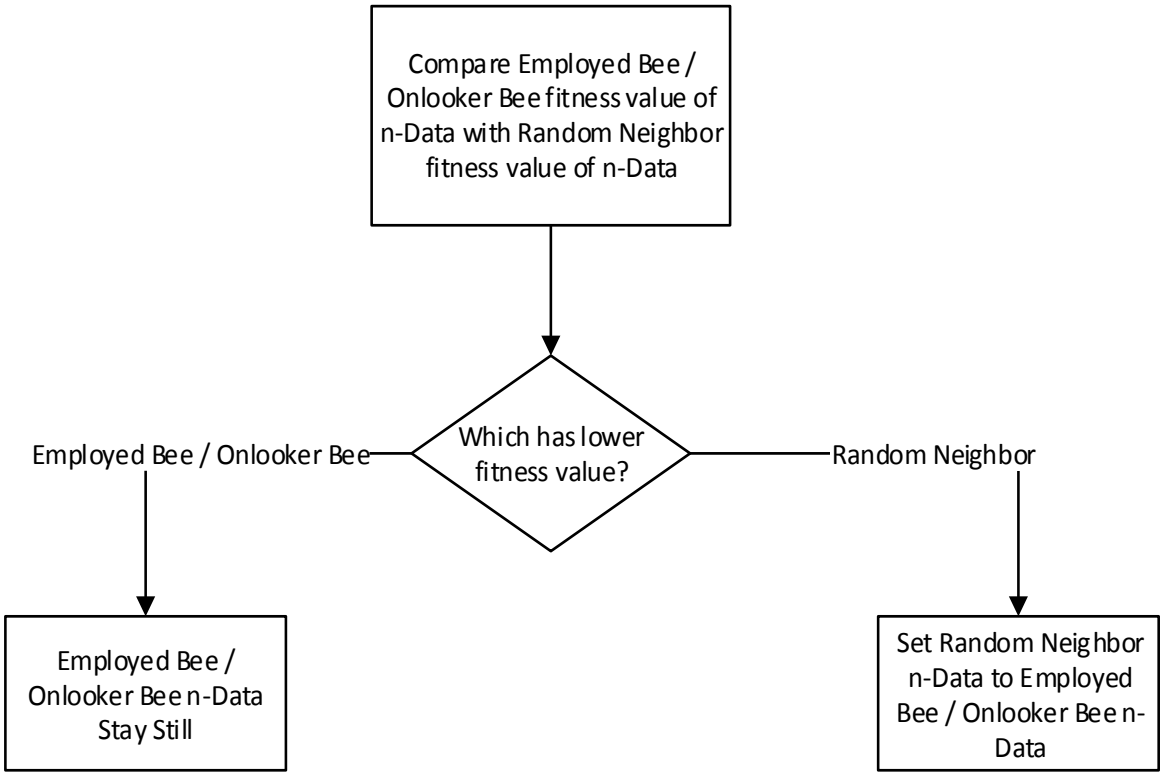

Figure 1: Greedy Selection Process.

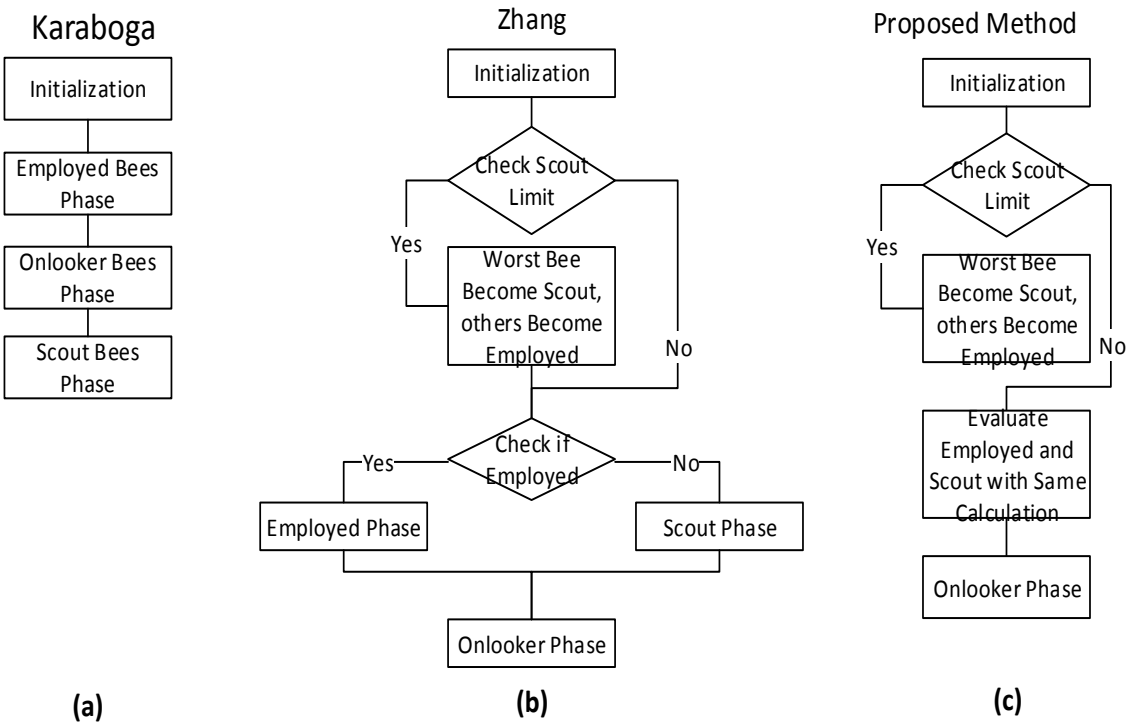

Figure 2: Comparison of ABC algorithm between (a) Karaboga's ABC (b) Zhang's ABC for clustering and (c) proposed method.

Step 5: Calculate the probability value of the current food sources with which employed bees are preferred by the onlookers.

Step 6: Produce new solutions for each onlooker bees, evaluate them, and apply the Deb's selection process to update the corresponding employed bee's memory or the current food sources.

Step 7: For each employed bee and scout, if its memorized position is better than the previous achieved best position, then the best position is replaced by it. If the termination condition is still not met, repeat the algorithm process from **Step 2**, otherwise stop the algorithm.

3 Proposed method

3.1 The concept

This section firstly is described the combination of Karaboga [8] and Zhang [7] algorithm. Most of steps uses the Zhang's algorithm, except selection process uses Karaboga's algorithm with greedy algorithm.

The greedy algorithm is the key to find the best solution. So, for the next experiment, the centroid or food source calculation is reconstructed, after the greedy algorithm has been done, whereas the employed bees compared its food source to another bee's food source as shown on Fig.1.

Besides of the greedy algorithm, the proposed metod, FABCC, also combines Zhang's algorithm and Karaboga's algorithm in scout phase. Unlike Karaboga's algorithm, bee on FABCC mimics Zhang's algorithm that the sequence process of bee is employed bee, scout bee, and then onlooker bee as shown in Fig. 2. However FABCC adopts Karaboga's algorithm calculation to determine fitness value.

3.2 ABCC and FABCC

From the literature studies in Section 2, it mentions there are three bees in this algorithm, consisting of employed

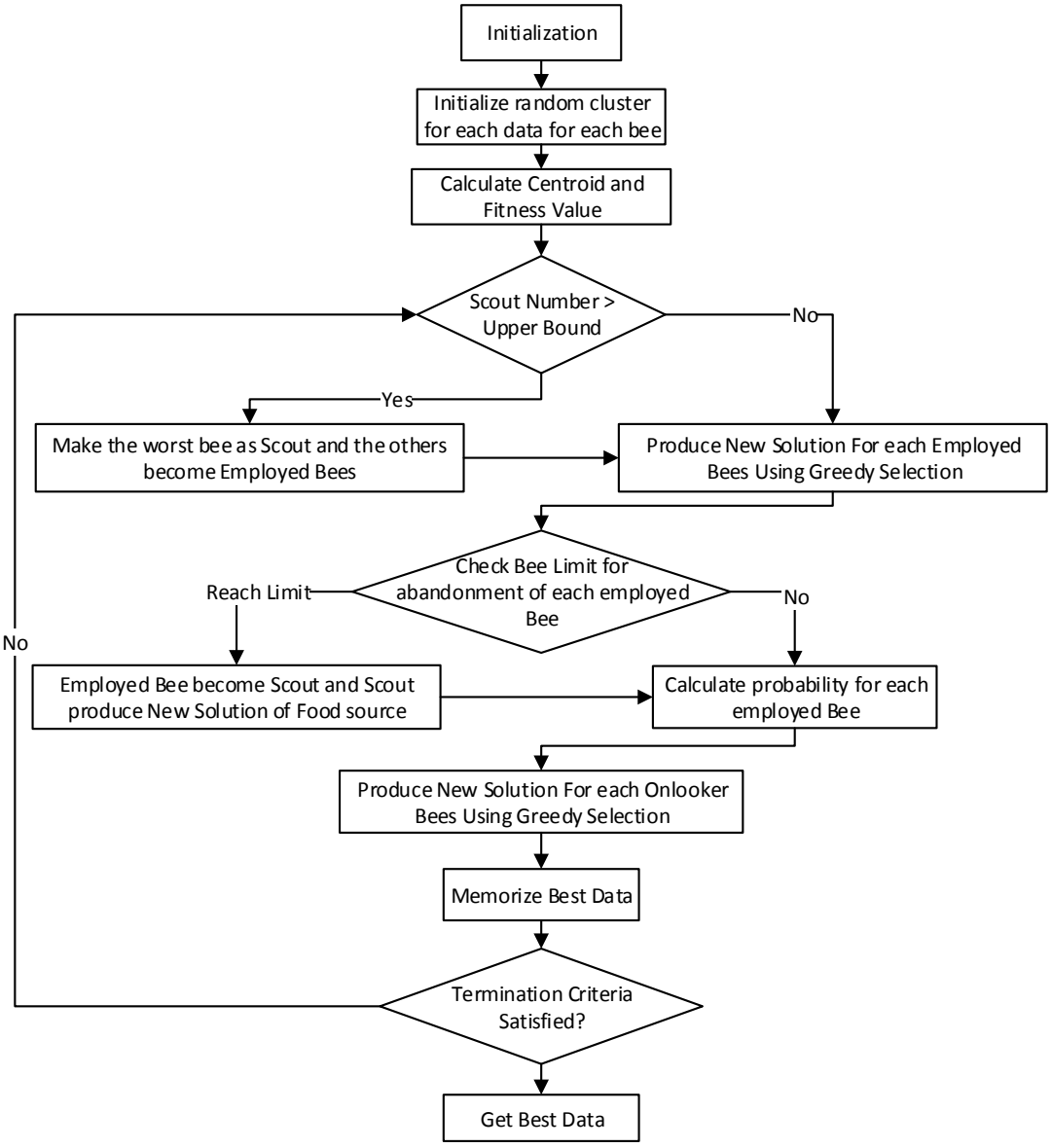

Figure 3: ABC for clustering algorithm flowchart.

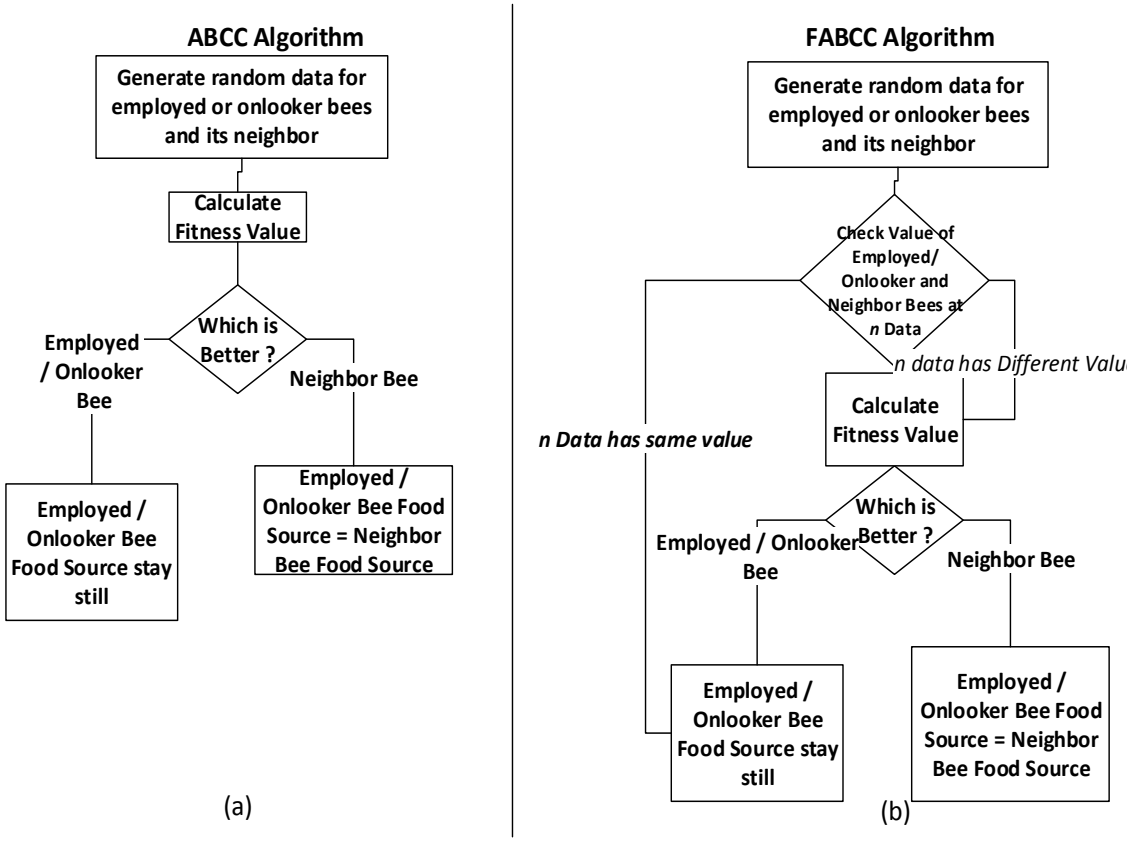

Figure 4: Comparison of employed bee / onlooker bee phase between (a) ABCC phase and (b) FABCC phase.

bee, onlooker bee, and scout bee. Which are each of them has their own fitness value. The process of searching the best fitness value consisting of eight steps and the flowchart is shown in Fig. 3.

Step 1: Initialize the population of solutions and its control parameter. Order the first half of colony consists of the employed bees and the second half includes onlooker bees. Generate random position for each employed bees and evaluate it. Set scout number to zero.

Step 2: If the number of scouts is more than its upper bound, order the first half of colony, make the bees with worst solution quality as scouts and others as employed bees. Update the scout number.

Step 3: Produce new solutions for each employed bees, evaluate them, and apply the greedy selection process.

Step 4: If the limit for abandonment is reached, the employed bee forgets its memory and become a scout for discover a search space and get the new food source randomly, and scout act like employed bees. The scout number is adding by one.

Step 5: Calculate the probability value of the current food sources with which employed bees are preferred by the onlookers.

Step 6: Produce new solutions for each onlooker bees, evaluate them, and apply the greedy selection process.

Step 7: Compare employed bees and onlooker bees which have same food source and save the best quality for each food source.

Step 8: Memorize the best food source quality overall. If the termination condition is still not met, repeat the algorithm process from **Step 2**, otherwise stop the algorithm.

In the experiment using ABCC Algorithm for big data that needs many iteration, ABCC takes too much computation time. This section explained a proposed method for Fast Artificial Bee algorithm to reduce the computation time. In Artificial Bee Colony algorithm, employed bees and onlooker evaluation phase. The comparison of employed or onlooker bee and its random neighbor is compared from their fitness value regardless the compared data is similar or not (Fig. 4.a). However in FABCC algorithm, one step is added for check the compared data. If the compared data of employed or onlooker bee and its neighbor is similar, the calculation of fitness value is skipped and the bee's data still same with the current data (Fig. 4.b).

4 Experimental results

The parameters used in fast ABC for clustering (FABCC) are shown in Table 1. The description of those parameters is as follows.

a. The number of bee is 20 which is grouped into 3 types of bee.

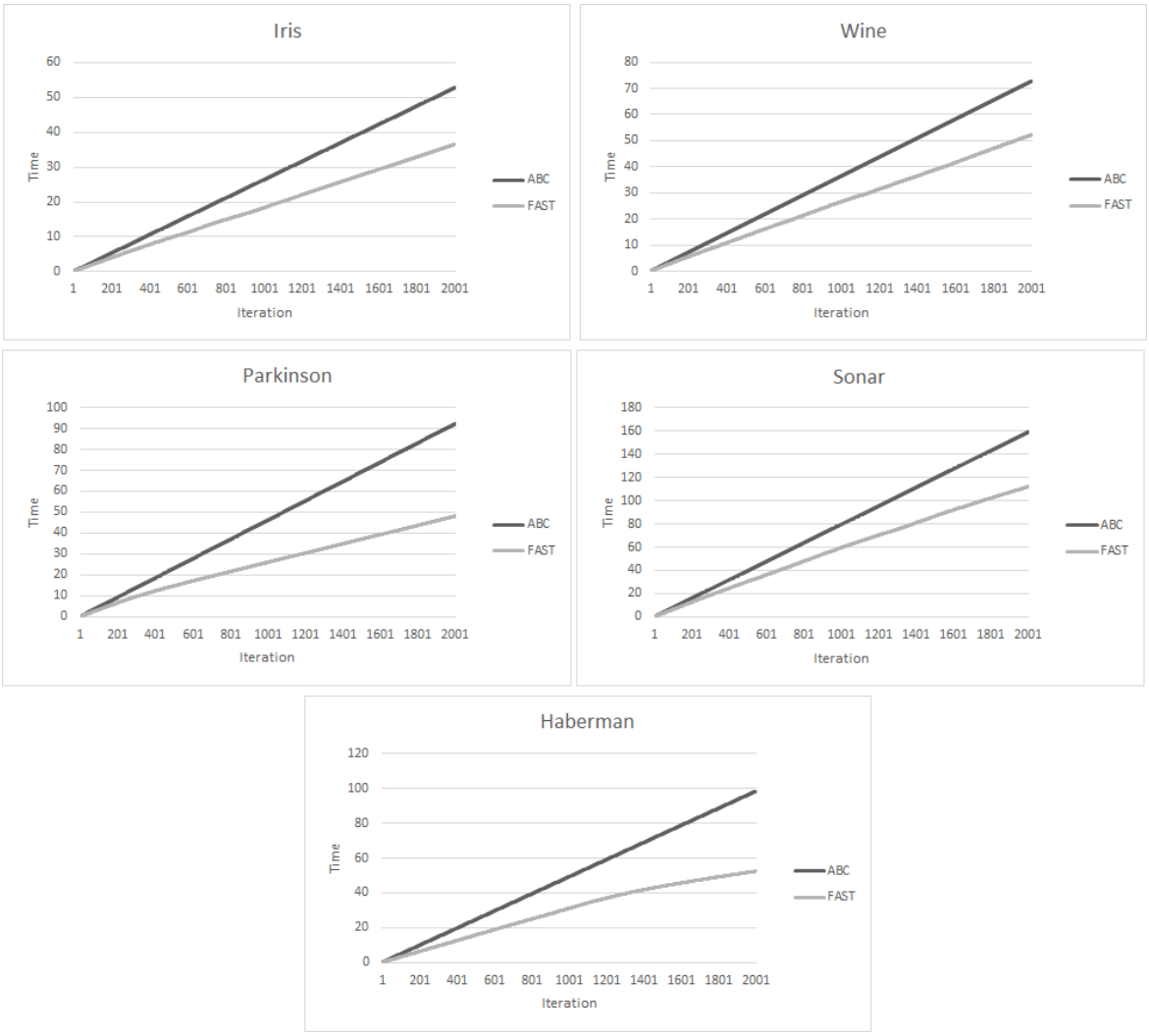

Figure 5: Duration time process of ABCC and FABCC.

b. The number of initial employed bee and the number of onlooker are 10.
c. The chance of searching better condition for employed bees means that if employed bees failed to gain better solution after 100 times consecutively, it will leave its pattern and become a scout bee to search another pattern.
d. The maximum cycle of number selected is 2000

Parameter	Value
Total Bees	20
Employed Bees	10
Scout	Up to 5
Onlooker Bees	10
Limit for Abandonment	100
Maximum Cycle Number	2000

Table 1: Parameter used for experiment.

This research uses several data sets to be evaluated as shown in Table 2. This research focuses in two aspects, quality and processing time. The quality of the program can be evalueted from the result of fitness value.

Fig. 5 shows the different computation time of original algorithm for clustering (ABCC) and modified algorithm for clustering (FABCC) for five data sets. In every data sets, the figures show that the differences of ABCC and FABCC starting in 300th iteration. It means there is no siginifance 1-200 iterations From 1st iteration computation time.

Data Set	Number of Patterns	Number of Clusters	Number of Attributes
Iris	150	3	4
Wine	178	3	13
Haberman	306	2	3
Connectionist Bench (Sonar)	208	2	60
Parkinson	195	2	22

Table 2: Data sets.

It can be divergent because in beginning the bees still generate some various pattern to be learned. After through some iterations, the preferred pattern will be created. Bees learn from the previous pattern.The pattern which has same with previous patterns in many times is identified as the repition process. Therefore, it can be pointed as the pruned pattern to prevent the redundant computation. For more detail, Fig. 6 shows that the computation time in FABCC tends to decrease in each iteration after several

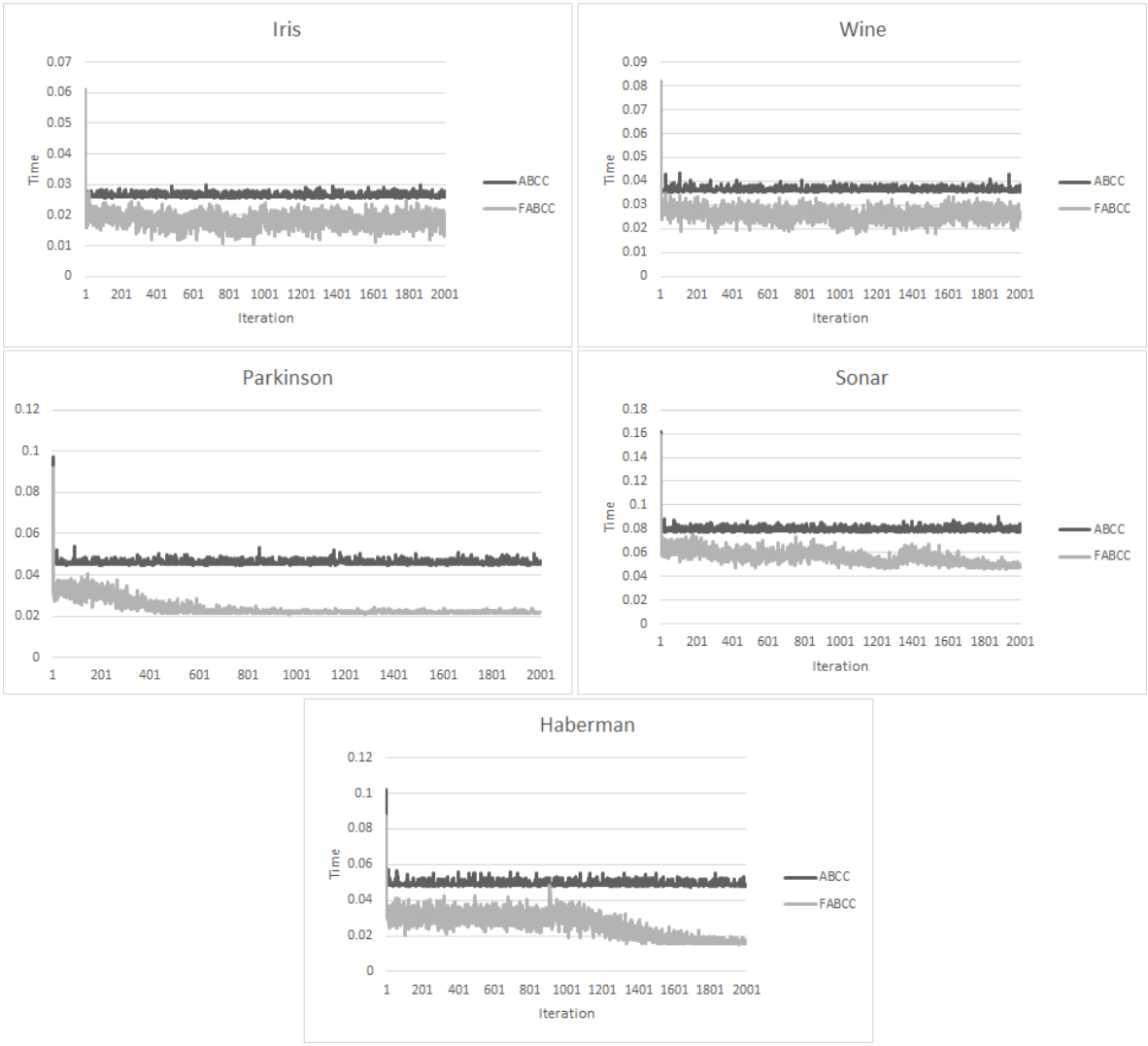

Figure 6: Computation time per iteration for ABCC and FABCC.

iteration while computation time in ABCC constant since the first iteration. In FABCC, the computation time could decrease because the bee in FABCC will learn the pattern of the data. When the algorithm generates patterns, bee will truncate some calculation. Therfore the computation time decrease after many iterations.

The results of parametes used in Table 1 are shown in Fig.7 and Table 3. Fig. 7 shows the convergent ABCC and FABCC. They start convergent after about 800 iteration. Table 3 shows the results of proposed algortithm, FABCC which is compared original algorithm, ABCC algorithm. The results are categorized two parts, computation time and quality (fitness value). This evaluations are run 30 times for each parts. Each of parts consists some test statistics such as mean, min, max, and standard deviation (SD). Min is considered as best solution while max is considered as the worst solution. All standard deviation (SD) of the results are too small comparing mean (less than 1 %). The small deviation indicates is almost same to the expected value. This study also conducted the Wilcoxon signed-rank test. The wilcoxon test is used to analyze the results of paired observations from two data (in this case results ABCC and FABCC) are different or not. The bound significant (α) is used less than 0.01. If *Asymp. Sig* < α , it indicated that these two related samples (FABCC and ABCC) are different significantly. Table 3 also shows that the difference of ABCC's and FABCC's fitness value are not significant. In best case, FABCC can achive as the rescults of ABCC in all of data sets. The FABCC is only a little outperform ABCC in mean, and worst value. The differences are only is less than 1 %. However the computation time in FABCC can be decrease significant comparing the ABCC. They can be different about 30-50 %. This means FABCC can be applied to reduce computation time as the problem of ABC as one of the heuristic algorithm. Table 3 also shows that the difference of ABCC's and FABCC's fitness value are not significant. In best case, FABCC can achive as the rescults of ABCC in all of data sets. The FABCC is only a little outperform ABCC in mean, and worst value. The differences are only is less than 1 %. However the computation time in FABCC can be decrease significant comparing the ABCC. They can be different about 30-50%. This means FABCC can be applied to reduce computation time as the problem of ABC as one of the heuristic algorithm.

Data Set	ABCC (Computation Time)				FABCC (Computation Time)				Asymp. Sig. (2-tailed) ABCC-FABCC
	Min (best)	Max (worst)	Mean	SD	Min (best)	Max (worst)	Mean	SD	
Iris	53.76	54.01	53.94	0.001	37.08	38.21	37.62	0.001	0.002
Wine	74.57	75.23	74.80	0.002	52.18	56.17	54.56	0.002	0.003
Haberman	100.76	101.2	100.91	0.002	46.42	59.22	53.91	0.012	0.005
Sonar	159.99	160.35	160.11	0.001	111.78	126.83	118.15	0.011	0.009
Parkinson	93.76	95.64	94.05	0.002	48.54	63.95	56.05	0.009	0.007
	ABCC (Fitness Value)				FABCC (Fitness Value)				
Iris	78.94	78.94	78.94	0.000	78.94	79.11	79.02	0.001	0.001
Wine	2370700	2370700	2370700	0.000	2370700	2370700	2370700	0.000	0.000
Haberman	30507	30507	30507	0.000	30507	30524	30513.4	0.004	0.002
Sonar	280.53	280.61	280.57	0.001	280.53	280.71	280.62	0.002	0.000
Parkinson	1343400	1343400	1343400	0.000	1343400	1343981	1343710	0.008	0.001

Table 3: Results of experiment.

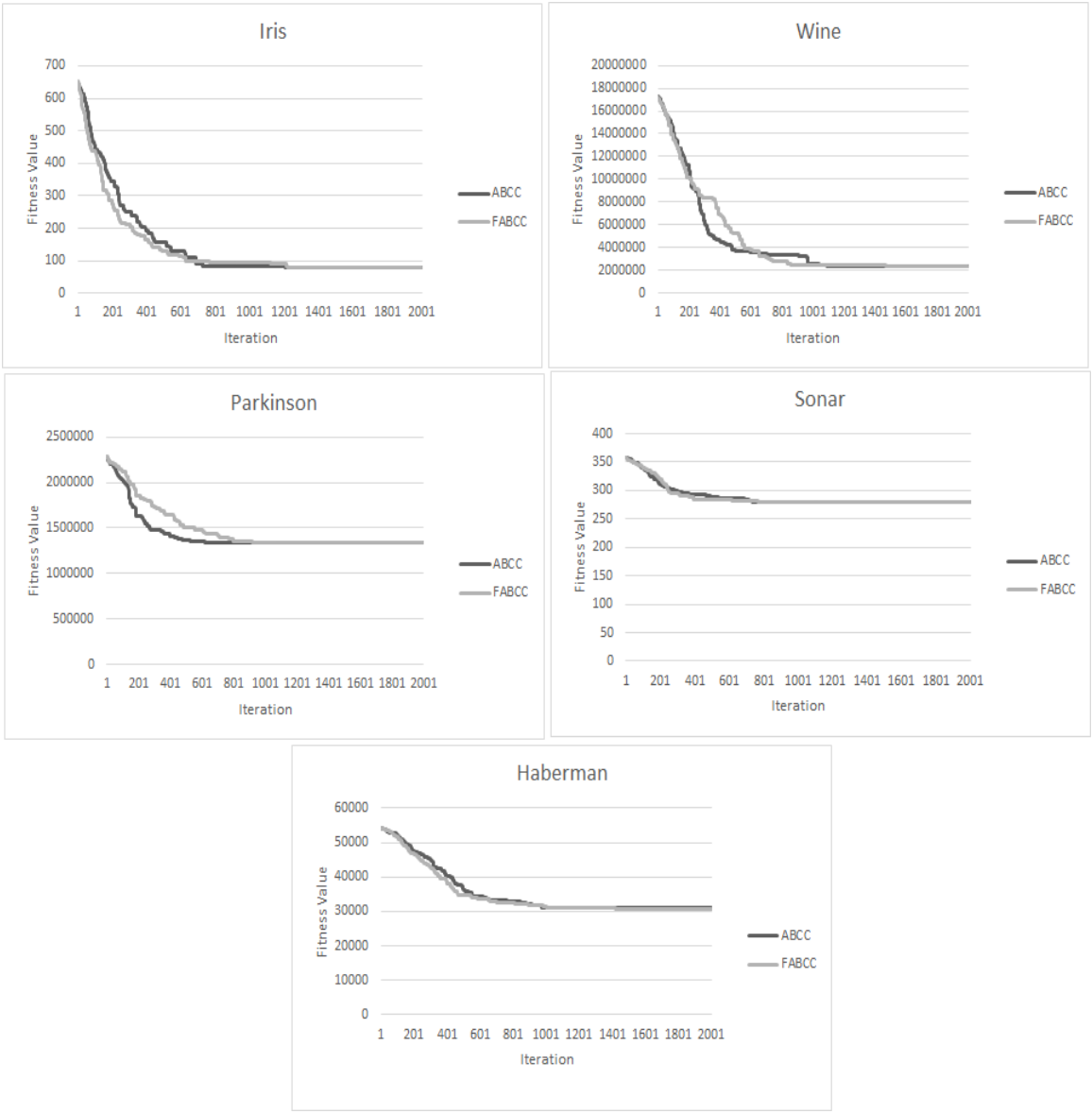

Figure 7: Fitness value of ABCC and FABCC.

5 Conclusion

This research uses bee colony algorithm by combining Zhangs and Karaboga algorithm. The choice of ABC algorithm sequence is based on Zhang and fitness value calculation is based on the original Karaboga. This proposed method, FABCC, also concludes that some redundant process occurs on bee colony algorithm for clustering. The redundant process identifies as a pattern that is able to compress. The result shows FABCC is effective to reduce computation time. It can be proved by conducts five datasets Iris, Wine, Haberman, Sonar, and Parkinson. The results shows that it can reduce 30-50% of computation time, while the fitness value only reduce less than 1%.

This study focuses on only small five datasets for clustering. It can be extended using the other big datasets. It might be have the different characteristic. However in our exploration, the redundant process always occurs in most of metaheuristic algorithm. In next research, researchers can put their effort to remake calculation of fitness value, in calculating in fitness value there are many iterations and redundant calculation to be observed to prune the redundant pattern.

6 References

[1] a. K. Jain, M. N. Murty, and P. J. Flynn, "Data clustering: a review," *ACM Comput. Surv.*, vol. 31, no. 3, pp. 264–323, 1999.

[2] Y. Yang and M. Kamel, "Clustering ensemble using swarm intelligence," *Swarm Intell. Symp. 2003. SIS '03. Proc. 2003 IEEE*, pp. 65–71, 2003.

[3] T. Zhang, R. Ramakrishnan, and M. Livny, "BIRCH: An Efficient Data Clustering Method for Very Large Databases," *Proc. 1996 ACM SIGMOD Int. Conf. Manag. Data*, vol. 1, pp. 103–114, 1996.

[4] B. Akay and D. Karaboga, "A modified Artificial Bee Colony algorithm for real-parameter optimization," *Inf. Sci. (Ny).*, vol. 192, pp. 120–142, 2012.

[5] A. Ouaarab, B. Ahiod, and X.-S. Yang, "Discrete cuckoo search algorithm for the travelling salesman problem," *Neural Comput. Appl.*, vol. 24, no. 7–8, pp. 1659–1669, 2013.

[6] S.-M. Chen and C.-Y. Chien, "Solving the traveling salesman problem based on the genetic simulated annealing ant colony system with particle swarm optimization techniques," *Expert Syst. Appl.*, vol. 38, no. 12, pp. 14439–14450, 2011.

[7] C. Zhang, D. Ouyang, and J. Ning, "An artificial bee colony approach for clustering," *Expert Syst. Appl.*, vol. 37, no. 7, pp. 4761–4767, 2010.

[8] D. Karaboga and C. Ozturk, "A novel clustering approach: Artificial Bee Colony (ABC) algorithm," *Appl. Soft Comput.*, vol. 11, no. 1, pp. 652–657, 2011.

[9] S. Goel, A. Sharma, and P. Bedi, "Cuckoo Search Clustering Algorithm: A novel strategy of biomimicry," *2011 World Congr. Inf. Commun. Technol.*, pp. 916–921, 2011.

[10] S. Rana, S. Jasola, and R. Kumar, "A review on particle swarm optimization algorithms and their applications to data clustering," *Artif. Intell. Rev.*, vol. 35, no. 3, pp. 211–222, 2011.

[11] C.-L. Huang, W.-C. Huang, H.-Y. Chang, Y.-C. Yeh, and C.-Y. Tsai, "Hybridization strategies for continuous ant colony optimization and particle swarm optimization applied to data clustering," *Appl. Soft Comput.*, vol. 13, no. 9, pp. 3864–3872, 2013.

[12] R. Storn and K. Price, "Differential evolution–a simple and efficient heuristic for global optimization over continuous spaces," *J. Glob. Optim.*, pp. 341–359, 1997.

[13] A. S. Girsang, C.-W. Tsai, and C.-S. Yang, "A Fast Bee Colony Optimization for Traveling Salesman Problem," *2012 Third Int. Conf. Innov. Bio-Inspired Comput. Appl.*, vol. 1, no. c, pp. 7–12, 2012.

[14] Y. Lu, S. Lu, F. Fotouhi, Y. Deng, and S. Brown, "FGKA: A fast genetic k-means clustering algorithm," *Proc. 2004 ACM …*, pp. 1–2, 2004.

[15] K. D. David E. Goldberg, "A comparative analysis of selection schemes used in genetic algorithms."

[16] J. Han and M. Kamber, "Data Mining: Concepts and Techniques," *Ann. Phys. (N. Y).*, vol. 54, p. 770, 2006.

[17] T. Kanungo, D. M. Mount, N. S. Netanyahu, C. D. Piatko, R. Silverman, and a. Y. Wu, "An efficient k-means clustering algorithm: analysis and implementation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 24, no. 7, pp. 881–892, 2002.

[18] J. a Hartigan and M. a Wong, "Algorithm AS 136: A k-means clustering algorithm," *Appl. Stat.*, vol. 28, no. 1, pp. 100–108, 1979.

[19] D. Karaboga and B. Basturk, "A powerful and efficient algorithm for numerical function optimization: artificial bee colony (ABC) algorithm," *J. Glob. Optim.*, vol. 39, no. 3, pp. 459–471, 2007.

[20] D. Karaboga, B. Gorkemli, C. Ozturk, and N. Karaboga, "A comprehensive survey: artificial bee colony (ABC) algorithm and applications," *Artif. Intell. Rev.*, vol. 42, no. 1, pp. 21–57, 2014.

[21] D. Karaboga and B. Basturk, "On the performance of artificial bee colony (ABC) algorithm," *Appl. Soft Comput.*, vol. 8, no. 1, pp. 687–697, 2008.