A New Method Used for Traveling salesman problem Based on Discrete Artificial Bee Colony Algorithm

Lei Meng[1], Shoulin Yin*[2], Xinyuan Hu[3]
Software College, Shenyang Normal University,
No.253, HuangHe Bei Street, HuangGu District, Shenyang, P.C 110034 - China
*Corresponding author, email: 8871346@qq.com[1], 352720214@qq.com[2], 1138074916@qq.com[3]

Abstract

We propose a new method based on discrete Artificial Bee Colony algorithm (DABC) for traveling salesman problem. We redefine the searching strategy and transforming mechanism of leading bees, following bees and scout bees according to discrete variables. The transition of swarm role is based on ratios factor of definition. Leading bees use 2-Opt operator and learning operator to accelerate the convergence speed and to search the neighborhood. The searching of following bees introduces tabu table to improve the local refinement ability of the algorithm. Scouts bees define an exclusive operation to maintain the diversity of population, so it is better to balance the exploration and exploitation ability of the algorithm. Finally, the experimental results show that the new algorithm can find relatively satisfactory solution in a short time, and improve the efficiency of solving the traveling salesman problem.

Keywords: *DABC algorithm, Tabu table, Traveling salesman problem, Exclusive operation, 2-Opt operator*

1. Introduction

Traveling salesman problem (it is abbreviated in TSP) [1] is an important combinational optimization problem in the area of mathematics. It belongs to Non-Deterministic Polynomial (NP) problem [2]. Although there are some precise algorithms which can be used to solve the problem, the principle of precise algorithms is complex, and it can produce “combination explosion problem” along with the increase number of city, therefore, the domestic and foreign scholars have been trying to seek a highly efficient and stable algorithm for solving this complex problem.

Artificial bees colony algorithm (ABC) [3] is one of heuristic search algorithms based on swarm intelligence, which was proposed by Karaboga et al. [4] in 2005. ABC algorithm [5] is based on the self-organization of the swarm simulation model with the advantages of less setting parameters, strong robustness, it has received extensive attention of scholars both at home and abroad, and been applied in many fields.

Nowadays, many researchers have studied the new algorithm about Artificial bees colony for TSP. Sharma P et al. [6] showed that bees speculative modified over time and based on the best solution found by the bee itself and the swarms of bees were dynamically divided into smaller groups and search process was performed by an independent smaller group of bees. Hong-Tao et al. [7] proposed a discrete artificial bee colony algorithm. And he introduced a tabu list and a repulsion operator.

The above methods almost are used for solving continuous domain optimization problems. However, ABC algorithm is relatively few used in the aspect of discrete domain application. To improve the performance of TSP solution, we propose improved discrete Artificial Bee Colony algorithm through redefining leading bees, following bees and scouts which better coordinates and balances the exploration and mining process of ABC algorithm. To facilitate the description, this paper also gives some definitions. We present a new discrete Artificial Bee Colony algorithm for traveling salesman problem. This new scheme takes balance of space exploration and the local refinement into consideration. Finally, we conduct some experiments to verify its performance. The results show that the new algorithm has a good effect on solving TSP question.

Received October 30, 2015; Revised January 5, 2016; Accepted January 24, 2016

2. Artificial Bees Colony Algorithm and Discretization

Optimization process of artificial colony algorithm simulates the behavior of bees searching a high quality nectar source. It divides artificial bee colony into leading bees, following bees and scouts according to their division of labor. They change roles based on certain conditions. Each bee is a possible solution of corresponding problem, profitability ratios of nectar source represents the quality of the solution. Each leading bee is corresponding to a certain nectar source, and in the process of iteration it starts to search among its neighborhood. After searching, leading bees would share the searched information with the following bee. Following bees choose nectar source according to certain probability, and they keep on searching in their neighborhood. If leading bees within a given number of searching don't accept better nectar source, they will give up this nectar source and leading bees would be transformed into scouts to randomly search feasible new nectar source.

ABC algorithm is an intelligent optimization algorithm, and exploration and production is two main factors, which decides to the performance of swarm intelligence optimization; the better exploring ability and the stronger searching ability for individual in globally searching unknown area. What this amounts is to, global optimization capability is stronger, the exploiting ability is better. The ability of individual searching optimal solution in local area is stronger, ability of local refinement in better. Therefor, to ensure the solution quality of ABC algorithm, it needs to coordinate and balance the exploration and mining process, which is a core problem in intelligent algorithm. When solving the problem of continuous variables, ABC algorithm lacks consideration of the exploration and exploitation ability. That is difficult to guarantee solving speed and quality of algorithm. Here are four definitions for discretization ABC.

Profitability ratio r_i : the ratio between searched nectar source nectar amounts of each bee fit_i and nectar source nectar amounts of optimum individual in swam fit_{best} . For TSP, the relation of fit_i and objective function $f(X_i)$ is:

$$fit_i = \frac{1}{f(X_i)} \tag{1}$$

Similarity $s_{i,j}$: level of similarity of any two solution vectors X_i and X_j , $s_{i,j} = M / N$. Where *N* is the number of cities and *M* is the same adjacent side number of two solution vectors. $s_{i,j} \in [0,1]$. For example, N=10, X_1=[1,5,6,2,4,8,3,10,7,9] and X_2=[1,3,6,2,4,8,10,5,7,9]. The two solution vectors have five same adjacent sides: [6 2], [2 4], [4 8], [7 9], [9 1], So $s_{i,j}$ =0.5.

Exclusive operation: randomly arrange same adjacent side of two solution vectors and get a new vector Y. And it acts Y on X_j , which decreases $s_{i,j}$.

Learning operation: randomly arrange same adjacent side of two solution vectors and get a new vector Y. And it acts Y on X_j , which increases $s_{i,j}$.

3. Discrete Artificial Bee Colony algorithm for TSP

According to the above definitions, we divide artificial bee colony into leading bees, following bees and scouts according to their division of labor. They change roles based on certain conditions. Each bee is a possible solution of corresponding problem, profitability ratios of nectar source represents the quality of the solution. We redefine the three bees.

3.1. Improved Leading Bees

The leading bees in the discrete ABC algorithm have the same function as ABC algorithm. It searches an optimal nectar source in the neighborhood of each nectar source. But it is a discrete variable problem for solving TSP, generation mechanism of neighborhood solution needs to be redefined. So the detailed processes of improved leading bees are as follows. Firstly, this paper adopts 2-opt neighborhood structure to search neighborhood of each nectar source at the beginning of optimization. When operating 2-opt neighborhood, secondly it removes two non-adjacent edges, and then re-operates corresponding four cities on the two non-adjacent edges: first city of first side is linked together with the first city of second side, second city of first side is linked together with the second city of second side. Then it obtains the new solution as the the neighborhood solution.

Example 1. There is a 5-note TSP, solution is $X_i = [1,2,3,4,5]$ as Figure1(a), and we operate side (1,2) and (4,5) based on 2-opt and get neighborhood solution $X_i' = [1,4,3,2,5]$ as Figure 1(b).

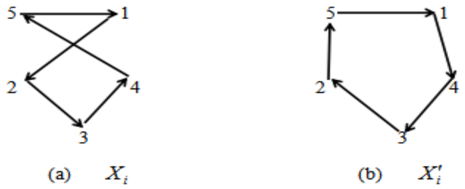


Figure 1. 2-opt neighborhood operation

Comparing X_i with X_i', the real operation of 2-opt is that it inverts the second city of first side as the the first city of second side.

3.2. Improved Following Bees

The following bees in the discrete ABC algorithm have the same function as ABC algorithm. It calculates the probability based on the roulette method and formula (2) to choose one nectar source and it starts to conduct local searching.

$$p_i = \frac{fit_i}{\sum_{i=1}^{SN} fit_i} \tag{2}$$

SN is the total number of nectar source. The improved following bees in the discrete ABC algorithm introduce Tabu table $tabu_i (i = 1,2 \ldots\ldots SN)$ into record neighborhood searching information of bees nearest nectar source. For example, the nearest neighborhood solution X_i' of nectar source X_i in example 2 is obtained by converting city 2 and city 4. If we again make 2-opt transformation for side (2,5) and side (1,4) in this two cities respectively, and it will get neighborhood solution X_i which is meaningless. In order to avoid the bees repeated searching within the same neighborhood and improve the production ability of algorithm, for Tabu table $tabu_i$ in example 2, it just needs to record (2,4) two points, namely record the second city of first side and first cit of second side.

3.3. Improved Scouts

The scouts are transformed by leading bees and have the responsible for finding the individual with possibility falling into a local optimum and updating it. That can reduce the probability of precocity. Leading bees are transformed into scouts, which satisfies the following conditions: it does not obtain better nectar source within the setting searching number. As we all know, there are three questions among scouts shift mechanism and searching behavior in ABC algorithm.

a. The changing role of scouts depends on the given searching number. This parameter is different for solving different practical problems and difficult to set. Also setting unsuitable parameters will affect the ability of algorithm to jump out of local optimal.

b. Scouts are transformed only after individual changing into precocity and it explores new nectar sources, which leads to slow speed to find the global optimal solution.

c. Scouts randomly search new nectar sources in ABC algorithm without considering position of current dominant searching leading bees. It cannot guarantee that scouts search more new nectar sources in global scope. Finally, this restrains exploring ability of the algorithm.

In order to solve the above shortcomings, this paper defines the profitability ratios, leading bees are transformed into scouts based on the index. The advantage of this definition, on the one hand, is that the index is more likely to be set when solving different problems. On the other hand, swarm role and search mechanism relies on the dynamic changing of searching number. At the optimizing initial stage, the profitability ratios is relatively poor, leading bees immediately turn into scouts. And it operates exclusive operation based on roulette method, so that it can find better nectar sources in global scope. In the late optimization, the profitability ratios are all good, leading bees no longer turn into scouts. The search mechanism is transformed from 2-opt to learning operation based on roulette method. Namely, we should adequately consider the individual species diversity at the initial stage to improve exploring ability of ABC algorithm. At the end of optimization, population information sharing should be taken into account to improve the production ability and optimization speed of ABC algorithm.

The detailed process of the proposed ABC algorithm is as follows:

a. Step 1. Initializing the swarm. Randomly initializing nectar sources, namely randomly generating population initial solution, and calculating profitability ratios of each nectar source by formula (1).

b. Step 2. The searching stage of leading bees. Calculating profitability ratios. If profitability ratios are greater than or equal to the setting value *r*, then leading bees execute 2-opt operation, search new nectar sources and get candidate neighborhood solution X_i'. Otherwise, leading bees adopt roulette method based on probability calculated by formula (2) to choose nectar source X_i, execute learning operation and search new nectar sources and get candidate neighborhood solution X_i'. If candidate profitability ratios $fit_{X_i'}$ is greater than $fit_{X_i'}$. X_i' will replace X_i.

c. Step 3. The searching stage of following bees. Following bees adopt roulette method based on probability calculated by formula (2) to choose nectar source X_i. And it executes 2-opt operation within the permissible operating scope, search a new nectar sources and get a candidate neighborhood solution X_i'. If candidate profitability ratios $fit_{X_i'}$ is greater than $fit_{X_i'}$. X_i' will replace X_i.

d. Step 4. The searching stage of scouts. Calculating profitability ratios. If profitability ratios are greater than or equal to the setting value *r. r* is a setting value. Then it gives up the nectar source X_i whose profitability ratios is less than *r*. It adopts roulette method based on probability calculated by formula(2) to choose nectar source X_i, executes learning operation and searches new nectar sources X_i'. Otherwise return to step5.

e. Step 5. Recording the searched best solution.

f. Step 6. Whether the process satisfies termination conditions or not. If YES, then finish. If NO, return back step2.

4. Experiments and Results

4.1. The Effect of Profitability Ratios Setting Value on DABC

The setting value *r* of profitability ratios is the main parameter in Discrete ABC algorithm(DABC). We make simulation experiments to analyze the effect of *r* on performances of algorithm solution under the MATLAB platform. We adopt different *r* to conduct experiments under the other same parameters conditions. Setting maximum iterations $T_{\max} = 2000$, number of bees is $m = 48$, algorithm runs 20 times independently. Table1 records the results with different *r* (*r* is a setting value).

From Table 1 we can know that when *r* gradually reduces about solution time, the average running time is little-changed. When *r=0.7, 0.8, 0.9*, average running time is relatively short. And *r=0.8*, average running time is 9.0341 which is the shortest time. But as a whole, the average running time of algorithm has a little change. This is mainly because whatever value *r* is, the sum of leading bees, following bees and scouts remains unchanged in the iteration process of algorithm. In current iteration, leading bees have been transformed into scouts, they no longer perform operations of leading bees in the next iteration. Also search strategy and complexity of swam has little difference at different stage based on profitability ratios. When *r* gradually reduces about solution quality, the solution performance gets good and then gradually becomes worse. When *r=1, 0.6, 0.5, 0.4, 0.3, 0.2, 0.1, 0*, average value of DABC is greater than 3.5×10^{-4}. The worst values are all greater than 3.6×10^{-4}. However, when *r=0.7, 0.8, 0.9*, average value is less than 3.5×10^{-4}. The worst values are all less than 3.6×10^{-4}. When *r=0.8*, average value is 3.43×10^{-4}, worst value is 3.49×10^{-4}. Best value is 3.34×10^{-4}. The standard deviation is 3.58×10^{-2} just over the standard deviation (3.50×10^{-2}) when *r=0.7*. Therefore, *r=0.8* is the best value for DABC.

Table 1. Different *r* with different influences on DABC

r	Best value×0.0001	Worst value ×0.0001	Average value ×0.0001	Standard deviation ×0.01	Average running time/s
1	3.39	3.59	3.50	5.88	9.9222
0.9	3.35	3.57	3.47	4.94	9.0368
0.8	3.38	3.49	3.43	3.58	9.0341
0.7	3.42	3.52	3.44	3.50	9.0462
0.6	3.47	3.60	3.54	4.80	9.6879
0.5	3.44	3.66	3.55	4.85	9.6235
0.4	3.47	3.65	3.56	5.43	9.6235
0.3	3.48	3.65	3.57	4.47	9.5878
0.2	3.45	3.62	3.56	4.33	9.6211
0.1	3.44	3.64	3.56	5.84	9.6019
0	3.43	3.66	3.56	6.28	9.6218

4.2. The Experiment of Algorithms Performance Comparison

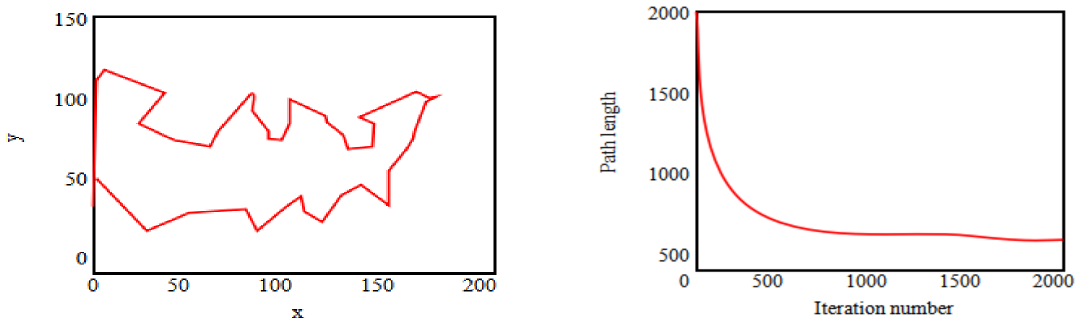
In order to further verify the feasibility and effectiveness of this paper's method, we choose multiple instances in international universal test library for testing, and use testing results to make a comparison to Resource Recovery Solutions Artificial Bees Colony (RRSABC) [8], Ant Colony Optimization [9] (ACO) and ABC.

In the experiment, digit indicates number of cities. Number in parentheses represents known optimal solution. "O" shows calculation result of the index is not given in the comparation literature. Experimental environment is the same as above. Parameters setting of DABC algorithm: maximum number of iterations $T_{\max} = 2000$, *r=0.8*, number of bees $m = n$, *n* is number of city. Parameters setting of ACO algorithm: $m = n$, $\alpha = 1$, $\beta = 5$, $\rho = 0.9$. The two algorithms run 20 times dependently. Calculation results are as shown in Table 2.

From Table 2, RRSABC algorithm gets the known optimal solutions in Berlin52, Oliver30 and Eil51. But iteration number of DABC has reduced by 98percent. What's more, steps of DABC are not complicated. The average value has increased by 0.5% (Not exceed 0.5%). Compared with ACO algorithm, the best value and average value is relatively small, and for Bays29, Oliver30 and Dantzig42, the standard deviation is smaller. For the five instances, the running time of DABC has saved by 98.2%, 98.3%, 98.66%, 98.96 and 98.98% respectively. ABC algorithm also can obtain the results closely to known optimal solutions for all instances. For Oliver30 and Bays29, DABC algorithm gets the same results as known optimal solutions. For Dantzig42, calculation result of DABC algorithm has decreased by 19.8% compared with ABC algorithm. Figure 2 shows the experimental simulation diagram of Dantzig42. The corresponding optimal path is [33,34,35,36,37,38,39,40,41,42,1,2,3,4,5,6,7,8,9,10,25,26,27,11,12,23,22,17,16,13,14,15,18,19, 20,21,28,29,30,31,32].

Table 2. Calculation Results

Instance	Calculation index	DABC	RRSABC	ABC	ACO
Bays29(2020)	Best value	2020	O	2020	2040
	Average value	2027.8	O	2031.4	2061.7
	Iteration number	2000	O	50	2000
	Running time	5.11	O	O	285.97
Oliver30(423.74)	Best value	423.74	423.74	423.5	425.82
	Average value	425.00	423.74	432.5	428.76
	Iteration number	2000	100000	50	2000
	Running time	5.36	O	O	317.73
Dantzig42(699)	Best value	679.20	O	699	696.12
	Average value	692.05	O	710.2	716.46
	Iteration number	2000	O	100	2000
	Running time	8.95	O	O	672.48
Eil51(429.98)	Best value	431.24	428.87	426.00	457.79
	Average value	444.09	431.11	427.60	468.51
	Iteration number	2000	100000	50.0	2000
	Running time	10.36	O	O	1001.04
Berlin52(7544.4)	Best value	7680.78	7554.37	7542	8048.07
	Average value	7941.27	7562.80	7573.48	131.51
	Iteration number	2000	100000	150	2000
	Running time	10.76	O	O	1055.01



(a) Optimal path (b) Best value evolution curve

Figure 2. Experimental simulation diagram of Dantzig42

The average value of DABC algorithm is relatively small compared with ABC algorithm for Oliver30, Dantzig42 and Bays29. ABC algorithm is mainly based on the ACO algorithm and has combined with other operations. When the number of population in ABC is equal to that in ACO, the running time in ABC is closely to that of ACO at least. Nevertheless, the running time in ACO is relatively long. Therefor, DABC algorithm has the better solution performance in terms of solution time and solution quality compared with ACO, ABC and RRSABC algorithm.

5. Conclusion

This paper proposes a new discrete artificial colony algorithm for solving TSP question. It extends continuous artificial swarm optimization algorithm to discrete domain. The new algorithm makes transition for colony characters and search mechanism based on income ratios index of defination. It defines rejection operator to keep the diversity of population, and it introduces Tabu table and defines the learning operator to improve the local refinement ability of the algorithm and the optimal speed. Therefore, the new scheme can better coordinate space exploration and the local refinement ability of ABC algorithm at the same time. What's more, it accelerates convergence speed of the algorithm. The experimental results show that the algorithm can find relatively satisfactory solution in a short time and improve the efficiency of solving the TSP. In the future, we would find more effective ABC algorithms to improve optimum searching method.

Acknowledgements

The authors also gratefully acknowledge the helpful comments and suggestions of the reviewers, which have improved the presentation.

References

[1] Carrabs F, Cordeau JF, Laporte G. Variable neighborhood search for the pickup and delivery traveling salesman problem with LIFO loading. *Informs Journal on Computing.* 2015; 19(4): 2007.

[2] Lenin Kanagasabai, B Ravindranath Reddy, M Surya Kalavathi. Atmosphere Clouds Model Algorithm for Solving Optimal Reactive Power Dispatch Problem. *Indonesian Journal of Electrical Engineering and Informatics (IJEEI).* 2014; 2(2).

[3] K Lenin, B Ravindranath Reddy. Voltage Profile Enhancement and Reduction of Real Power loss by Hybrid Biogeography Based Artificial Bee Colony algorithm. *Indonesian Journal of Electrical Engineering and Informatics (IJEEI)*, 2014; 2(2).

[4] Karaboga D. An idea based on honey bee swarm for numerical optimization. Technical Report TR06. 2014.

[5] Anandhakumar R, Subramanian S, Ganesan S, et al. Modified ABC Algorithm for Generator Maintenance Scheduling. *Electric Power Systems Research*. 2011; 24: 812-819.

[6] Sharma P, Gupta M. *TSP Problem Using Modified ABC Based on Dynamically Division of Bees.*/ Computing Communication Control and Automation (ICCUBEA), 2015 International Conference on. IEEE. 2015.

[7] Hong-Tao YU, Gao LQ, Tian WH. Discrete Artificial Bee Colony Algorithm for TSP. *Journal of Northeastern University.* 2015.

[8] Li S, Gao X, Wang L, et al. A novel maximum power point tracking control method with variable weather parameters for photovoltaic systems. *Solar Energy*. 2013; 97(5): 529-536.

[9] Yu L, Li M, Yang Y, et al. *An Improved Ant Colony Optimization for Vehicle Routing Problem.* Logistics@sThe Emerging Frontiers of Transportation and Development in China, ASCE. 2015: 3360-3366.