OPEN ACCESS
Indo-JC Indonesia Journal of Computing
ISSN 2460-9056
socj.telkomuniversity.ac.id/indojc

Ind. Journal on Computing
Vol. 5, Issue. 1, March 2020. pp. 85-92
doi:10.21108/indojc.2020.5.1.382

Determining *N*-Days Tourist Route Using Swap Operator Based Artificial Bee Colony Algorithm

Ayunda Farah Istiqamah [#1], Z. K. Abdurahman Baizal [#2], Yusza Reditya Murti [#3]

 # School of Computing, Telkom University
Jl. Telekomunikasi 1, Bandung, 40257, Indonesia

[1] f.ayunda@gmail.com
[2] baizal@telkomuniversity.ac.id
[3] yuszaa@telkomuniversity.ac.id

Abstract

Traveling is one of the activities chosen by many people to spend holidays. Some tourists want to go on vacation in a place they have never visited before, so they need a tool to plan a tour. Planning this tour includes determining tourist route. We analogize the determination of tourist routes using Traveling Salesman Problem (TSP). The main purpose of this study was to find the optimal tourist route using Swap Operator Based Artificial Bee Colony Algorithm. We use Multi-Attribute Utility Theory (MAUT) to accommodate user needs for the route that recommended by the system. The criteria for user preferences used in this study are: 1) routes with as many tourist attractions as possible, 2) routes that pass popular destinations, and 3) routes with minimal costs. Based on the experiment results, Swap Operator Based Artificial Bee Colony gives more optimal results than the Simulated Annealing, especially in terms of the number of tourist attractions (nodes) that can be visited in one trip.

Keywords: Multi-Attribute Utility Theory, Swap Operator Based Artificial Bee Colony Algorithm, Traveling Salesman Problem

Abstrak

Berwisata merupakan salah satu kegiatan yang dipilih banyak orang untuk mengisi waktu libur. Beberapa wisatawan ingin pergi berlibur ke tempat yang sebelumnya tidak mernah mereka kunjungi, sehingga mereka memerlukan suatu alat untuk merencanakan perjalanan wisata. Perencanaan perjalanan wisata ini meliputi penentuan rute wisata. Rute wisata ini dapat dianalogikan sebagai sebuah Traveling Salesman Problem (TSP). Tujuan utama penelitian ini adalah untuk menemukan rute wisata yang optimal menggunakan algoritma Swap Operator Based Artificial Bee Colony. Kami menggunakan Multi-Attribute Utility Theory (MAUT) untuk mengakomodasi kebutuhan pengguna terhadap rute yang direkomendasikan oleh sistem. Kriteria preferensi pengguna yang digunakan dalam penelitian ini adalah: 1) rute dengan sebanyak mungkin tempat wisata yang dikunjungi, 2) rute yang melewati destinasi populer, dan 3) rute dengan biaya yang minimal. Berdasarkan hasil pengujian, algoritma Swap Operator Based Artificial Bee Colony memberikan hasil yang lebih optimal dibanding algoritma Simulated Annealing, terutama dari segi jumlah destinasi wisata yang dapat dikunjungi dalam satu perjalanan.

Kata Kunci: Multi-Attribute Utility Theory, Swap Operator Based Artificial Bee Colony Algorithm, Traveling Salesman Problem

Received on Jan 2020. Accepted on March 2020

I. INTRODUCTION

TRAVELING is one of the activities chosen by many people to spend holidays. Some tourists want to go on vacation in a place they have never visited before, so they need a travel route planning where tourists can choose for themselves the tourist attractions they want to visit based on their preferences.

We analogize the determination of tourist routes using Traveling Salesman Problem (TSP). TSP is the combinatorial optimization problem to find the shortest route from a series of cities. TSP is very concerned about time complexity so that it is included in the Nondeterministic Polynomial-hard (NP-hard) [1]. TSP is widely used in finding optimal solutions in route determination [2]. Some algorithms that have been used to test the TSP are Particle Swarm Optimization (PSO) [3], Firefly Algorithm (FA) [4], Simulated Annealing (SA) [2], and Genetic Algorithm (GA) [5].

The main objective of the research in this paper is to find the optimal tourist route using the Swap Operator Based Artificial Bee Colony Algorithm. We chose this algorithm because it can solve combinatorial optimization problems and provide good performance [6]–[8]. Swap operators were first introduced by Wang et al. [9]. In 2003 they applied the swap operator to Particle Swarm Optimization (PSO) to solve TSP. In 2012, Li et al. [6] also implements swap operator to Discrete Artificial Bee Colony (DABC) Algorithm to solve TSP.

We also use Multi-Attribute Utility Theory (MAUT) to accommodate user needs for the route that recommended by the system. Users can express their needs through three criteria, that are: 1) routes with as many tourist attractions as possible, 2) routes that pass popular destinations, and 3) routes with minimal costs. The results of the MAUT are then used as fitness values in this study. The system takes into account the opening and closing hours of each destination that user wants to visit in determining tourist routes and in this study can only form a maximum route of three days, where each day starts at 08:00 until 18:00.

II. ARTIFICIAL BEE COLONY ALGORITHM

In 2005, Dervis Karaboga introduced Artificial Bee Colony (ABC) Algorithm [10]. ABC is included in swarm intelligence, which is an intelligence based on the behavior of a population that is able to control itself [11]. The algorithm is inspired by the behavior of bee colonies in finding food sources. There are 3 types of bees on ABC, namely employed bee, onlooker bee, and scout bee.

The ABC algorithm process begins with random food sources initialization. Employed bees will exploit a food source and bring the results back to the hive. They will look for the best solution in the food source area and calculate the fitness value. Old food sources will be replaced with new food sources if the fitness value of the new sources is better than the old ones. In the bees hive, they will perform a waggle dance. Onlooker bees will choose the dance that they think is the most interesting based on the probability value using Equation (1). We use Equation (1) to get the best probability value from the food sources chosen by the bees. They will then go to the food source to look for food in the surrounding area and bring the results back to the hive. When the source has been exploited, scout bees will explore new food sources and bring the information back to the bee hive.

$$Pi = \frac{fit_i}{\sum_{k=1}^{N} fit_k} \tag{1}$$

Where fit_i is the i food source fitness value and fit_k is the k food source fitness value with $k = (1, 2, \ldots, N)$.

III. SWAP OPERATOR BASED ARTIFICIAL BEE COLONY ALGORITHM

Swap operators are used to exchange two cities on a set of TSP routes. For example there is a route with eight cities *A = (1, 2, 3, 4, 5, 6, 7, 8)* and there is a swap operator *SO (2,8)* then the new solution that will be formed is *A' = A + SO (2,8) = (1, 8, 3, 4, 5, 6, 7, 2)*. A collection of swap operator is called swap sequence. Each swap operator in a swap sequence will work in order. For example there are swap sequence $SS = (SO_1, SO_2, \ldots, SO_N)$ then the new solution to be formed is $A' = A + SS = A + (SO_1, SO_2, \ldots, SO_N) = ((A + SO_1) + SO_2) + \ldots + SO_N$.

Based on [6] the steps of Swap Operator Based Artificial Bee Colony Algorithm on TSP consist of five phases, *Initial Phases*, *Employed Bees Phases*, *Onlooker Bees Phases*, *Scout Bees Phases*, and *Stop Criteria* which will be explained as follows.

1) Initial Phase: At this phase we initialize first population and all parameters that needed in the study, such as colony size, problem size, limit, and max iteration. Each bee represents a route where the starting point and end point of the route is a hotel.

2) Employed Bees Phase: The food sources from the first population will be exploited by each employed bees using swap operator. Each bee will form a new solution to get a better fitness value than the previous solution. The old solution will be replaced with a new solution if the fitness value of the new solution is better. Limit added by 1 if a bee do not get a better solution.

3) Onlooker Bees Phase: Based on the probability value obtained from Equation (1), onlooker bee chooses a food source from employed bee. Each bee will form a new solution with swap operator to get a better fitness value than previous solution. The previous solution will be replaced with a new solution if the fitness value of the new solution is better. Limit added by 1 if a bee do not get a better solution.

4) Scout Bees Phase: Calculate the number of limit with L. If L > limit then discard the solution and scout bee will create a new solution randomly. If L < limit then the previous solution is still used. Memorize the best solution that has been obtained so far.

5) Stop Criteria: This process is declared complete if it has reached max iteration. Otherwise, return to the *Employed Bees Phase*.

IV. MULTI-ATTRIBUTE UTILITY THEORY

Multi-Attribute Utility Theory (MAUT) is used to calculate the level of user satisfaction based on consideration of the value of each object [12]. In this study we use MAUT to get a fitness values from each tourist attractions. Equation (2) describes the value of an object with *n* attributes which are used as fitness value in this study.

$$U(X) = \sum_{i=1}^{n} w_i u_i(x_i) \tag{2}$$

Where x_i is the *i* attribute, u_i is the value of the *i* attribute, and w_i is the weight of the *i* attribute.

To calculate MAUT as a fitness value we carry out the normalization process so that the values of the attributes used in the calculation are aligned with a certain range. In this study, we use the min-max normalization in the range [0,1] described by (3).

$$x_{norm} = \frac{x - x_{min}}{x_{max} - x_{min}} \tag{3}$$

Where x_{max} is the maximum value in the data, x_{min} is the minimum value of the data, and *x* is the value of the attribute.

In this study we used three attributes, there are travel time, rating, and cost. Travel time attribute represents the routes with as many tourist attractions as possible criteria. The shorter the travel time, the more tourist attractions that can be visited. Rating attribute represents the routes that pass popular destinations criteria, we choose the highest rating to get popular destinations. Cost attribute represents routes with minimal costs, we choose the lowest cost from each tourist attraction to get the routes with minimal costs. This explains that for the rating attribute we chose the highest value, while for the travel time and cost attribute we chose the lowest. In order to take the highest value of each attribute, we use (4) for the travel time and cost attribute after normalization.

$$x = |1 - x_{norm}| \tag{4}$$

Where x_{norm} is the attribute value after normalization.

V. Experiments

In this section, we conduct parameter test and performance test based on the number of tourist attractions (nodes), number of days of visits, and running time. In this study, we took a case study of Bandung Raya using 20 tourist attractions and a maximum of three days of visiting days. The results of this test are then compared with the Simulated Annealing (SA) Algorithm from the previous study [2]. We chose to compare with SA because in the previous study it used multi attributes with the same criteria as we used in this study and it can also solve combinatorial problems.

A. Parameter Test

We performed a parameter test to obtain the best parameters used in the study, such as colony size, problem size, limit, and max iteration. In this study, we use two static parameters, limit = 50 and problem size = 20 tourist attractions. The test is done by combining each parameter, where the value for colony size = {10, 50, 100, 150} and the value for max iteration = {10, 50, 100, 150}. Each combination is tested 10 times to get the best results.

TABLE I
Parameter Test

Colony Size	**Max iteration**	**Fitness**	**Running Time**	**Number of Nodes in Tourist Routes**
10	10	0.6835	0.0935	12
50	50	0.6868	1.0501	13
50	100	0.6818	2.0211	14
100	50	0.7299	2.3938	10
100	100	0.7234	4.1890	12
100	150	0.6811	6.0598	13
150	100	0.7064	6.7082	12
150	150	0.6990	10.2538	13

Table 1 shows a number of combinations of parameters that we have tested. We choose the best parameter based on the highest fitness value with running time in less than five seconds. The fitness values obtained from Equation (2) based on criteria as described in previous section. We get the best parameter with a combination of colony size = 100 and max iteration = 50.

B. Performance Test

We conducted a performance test by testing inputs from 2 to 20 nodes, each of which was tested 10 times. This performance test is carried on three variables, based on the number of tourist attractions (nodes), number of days of visits, and running time.

1) Performance Test Based on Number of Tourist Attractions (Nodes)

Fig. 1 shows the results of performance test based on the number of tourist attractions (nodes). In the number of 2 to 4 input nodes, there is no significant differences between the two algorithms. The only difference is in the input of 2 nodes, the Swap Operator Based Artificial Bee Colony gives better results. Meanwhile, in the input of 5 to 20 nodes using the Swap Operator Based Artificial Bee Colony produces a greater number of nodes than the Simulated Annealing, with the highest number of nodes produced in one tour is 13,3 nodes while the Simulated Annealing is 10,9 nodes. Swap Operator Based Artificial Bee Colony works by calculating all possible routes until it gets the best fitness value. It processes all input nodes that produces routes with the best fitness value and meet the specified time duration.

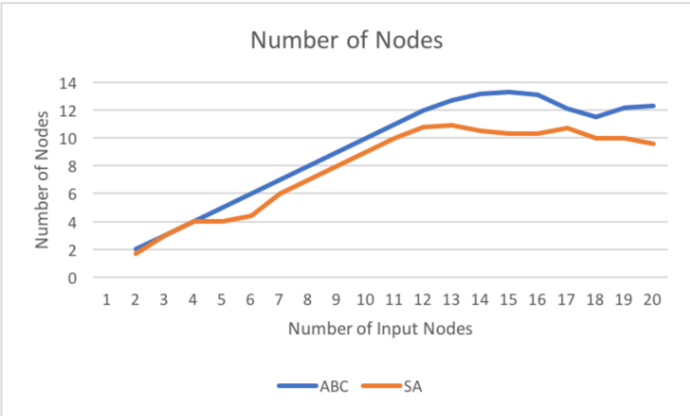

Fig. 1 Performance Test Based on Number of Tourist Attractions (Nodes)

2) Performance Test Based on Number of Days of Visits

Fig. 2 shows the performance test results based on the number of days of visits. Fig. 2 shows that between the Swap Operator Based Artificial Bee Colony and Simulated Annealing there is no significant difference. It happens because of the same input for both algorithm and the specified number of days, which is a maximum of 3 days. In the input of 2 to 5 nodes, the total number of days of visit was 1 day, in the input of 7 to 9 nodes were 2 days, and in the input of 10 to 20 nodes were 3 days. The only difference is in the input of 6 nodes, where the Swap Operator Based Artificial Bee Colony shows that total days of visits are 2 days on average and Simulated Annealing is 1,4 days.

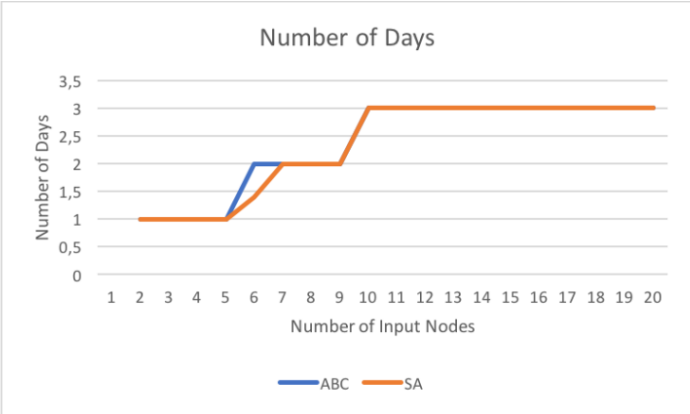

Fig. 2 Performance Test Based on Number of Days of Visits

3) Performance Test Based on Running Time

The performance test results based on running time are shown in Fig. 3. This shows that the average running time in the Swap Operator Based Artificial Bee Colony is longer than the Simulated Annealing. This is because in the Swap Operator Based Artificial Bee Colony, every process of the algorithm only provides a route for one day of visit. If the number of nodes can reach 3 days of visit, the algorithm is processed 3 times to get a 3 days tour. If the input nodes increases, the running time is also takes longer. This is done to get a better fitness value from each day of tourist visits.

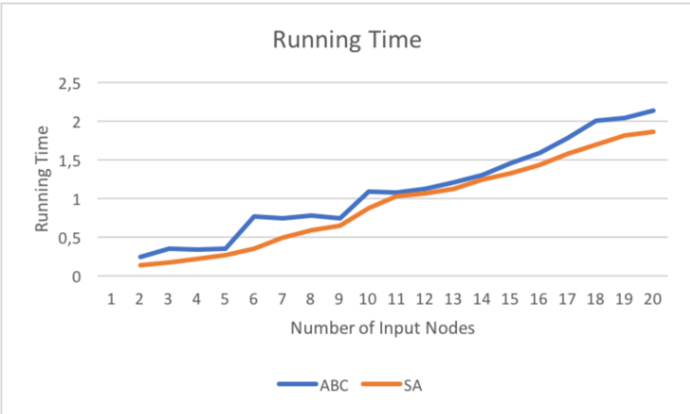

Fig. 3 Performance Test Based on Running Time

V. Conclusion

Based on the test results, we can concluded that Swap Operator Based Artificial Bee Colony Algorithm can be applied to determining the optimal *n*-days tourist route, where in this study we only limit it to 3 days. It also shows that this algorithm gives better results compared to Simulated Annealing. From the performance test result based on number of tourist attractions (nodes) it can be seen that Swap Operator Based Artificial Bee Colony Algorithm gives the higher number of nodes. This because this algorithm works by finding the route with the biggest fitness value and many nodes that fits to the duration of the tour that has been determined. In terms of number of days of visits, there were no significant differences between both algorithm, only one number of input nodes that did not give the same results. But in terms of running time, Swap Operator Based Artificial Bee Colony Algorithm takes longer than Simulated Annealing because the process of the algorithm only provides a route for one day of visit. If the number of nodes can reach 3 days of visits, the algorithm is processed 3 times to get a 3 days tour, and if the input nodes increases, the running time is also takes longer. Summary of the comparison of two algorithms is shown in Table 2. It shows the average number of the tourist attractions (nodes), days of visits, and running time from each algorithm.

TABLE II
SUMMARY OF THE COMPARISON

	Swap Operator Based Artificial Bee Colony	Simulated Annealing
Number of Tourist Attractions (nodes)	9,33	7,90
Number of Days of Visits	2,36	2,33
Running Time	1,11 (sec)	0,94 (sec)

REFERENCES

[1] M. A. P. Muniandy, L. K. Mee, and L. K. Ooi, "Efficient route planning for travelling salesman problem," *ICOS 2014 - 2014 IEEE Conf. Open Syst.*, pp. 24–29, 2014.
[2] Z. K. A. Baizal, K. M. Lhaksmana, A. A. Rahmawati, M. Kirom, and Z. Mubarok, "Travel route scheduling based on user ' s preferences using simulated annealing," 2018.
[3] Y. Shigehiro, T. Katsura, and T. Masuda, "An Application of Particle Swarm Optimization to Traveling Salesman Problem," *SICE Annu. Conf. 2010*, pp. 1629–1632, 2010.
[4] S. S. N. Kumbharana, G. Pandey, S. N, and P. G. M. Pandey, "Solving Travelling Salesman Problem using Firefly Algorithm," *Int. J. Res. Sci. Adv. Technol.*, vol. 2, no. 2, pp. 053–057, 2013.
[5] F. H. Prabowo, "A Multi-Level Genetic Algorithm Approach for Generating Efficient Travel Plans," *2018 6th Int. Conf. Inf. Commun. Technol.*, vol. 0, no. c, pp. 86–91, 2018.
[6] L. Li, Y. Cheng, L. Tan, and B. Niu, "A Discrete Artificial Bee Colony Algorithm for TSP Problem," pp. 566–573, 2012.
[7] D. Karaboga and B. Akay, "A comparative study of Artificial Bee Colony algorithm," *Appl. Math. Comput.*, vol. 214, no. 1, pp. 108–132, 2009.
[8] D. Karaboga, "A Combinatorial Artificial Bee Colony Algorithm for Traveling Salesman Problem," pp. 50–53, 2011.
[9] K. Wang, L. A. N. Huang, C. Zhou, and P. G. We, "PARTICLE SWARM OPTIMIZATION FOR TRAVELING SALESMAN PROBLEM," no. November, pp. 1583–1585, 2003.
[10] D. Karaboga, "AN IDEA BASED ON HONEY BEE SWARM FOR NUMERICAL OPTIMIZATION," *J. Biol. Chem.*, vol. 280, no. 40, pp. 33960–33967, 2005.
[11] 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.
[12] L. Chen and P. Pu, "Critiquing-based recommenders: Survey and emerging trends," *User Model. User-adapt. Interact.*, vol. 22, no. 1–2, pp. 125–150, 2012.