

HABCO: A Robust Agent on Hybrid Ant-Bee Colony Optimization

Abba Suganda Girsang^{*1}, Chun-Wei Tsai², and Chu-Sing Yang²

¹Master in Computer Science, Bina Nusantara University, Jakarta, Indonesia

²Department of Computer Science and Engineering, National Chung-Hsing University, Taichung, Taiwan

³Institute of Computer and Communication Engineering, Department of Electrical Engineering, National Cheng Kung University, Tainan, Taiwan

*Corresponding author, e-mail: agirsang@binus.edu

Abstract

The purpose of this research is to generate a robust agent by combining bee colony optimization (BCO) and ELU-Ants for solving traveling salesman problem (TSP), called HABCO. The robust agents, called ant-bees, firstly are grouped into three types scout, follower, recruiter at each stages. Then, the bad agents are high probably discarded, while the good agents are high probably duplicated in earlier steps. This first two steps mimic BCO algorithm. However, constructing tours such as choosing nodes, and updating pheromone are built by ELU-Ants method. To evaluate the performance of the proposed algorithm, HABCO is performed on several benchmark datasets and compared to ACS and BCO. The experimental results show that HABCO achieves the better solution, either with or without 2opt.

Keywords: Ant Colony Optimization, Bee Colony Optimization, Hybrid, Robust Agent, Traveling Salesman Problem

Copyright © 2017 Universitas Ahmad Dahlan. All rights reserved.

1. Introduction

Intelligent algorithm is popular to solve several optimization problems because of its performance [1-11]. Some intelligent algorithms such as ant colony optimization [1-2], genetic algorithm [3], particle swarm optimization [4-5], simulated annealing [6], cat swarm optimization [7], bee colony optimization [8-10], and so forth were proposed to solve travelling salesman problem (TSP).

In ant colony optimization (ACO), ants lay a pheromone trail in their paths while traveling from their nest to food source [1] to find the shortest tour for the food. Since then, many researchers also continued this approach for better results on various optimization problems [11-13]. In [12] proposed an adaptive parameter control scheme for developing an adaptive ACS namely AACS. The parameters value in AACS are adaptively controlled based on information of pheromone trails distribution in each iteration that is utilized to estimate the optimization state.

Naimi and Taherinejad proposed ELU-Ants and KCC-Ants to modify ACS by introducing new local update [13]. This research emphasized that the update pheromone on earlier step is more than the later step. The other researchers presented a combining ACO with the other methods [14-15]. Lucic et al [16-17] then proposed another intelligent algorithm bee colony optimization (BCO), which is designed to create the multi-agent system (colony of artificial bees) for solving the combinatorial optimization problems [17-24].

This research is tried to adopt ELU-Ants algorithm which considers the beginning step as the more important step to construct the tour. Logically, an agent with bad choice of traveling some cities (part of tour) should be discarded but an agent with good choice should be duplicated at the beginning step. The discarded and duplicated agent in earlier step guarantees the agent will be more competitive. Therefore the update pheromone on earlier step is more than the later step. In spite of starting with good or bad choice, ants must continue selecting a path for completing their tour.

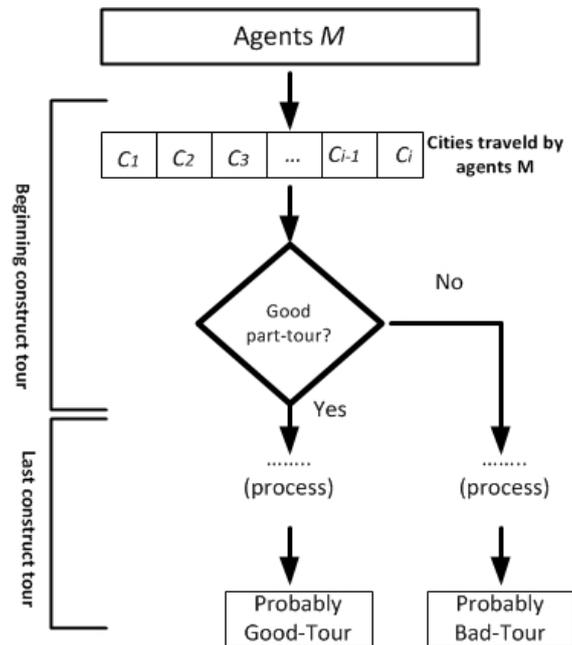


Figure 1. Illustration agent M travels cities with two step constructions.

Fig. 1 describes the agents M constructs the tour which is split into two important step constructing. Suppose in beginning step, agents M construct the part tour which consists i cities. Each agents travels cities $C_1, C_2, C_3, \dots, C_i$ at beginning step. While constructing the tour, the ant is possible to get bad choice at first stage and leads trapped on bad constructing as well. At last it probably brings to the bad-tour.

In BCO, after traveling some nectars or stage, each bees backs to the hive to communicate their way. In the hive, they group into three type and perform waggle dance to identify their quality of food source. The longer dance indicates the better quality of nectars found. Bees exchange information to each other for their next stage-tour. Bees finding bad-source food tend to follow the others with good-source food at next stage. This step shows that BCO discards bee with bad stage-tour in order to get the better tour for the next stage, and in other side automatically the bee with good stage-tour is duplicated.

So based on discussion above, the proposed algorithm, HABCO, is generated by constructing the robust agent as BCO algorithm, however the process of selecting nodes (cities) is done by ELU-Ants. HABCO was first presented by Abba et al in a proceeding of a conference [25]. However this paper presented more detail of the related work and processing of judgment agent.

2. Related Work

2.1. Ant Colony System

Ant colony system (ACS) is one of the most widely used and best-performing ACO. While ants search food, they will deposit the pheromone on the path where they passed. The ants will tend to choose the path whose the highest amount pheromone on it, and avoid the low pheromone path. There are some substantial parts to describe ACS briefly [1].

Construction Tour : While the k th ant is in city i , the next city j is selected from the unvisited city set $J_k(i)$ according to the two possibilities, $P_k(i,j)$, in Eq. (1) and in Eq. (2) :

a. if $q \leq q_0$ (exploitation)

$$j = \arg \max [\tau(i, u) \cdot \eta(i, u)^\beta], \mu \in J_k(i), \quad (1)$$

b. if $q > q_0$ (bias exploration)

$$P_k(i, j) = \frac{[\tau(i, j)] \cdot [\eta(i, j)]^\beta}{\sum_{\mu \in J_k(i)} [\tau(i, j)] \cdot [\eta(i, j)]^\beta}, \quad (2)$$

where q is a random number ; q_0 is a parameter; τ represents pheromone value. *Local pheromone updating* : Ants visit edge i, j and change the pheromone $\tau(i, j)$ edge level by local updating rule in Eq. (3) :

$$\tau(i, j) = (1 - \rho)\tau(i, j) + \rho\Delta\tau(i, j), \quad (3)$$

Global pheromone updating: The pheromone of edges on the globally best tour is updated by global updating rules Eq. (4) and Eq. (5).

$$\tau(i, j) = (1 - \alpha)\tau(i, j) + \alpha\Delta\tau(i, j), \quad (4)$$

$$\Delta\tau(i, j) = \begin{cases} (L_{gb})^{-1} & \text{if } (i, j) \in \text{global best tour} \\ 0 & \text{Otherwise} \end{cases} \quad (5)$$

$0 < \alpha < 1$ is the pheromone decay parameter, and L_{gb} is the length of the globally best tour.

2.2. ELU-Ants

ELU-Ants [4] is a modified of the original ACS. Hosein in his research offered two equation to implement the algorithm, one is named Kcc-Ants and the another one is ELU-Ants. Al though both of them used the similar approach, they represented their proposed on different equation. The results of both equation are almost similar. This research uses the ELU-Ants approach. The main idea of this research is the ants have more ability to add more the pheromone of the primary links than the last links. While in early steps, the ant has a lot of options to choose which city traveled, because it has passed just a few cities. As a consequence, it makes just a few edge are prohibited to be selected. Therefore, the ant can freely choose the most desirable link (with more pheromone and less length) as its next link. Contrary on final steps of tour, the ant already passed most of the cities, and the current selected link may not have a significant effect on the quality of the tour, so it seems logical to reduce its ability of changing the last link pheromone. In other words, ants have more effect on pheromone update where they are in their initial steps and less effect when they are going to finish the tour. Based on above discussion, local update original (Eq.(3)) is modified to be ELU-Ants (Eq. (6)) and Kcc-Ants (Eq. (7)).

$$\tau(i, j) = \tau(i, j) + \tau_0 \cdot e^{\frac{-5 \cdot cc}{n}}, \quad (6)$$

$$\tau(i, j) = \tau(i, j) + \frac{K \cdot cc \cdot \tau_0}{Cl \cdot \omega}, \quad (7)$$

where cc represents the number of cities passed till now; n is total number of cities ; K and ω denote two parameters given; Cl represents current length of passed path of each ants and represents initial pheromone.

2.3. Bee Colony Optimization

Bee colony optimization (BCO) is designed to create the multi agent system to solve the combinatorial optimization problems. Lucic et al [16-17] proposed BCO to solve TSP. On implementation BCO on TSP, the tour consists numerous stages. A stage is a collection of a certain amount of nectars. In TSP model, nectars can represent cities. In constructing the tour, a bee starts as an unemployed forager without knowledge about the food (nectar). Bees start exploring from nectar to nectars or travel. This step is called forward pass (FP). As a notion, not all bees start foraging in the first stage. Some of them start at the second stage, third stage, etc.

Once a bee is on, she will be active till the end of iteration. After performing one stage, a bee returns to the hive in order to unload and store the nectar. It is called backward pass (BP). In BP, bees will perform waggle dance to show their quality tour and start exchanging information. The kind of communication between individual bees contributes to the formation of the bee colony. Before communicating, BCO selects randomly with a particular probability (e.g. 10 %) to be scout. The bees which retain the previous stage and continue the next stage without interacting with the others are named scout. The remind bees will be categorized into follower and recruiter. BCO uses Eq. (8) for determining the probability that the bee k in next stage ($u+1$) with the same partial tour at stage u in iteration z as described below :

$$P_k(u+1, z) = e^{-\frac{L_k(u,z) - \min_{r \in w(u,z)}(L_r(u,z))}{uz}} \quad (8)$$

where $L_k(u,z)$ is the length stage-tour that is discovered by bee k in stage u in iteration z . As a result of interaction, in addition of scout, the bees have the two other types, recruiter and follower. The bees which retain their previous stage and recruit the follower bees on their way are named recruiter. The bees which abandon the food source (previous stage) and follow the recruiter bees on their way are named follower. Since the probability of best path bee will be 1 ($P_k = 1$), she will use absolutely the same partial and categorized as group of recruiter. BCO introduced the probability of partial tour will be chosen by any bee that decided to choose the new route in one stage. The probability is generated regarding two main attribute the total length tour and number bee that are advertising the partial tour. The smaller the normalized value of the total length, the better is the partial tour, while the bigger normalized value of number bees, the better is the partial tour. Procedure BCO can be described as on Fig. 2.

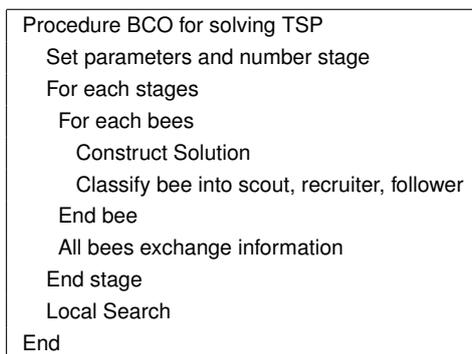


Figure 2. BCO Algorithm.

3. Proposed Algorithm

3.1. The Concept

Since there is a consideration for beginning step and last step in constructing tour, the tour should be split into some parts of tour. This part of tour can be identified as the stage in BCO which consists some cities. The stage-tour agent constructed is expected to judge the quality agent. On BCO, after traveling one stage, bees back to hive and group into three types follower, recruiter, and scout based on their quality. This algorithm uses "ant-bee" as its agent and mimics the behavior of bee on BCO. Follower ant-bee is identified as the agent which performs bad choice on its stage-tour. Recruit ant-bee is identified as the agent which perform good choice on its stage-tour. Scout ant-bee is identified as the agent which always find the new alternative tour. Further, follower will change their stage-tour and follow the stage-tour of recruiter agents. Scouts neither follow nor recruit the other agent, yet keep their stage-tour.

Fig. 3 gives the illustration how generates the competitive agents. Suppose there are three agents M_1 , M_2 and M_3 and three cities at first stage. S_{11} , S_{12} , S_{13} are the collection of

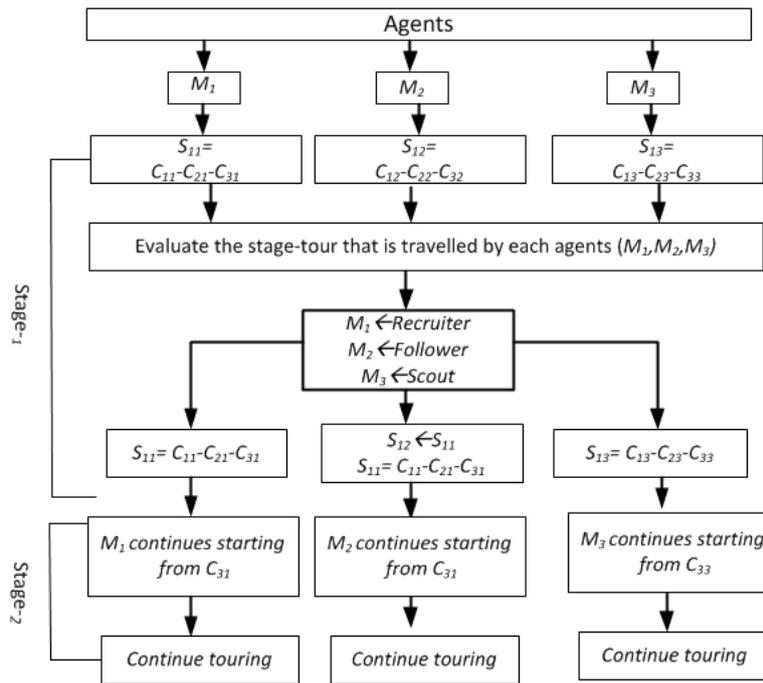


Figure 3. Illustration how generates the competitive agent.

cities traveled at the first stage by agent M_1 , M_2 and M_3 respectively. $C_{11}-C_{21}-C_{31}$, $C_{12}-C_{22}-C_{32}$, $C_{13}-C_{23}-C_{33}$ are the sequence cities traveled at first stage by M_1 , M_2 and M_3 respectively. After traveling the first stage, all agents are evaluated to conclude their type. Suppose M_1 , M_2 and M_3 become recruiter, follower and scout respectively. At the end of first stage, M_1 as recruiter keeps its stage-tour, M_2 as follower changes its stage-tour and follow M_1 's stage-tour, M_3 as scout keeps its stage-tour. Therefore, at beginning second stage the agents M_1 , M_2 and M_3 should continue the tour from city C_{31} , C_{31} , C_{33} respectively. The interesting question is how to make a good judgment in order to classify the quality of agents on constructing stage-tour. The sub section 3.3 will explain that the total amount pheromone gathered on each stage can be used as a good judgment.

3.2. Construct Tour

In construction tour for solving TSP, the ant-bee chooses cities with ACS algorithm (Eq. (1) and Eq. (2)). After each ant-bees traveling one stage using the rule of ACS, it backs to hive to interacting with each other (supposing distance from hive to each city=0), and calculating the total amount of pheromone. HABCO also categorizes ant-bees into three types based on pheromone gathered namely scout(S), follower(F), and recruiter(R) as BCO. Scout retains the previous stage and continues the next stage without interacting with the others, follower abandons the previous stage and follows the recruiter, recruiter retains her previous stage and recruits the follower to join her previous stage tour. On HABCO, the judgment of quality of agents is based on the amount pheromone that agent gathers in each stages. Judgment is used to classify ant-bees into Follower, and Recruiter. Scout agent is firstly chosen 10 % from all agents. The scout agent guarantees the alternative tour is kept on the process. The remainder agents is categorized as follower and recruiter. The probability ant-bee into follower after traveling one stage is used by Eq. (9).

$$P_F = \frac{\tau_{max} - \tau_i}{\sum_{i=1}^M (\tau_{max} - \tau_i)}, \quad (9)$$

where i is agent index; M is the total number agent ant-bees; τ_i represents pheromone gathered at each stages by agent; τ_{max} represents pheromone maximum gathered at each stages by agent. From Eq. (9) shows that agent which gathers the high pheromone has low probability to be follower. For the highest pheromone (τ_{max}) agent is impossible to be follower ($P_F=0$). In other words, it will absolutely become a recruiter. After categorized the agents into follower and recruiter, the follower will change her tour and follow one of the follower agent randomly. After exchanging information on hive, the agent ant-bees continue the next stages until finish the tour.

3.3. Pheromone updating

There are 3 types pheromone updating on HABCO. They are local update, semi-global update and global update. Local update is performed as local updating rule in ACS algorithm (Eq. (3)). While ant-bees choosing their edge, they also modify their pheromone. After performing one stage, HABCO updates the pheromone called semi-global update. This update is based on modified ELU-Ants algorithm that ants have more effect on pheromone update where they are in their earlier steps and less effect when they are going to finish the tour.

$$\tau(i, j) = \tau(i, j) + \tau_0 \cdot e^{-\frac{5S}{|S|}}, \quad (10)$$

where S is current stage; $|S|$ is total number stage. Obviously from Eq. (10), the pheromone after performing the first stage ($S=1$) is added more than the next stage. So the ant-bees play fewer impact in update pheromone when they are in their final part of tour. In last stage (final part), $S = total\ number\ stage$, it makes the impact pheromone update toward zero. At last, the pheromone on the edges are travelled by the best ant-bee will be updated with global updating rule as the ACS Eq. (4) and Eq. (5).

3.4. Ruin local optima

In the observation, the best length tour is possible same in some iterations in a row especially on the relative small benchmark. The best ant-bee probably travels on same edge in each iterations. So, it might the process traps to be local optimal. This situation makes the algorithm sometimes fails to get the optimal solution. To avoid this problem (ruin local optimal), the pheromone on that edges is reduced/evaporated, when the best of length tour is same for t times iterations ($t=100$) in a row.

$$\tau_{rs} = x \cdot \tau_{rs}, \quad (11)$$

where $0 < x < 1$. Variable is considered as a fair number. When x is set too high, ruining of local optimum will fail to be obtained. Contrarily, if x is set too small, it will decrease pheromone significantly and affect the construction of tour. The algorithm of HABCO is described as Fig. 4.

4. Performance and Analysis Results

4.1. Analysis Constructing the Robust Agent

As aforementioned, the pheromone and distance nodes are two variables important in ACO. This section probes whether the two variables can be to a judgment of the quality agent. One benchmark TSP, *st70*, is investigated to see impact of the distance and amount pheromone at each stages that is performed by ants. This actions are tried on three stages while number agents are ten. Firstly the agent (ant) travels all cities stage by stage with follow the ACS rule on Eq. (2). We observe two kind the agents. The first is the best agent which get the best complete tour. The second one is the worst agent which get the worst complete tour. While traveling, the total distance and the amount of pheromone at each edges is added and calculated in one stage. After traveling one stage, each agent continues next stage without interaction each other. At last, after each agents completes their tour, the length complete tour and pheromone of each ants also be calculated and ranked.

In the observation, considering by calculating the distance traveled at one stage is not reliable. It shows the best ant which gets the shortest distance tour has various position rank on first stage. Further, the worst ant which travels the longest distance has not always bad position

```

Set parameters and number stage
For each stages
  For each ant-bees
    construct solution; local update
  End ant-bees
  Semi-global update; Ranked ant-bees
  For each ant-bees //Back to hive
    Classify ant-bees into scout, recruiter, follower
  End ant-bees
  All ant-bees exchange information
  If (stage=number stage)
    Calculating the length complete tour of each ant-bees
    Global update by the best ant-bee
    Ruin local optima
  End if
End Stage
Local Search

```

Figure 4. HABCO Algorithm.

on first stage. It means that an ant can not guarantee getting the relatively short in the complete tour when traveling relatively short at the first stage. Contrarily, when an ant travels relatively long at first stage, she can not guarantee getting relatively long in the complete tour. It is because each ants is possible to choose several cities which have short distance in the beginning, but she has no choice except continuing the long distance at last. Thus, the total tour might be still relative long at last.

The second observation is to rely on the total amount of pheromone in one stage-tour traveled by each ants. In this experiment, it shows that the ant which gathered a lot of pheromone at first stage tends to be a good or best ant in the complete tour. The other hand, the ant which get the shortest distance probably gets the high pheromone on its first stage. So, it is more reasonable that total amount of pheromone at each stage indicates the quality sub-tour. The greater total pheromone that the ant achieves in one stage, the ant has high probability of getting the good complete tour. So if the amount at early stage-tour is higher, the ant tends to get the best complete tour. Contrary, if the pheromone at early stage-tour is lower, the ant tends to the worst complete tour.

As a notion, this research is performed on three stages. In our investigation, the result of four or more stages is not be reliable. It means the pheromone gathered by agents at first stage can not be a good judgment of quality agents.

Experiments are conducted to evaluate the performance of HABCO. Table 1 shows the settings of various parameters for the proposed algorithm. This paper details two comparison. The first one is a comparison HABCO and BCO, and the second one is a comparison HBCO and ACS. All algorithms are executed 10 times and nine dataset benchmark TSP.

Table 1. Parameter Setting of HABCO algorithm

Parameter	Symbol	Set value
number ant-bee	M	10
number stage	S	3
iteration	nc	1000
degree pheromone	B	2
parameter local update	γ	0.1
pheromone decay	α	0.1
limit exploitation	q_0	10 %
factor ruin local optima	X	0.95

Table 2 display the results of all algorithms and optimal solution (BKS/best known solu-

tion) as well. The first comparison is HABCO and BCO. Table 4 shows HABCO+2opt outperforms BCO+2opt. On some datasets (eil51, kroa150, a280), HABCO without 2opt outperforms BCO+2opt. At first HABCO uses ACS method (Eq. (1) and Eq. (2)) to construct the tour. ACS has an advantage to make the edge more dynamic by perform local update. With local update, pheromone on edge visited is diminished and make it less desirable. Therefore the premature converge can be avoided. As a consequence, the agent (ant) has many various options to develop its tour. With only ten agents, ACS can get the good quality tour. In other hand, BCO has main advantage on exchange information agents (bees) on their hive. Before backing to hive, construction bee depends only the role on Eq. (7). It makes the selection candidate node is more static. BCO has also a slightly complex formula than ACS, and also has many variables to set. The difference setting sometimes generates the different significant results. The second comparison should be discussed is comparing HABCO and ACS. Table 4 shows that HABCO outperforms ACS either with 2opt or without 2opt. HABCO also shows slightly better either in average or best case. Although ACS and HABCO uses the same role to construct the tour (Eq. (1), Eq. (2)) at each stages, HABCO has an advantage in producing the competitive tour of their agents by duplicating the good sub tour and discarding the bad sub tour in their hive. Therefore, in constructing the tour, HABCO has smarter agents than ACS.

Table 2. Performance HABCO, ACS and BCO on nine benchmark datasets

Bench mark	BKS	ACS		ACS+2opt		BCO+2opt		HABCO		HABCO+2opt	
		Avg	Best	Avg	Best	Avg	Best	Avg	Best	Avg	Best
eil51	426	432.4	430	430.3	428	432.1	429	430.7	428	429.1	427
berlin52	7542	7715.8	7542	7611.4	7542	7652.6	7542	7577.9	7542	7543.5	7542
st70	675	693.4	684	686.2	680	683.2	675	686.5	680	681.1	678
kroa100	21282	22054.8	21716	21568	21369	21927.3	21369	21789.9	21402	21413.5	21282
eil101	629	649.9	641	643.9	636	656.2	645	643	634	641.7	634
kroa150	26524	27476.9	26889	27219.1	26912	27452.4	27063	27139.1	26881	27015.9	26781
d198	15780	16480.2	16202	16207	15918	16351.5	16021	16421.5	16182	16030	15895
a280	2579	2739.6	2686	2669.9	2640	2834.1	2701	2702.4	2653	2665.1	2630
u724	41910	46363	44688	44136.4	43781	46516.7	44091	45117.8	43901	44002.3	43432

5. Conclusion

HABCO as combination three algorithms, ACS, BCO and ELU-Ants, is an affective algorithm to solve the TSP problem. In solving TSP, HABCO divides the tour into some stages likes on BCO algorithm. In touring some cities in a stage, HABCO uses the ACS method from choosing the nodes, and updating local and global updating pheromone. The quality of each agents HABCO is judged based on the pheromone gathered at each stages. The pheromone gathered in ACS algorithm classifies agent into follower, recruiter and scout agent like on BCO algorithm. A good judgment on premature step makes the probably good agent (identified as recruiter) duplicated, and the probability bad agent (identified as follower) is discarded. The scout agent is exist to maintain the new alternative tour. This process generates the competitive agent in each iteration. This algorithm also proposed the semi-global update pheromone to adopt the ELU-Ants algorithm. Comparing ACS and BCO, HABCO can solve TSP problem better. It can be seen from solving some dataset TSP problem with local or without local search.

6. Acknowledgement

This article is extended version of proceeding [25] with entitled "A Hybrid Ant-Bee Colony Optimization for Solving Traveling Salesman Problem with Competitive Agents.

References

- [1] Dorigo M, Gambardella L M. Ant Colony System: A Cooperative Learning Approach to The Traveling Salesman Problem. *IEEE Transactions on Evolutionary Computation*. 1997; 1(1): 5366.

- [2] Girsang A S, Tsai C W and Yang C S. Rectifying the Inconsistent Fuzzy Preference Matrix in AHP Using a Multi-Objective Bicriterion Ant. *Neural Processing Letters*. 2016; 44(2) : 519-538.
- [3] Potvin J V, Genetic Algorithms for the Traveling Salesman Problem. *Annals of Operations Research*. 1996; 63 : 339-370.
- [4] Clerc M. Discrete Particle Swarm Optimization Illustrated by The Traveling Salesman Problem. *New optimization Techniques in Engineering*. 2004; 219-239.
- [5] Wang K P, Huang L, Zhou C G, Pang W. *Particle Swarm Optimization for Traveling Salesman Problem*. International Conference on Machine Learning and Cybernetics. 2003; 1583-1585.
- [6] Chu S C, Tsai P W, Pan, J S. *Cat Swarm Optimization*. PRICAI 2006: Trends in Artificial Intelligence. 2006; 854-858.
- [7] Lucic P. Modeling Transportation Problems Using Concepts of Swarm Intelligence and Soft Computing. PhD Thesis. Virginia : Faculty of the Virginia Polytechnic Institute and State University; 2002.
- [8] Teodorovic P, Lucic P, Markovic G, DellOrco M. *Bee Colony Optimization: Principles and Applications*. 8th Seminar on Neural Network Applications in Electrical Engineering, NEUREL. 2006.
- [9] Wong L P, Low M Y H, Chong C S. *A Bee Colony Optimization Algorithm for Travelling Salesman Problem*, Proceedings of Second Asia International Conference on Modelling and Simulation (AMS). 2008; 818-823.
- [10] Lucic P, Teodorovic D. Vehicle Routing Problem with Uncertain Demand at Nodes: The Bee System and Fuzzy Logic Approach. *Fuzzy Sets in Optimization*. 2003; 67-82.
- [11] Girsang A S, Tsai C W, Yang, C S. *A Fast Bee Colony Optimization for Traveling Salesman Problem*. Third International Conference on Innovations in Bio-Inspired Computing and Applications (IBICA).Kaohsiung. 2012; 7-12.
- [12] Yu W J, and Zhang J. *Pheromone-distribution-based adaptive Ant Colony System*. Proceedings of the 12th annual conference on Genetic and evolutionary computation. Portland. 2010.
- [13] Naimi H M, Taherinejad N. New Robust and Efficient Ant Colony Algorithms: Using New Interpretation of Local Updating Process. *Expert Systems with Applications*. 2009; 36 (1): 481-488.
- [14] Chen S M, Chien C Y. Parallelized Genetic Ant Colony Systems for Solving The Traveling Salesman Problem. *Expert Systems with Applications*. 2011; 38 (4): 3873-3883.
- [15] Chen S M, Chien C Y. Solving The Traveling Salesman Problem Based on The Genetic Simulated Annealing Ant Colony System with Particle Swarm Optimization Techniques. *Expert Systems with Applications*. 2011; 38 (12): 14439-14450.
- [16] Lucic P. *Modeling Transportation Problems Using Concepts of Swarm Intelligence and Soft Computing*. Virginia. 2002.
- [17] Teodorovic D, Lucic P, Markovic P, Orco M D. *Bee Colony Optimization: Principles and Applications*. 8th Seminar on Neural Network Applications in Electrical Engineering, NEUREL, 2006.
- [18] Lucic P, Teodorovic D. Vehicle Routing Problem with Uncertain Demand at Nodes: The Bee System and Fuzzy Logic Approach. *Studies in Fuzziness and Soft Computing*. 2003; 126 : 67-82.
- [19] Karaboga D, Basturk B. A Powerful and Efficient Algorithm for Numerical Function Optimization: Artificial Bee Colony (ABC) Algorithm. *Journal of Global Optimization*. 2007; 39(3): 459-471.
- [20] Benatchba K, Admane L, Koudil M. Using Bees to Solve a Data-Mining Problem Expressed as a Max-Sat One. *Artificial Intelligence and Knowledge Engineering Applications: A Bio-inspired Approach*. 2005 ; 75-86.
- [21] Chong C S, Low M Y H, Sivakumar A I, Gay K L. *A Bee Colony Optimization Algorithm to Job Shop Scheduling*. Proceedings of Winter Simulation Conference. 2006; 1954-1961.
- [22] Fathian M, Amiri B, Maroosi A. Application of Honey-Bee Mating Optimization Algorithm on Clustering. *Applied Mathematics and Computation*. 2007; 190 (2), 1502-1513.
- [23] Wang J, Zhang D. Image Denoising Based on Artificial Bee Colony and BP Neural Network.

- TELKOMNIKA (Telecommunication Computing Electronics and Control)*. 2015; 13(2) : 614-623.
- [24] Meng L, Yin S, Hu X. A New Method Used for Traveling Salesman Problem Based on Discrete Artificial Bee Colony Algorithm. *TELKOMNIKA (Telecommunication Computing Electronics and Control)*. 2016; 14(1):342-348.
- [25] Girsang A S, Tsai C W, Yang C S . *A Hybrid Ant-Bee Colony Optimization for Solving Traveling Salesman Problem with Competitive Agents* Proceeding in Mobile, Ubiquitous, and Intelligent Computing. 2014; 643-648.