Parallel Artificial Bee Colony Algorithm for the Traveling Salesman Problem

Kun Xu, Mingyan Jiang, Dongfeng Yuan
The School of Information Science and Engineering
Shandong University, Jinan, 250100, China
E-mail: xukun_sdu@163.com, corresponding author: jiangmingyan@sdu.edu.cn, dfyuan@sdu.edu.cn

***Abstract*—Artificial Bee Colony Algorithm (ABCA) is a novel swarm intelligence algorithm which a colony of artificial bees cooperate in finding good solutions for numerical optimization problems and combinatorial optimization problems. Traveling Salesman Problem (TSP) is a famous combinatorial optimization problem which has been used in many fields such as network communication, transportation, manufacturing and logistics. However, it requires a considerably large amount of computational time and resources for solving TSP. To dealing with this problem, we present a Parallel Artificial Bee Colony Algorithm (PABCA) in several computers which operation system is Linux based on the Message Passing Interface (MPI). The entire artificial bee colony is divided into several subgroups by PABCA equally. Each subgroup performs an ABCA for TSP on each processor node, respectively. Each sub-colony on every processor node communicates the current best fitness function and parameters of current best fitness function according to ring topological structure during calculation process. Some well-known benchmark problems in TSP are used to evaluate the performance of ABCA and PABCA. Meanwhile, the performance of PABCA is compared with Genetic Algorithm (GA) and Particle Swarm Optimization (PSO). Experimental results show that the PABCA can obtain solutions with equal precision and reduce the time of computation obviously in comparison with serial ABCA. And PABCA have much better performance in contrast with GA and PSO.**

Keyword: parallel algorithm; artificial bee colony algorithm; traveling salesman problem; message passing interface

I. INTRODUCTION

The Traveling Salesman Problem is a famous combinatorial optimization problem known to be NP-complete. It has been used in various application fields such as network communication, transportation, manufacturing and logistics [1-4]. In recent years, many researchers have already proposed many methods in order to solve TSP, such as Artificial Fish Swarm Algorithm (AFSA) [5], Genetic Algorithm [6] and Particle Swarm Optimization [7].

Artificial Bee Colony Algorithm is a novel heuristic algorithm inspired by collective behavior of honeybees to find food sources around the hive. This algorithm has been applied to problems such as optimal tuning of PID controller [8], training neural networks [9] and image edge enhancement [10], etc. A bee colony consists of three kinds of bees: employed bees, onlooker bees and scouts. Each kind of bee is used to execute a kind of function in ABCA.

Parallel implementation can roughly be categorized into two models [11]: one is shared memory architecture; the other is distribution memory architecture. The shared memory architecture mainly has one disadvantage. Parallel algorithm must be confined in only one computer and it cannot make use of different computation resources in different computers, so the shared memory architecture has no scalability. A Parallel Artificial Bee Colony Algorithm based on Message Passing Interface in Linux operation system is proposed to overcome the disadvantage. In our parallel algorithm, the entire artificial bee colony is divided into several subgroups by PABCA equally. Each subgroup performs an ABCA for TSP on each processor node, respectively, so it is distribution memory architecture obviously. Each subgroup communicates parameters and data which are defined and used by PABCA with others using MPI during the calculation process. MPI is a library specification for message-passing, proposed as a standard by a broadly based committee of vendors, implementers, and users. It is designed for high performance on both massively parallel machines and on workstation clusters. [12] In this paper, it is mainly used to implement sending and receiving of data and allocation of tasks in PABCA. All of algorithms and programs are executed in Linux operation system.

II. TRAVELING SALESMAN PROBLEM

The TSP can be described as follows. Given a collection of cities and the cost of traveling between each pair of them, the TSP is to find the cheapest way of visiting all of the cities and returning to its starting point. When a salesman takes his tour he must visit every point once and each point must be visited only one time. In the standard version we study, the travel costs are symmetric in the sense that traveling from city *A* to city B costs just as much as traveling from *B* to *A*.

The cost of tour directly depends on the tour length. So the cost between city *A* and city *B* is calculated as Euclidean distance in (1).

$$d(A,B)=\sqrt{(x_A-x_B)^2+(y_A-y_B)^2}. \qquad (1)$$

Where the parameter x_A, x_B, y_A, y_B represents the *x* coordinate value of point *A*, *x* coordinate value of point *B*, *y* coordinate value of point *A*, *y* coordinate value of point *B*, respectively. $d(A,B)$ refers to the distance between city *A* and city *B*.

The total length of *n* city can be shown in (2).

$$total\_f=\sum_1^{n-1} L_{ij}+L_{1n}. \qquad (2)$$

The parameter $total\_f$ describes the total length which a salesman travels around all cities just one time and return to his start point. Parameter *n* refers to the total number of

© the authors

cities. L_{ij} is the Euclidean distance between the city *i* and the city *j* and L_{1n} is the distance between the last city and the first city .

III. ARTIFICIAL BEE COLONY ALGORITHM FOR TRAVELING SALESMAN PROBLEM

In ABCA, the colony of artificial bees contains three groups of bees: employed bees, onlookers and scouts. Employed bees carry out randomly search in the food sources when the first operation is executed and conduct the neighbor search based on their local positions to find out good sources. Onlookers are waiting in the dancing area for making decision on which food sources to be chosen and then execute neighbor search following the employed bee they choose before. Scouts are placed randomly in other food source when the current employed bee or onlooker is trapping into a local solution. There are three parameters should be defined by users before the execution of algorithm. They are named as population, iteration and limit, respectively. The population defines the number of bees in an entire colony; the iteration is the iterative times which are expected by users; the parameter limit defines a limited value. If a variable exceeds this value, an employed bee will give up the current neighbor searching and a scout should be released.

In general, the number of employed bees and the number of onlookers are equal to half of the population and just allow one scout to be sent out in an iterative process.

The main steps of ABCA are showed as follows:

A. Initialize Artificial Bee Colony

- Initialize population, iteration times and limit times.
- Generate a set of initial solutions and calculate their fitness function.

B. Search for Food Sources

- Send employed bees to carry out the neighbor search.
- Onlookers make judgments depending on fitness function.
- Place scouts on the food source randomly.

C. Judge the Terminal Condition

- If the terminal condition is satisfied, the algorithm is over. Otherwise, the algorithm go back to B

In ABCA, every iterative process contains three stages. First of all, employed bees are initialized randomly and then to evaluate their function fitness. In the next stage, employed bees return to their previous food source and go on exploring the neighborhood of the source, meanwhile, every onlooker chooses a food source offered by employed bees according to its fitness, and starts to explore its neighborhood. During the neighborhood exploration, the employed bees would be substituted by the bees with better fitness. In the last stage, the scouts are sent to the possible food source randomly [9].

The equation which calculates the probability is showed in (3). And the onlookers make decision whether to follow the employed bees or not depend on (3).

$$p_i = \frac{fit_i}{\sum_{1}^{SN} fit_i}. \tag{3}$$

Where the fit_i represents the fitness value of *ith* employed bee, p_i represents the probability which the *ith* employed bee can be chosen by the onlookers, *SN* is the total number of employed bees.

The equation which produces candidate food source around eb_i^j is described in (4). The employed bees and onlookers conduct the neighbor search based on (4).

$$V_i^j = eb_i^j + \zeta \bullet (eb_i^j - eb_k^j). \tag{4}$$

The eb_i^j represents the *jth* parameter of the *ith* employed bee, the V_i^j represents a new solution of the *ith* employed bee; the parameter *i* indicates the *ith* employed bee; *j* represents the *jth* parameter in this employed bee. The parameter ζ is a random number from -1 to 1, and $k \in \{1,2,...,NP\}$; $k \neq i$; $j \in \{1,2,...,D\}$. *NP* is the total number of artificial bees and *D* is the amount of function dimensions.

ABCA for TSP can be described concretely as follows:

- The entire parameters of a bee refer to a path which traverses all cities from the starting point to the end point.
- The value of fitness function is showed in (5):

  $$fitness\_f = 1/total\_f. \tag{5}$$

  Where $total\_f$ is the total length which a salesman travels around all cities just one time and return to his start point. From (5) we can know that, the bee which has higher value of fitness function has the shorter path.

IV. PARALLEL ARTIFICIAL BEE COLONY ALGORITHM

ABCA is a novel heuristic intelligent search method widely used to find proper solutions to a variety of NP problems within a reasonable amount of time (e.g. Traveling Salesman Problem, Job Shop Schedule Problem and so on). However, when they are applied to more complicated problems (e.g. the number of cities increases in TSP), the time required to find an adequate solution increases dramatically. In order to reduce the time, we proposed a PABCA by making use of the parallelism hidden in ABCA. PABCA is an algorithm that combines several computers with the inherent parallelism of ABCA, and enhances the speed of algorithm to obtain a best solution.

A. Parallel Stratagy of PABCA

ABCA has parallel attribution in nature. In this paper, we divide the entire bee colony into several sub-colonies equally according to the number of processor nodes which engaged in calculation. Every sub-colony is assigned to a processor node. Each processor node executes an ABCA to find the

© the authors

global best solution independently. During the calculation process, each sub-colony or processor node communicates their current best solution and gets rid of the current worst solution in fixed number of iteration to make sure that successful solutions can be spread to other nodes and the algorithm can obtain a best solution as far as possible.

In the concrete implementation of PABCA, we choose a computer to be the master node in charge of results display, parameter setting and computational resource allocation. And other nodes are just used to engage in calculation. However, the master node also takes part in calculation as the same as the other nodes with the exception of above works we talk about.

The equation which describes parameters relationship between the entire colony and the sub-colony is shown as follows.

$$sub\_SN = \frac{SN}{num}. \quad (6)$$

$$sub\_\text{iteration} = \frac{total\_iteration}{comuni\_num}. \quad (7)$$

$$sub\_\text{LIMIT} = \text{LIMIT}. \quad (8)$$

Equation (6) shows the population relationship between the entire colony and every sub-colony. sub_SN is the population in every sub-colony; SN is the total number of entire colony and num refers to the number of processor participated in calculation. $total\_iteration$ is the total number of iteration which is expected in PABCA. $sub\_\text{iteration}$ refers to the number of iteration in every sub-colony. $comuni\_num$ is the times of communication of every sub-colony. The relationship between $total\_iteration$ and $\text{sub}\_iteration$ is described in (7). The parameter *LIMIT* defines a limited value. If a variable exceeds this value, an employed bee will give up the current neighbor searching and a scout should be released. The parameter *LIMIT* and *sub_LIMIT* is equal in PABCA.

B. Adjusment of ABCA to Adapt to Parallel Algorithm

The mathematics mechanism description of the employed bees is (4). Equation (4) describes that the ABCA conduct the neighbour search by using the correlation between the current bee and any other bee in the same colony. However, because the entire bee colony has been divided into several sub-colony, the population of bees in every sub-colony is much less than the population of entire bees. If we continue to use (4) to execute the function of employed bees, the diversity of PABCA will declines obviously and the algorithm is very easy to trap into the local best solution. In order to improve this situation, we use a new strategy to conduct neighbour search instead of using (4). Because the entire parameters of a bee refer to a path which traverses all cities from the starting point to the end point, the new strategy is that the algorithm pick up two parameters randomly in entire parameters of a bee and exchange them to generate a new possible solution for the neighbour search.

C. Topological Structure and Communication Mechanism

Topological structure and communication mechanism among every sub-colony is of vital importantce in PABCA and it affects the efficiency and effectiveness of PABCA directly.

The ring topological structure is introduced to implement data communication among each processor node in PABCA. Every processor node will be given a serial number at the beginning of algorithm execution. Every processor node uses this serial number to determine the communicated object. Each processor node just achieves data communication with the node whose serial number next to it. Each node receives a current best solution which is send by the former one and sends out the current best solution itself to the latter one. In addition, each node has a ablity to find out its current worst solution and replace it with a best solution which the node receives from the former one.

Data communication happens after the algorithm excuting a fixed times of iteration. All operations of communications are accomplished by functions which are offered by MPI.

D. The Procedure of PABCA

a) Initialize parameters of PABCA in a master node. It includes population of entire bees, iterations times, limitation value and times of communication during the computation.

b) Calculate parameter for sub-colony. It includes population of sub-colony, times of sub-iteration and those points executed data communications.

c) Every sub-colony receives the data which is sending from the master node.

d) Every sub-colony generates a series of random solutions and calcuates their fitness functions according to parameters which are send from the mater node.Find out the minimun and maximun value of fitness functions in each sub-colony and replace the variable global best solution in every sub-colony with their maximun fitness functions.

e) Each sub-colony exeuctes the model of employed bees .

f) Every sub-colony calcuates the probability which depends on (3) .

TABLE I. PARAMETER OF PABCA AND CALCULATION TIME OF PABCA

	population	iteration	communication	Time of Serial ABCA (ms)	Time of PABCA in two processor nodes (ms)	Time of PABCA in three processor nodes (ms)	Time of PABCA in four processor nodes (ms)
ch130	96	100000	200	161326356	80426646	53767839	40292464
ch150	192	100000	200	198129667	102457252	83523927	61578964
pr107	96	150000	150	107141776	55499631	44869227	33762819

TABLE II. MINIMUM VALUE OF TSP OBTAINED BY DIFFERENT METHORDS

	Best Solution	Serial ABCA	two nodes of PABCA	three nodes of PABCA	four nodes of PABCA	GA	PSO
ch130	6110	9898.550345	9241.396296	9360.887073	9202.420273	12667.405037	39090.87474
ch150	6528	10556.23765	10235.94558	10777.13988	11038.74671	14245.400584	46133.487016
pr107	44303	103367.2301	100946.9481	88963.79192	99244.20649	142208.535127	433900.956305

g) Onlookers make decisions based on the probability obtained in f). And then, it conducts the neighbour search following the employed bee which it chooses.

h) Every sub-colony finds out the current best solution and current worst solution, updates the global best solution in each sub-colony.

i) Every sub-colony carrys out the model of scouts. Judge variables which records times of neighbor search of every employed bee. If some variables exceed a value we set in the beginning of the algorithm, the variable return to zero and correponging employed bees of this variable is replaced by a new randomly generating solution. Otherwise, the variable add one.

j) Every sub-colony judges the iteration times, if the iteration times doesn't reach the limit times, go to step e). If the iteration times reaches the limit times, go to step k).

k) Every sub-colony communicates the best fitness function with each other according to the ring topological structure we talk above. Replace the worst fitness function by using the best fitness function receving from other colonies and the variable which records times of neighbor search of every employed bee returns to zero.

l) Every sub-colony judges the communication times, if the communication times doesn't reach the limit times we set before, go to step e). If the iteration times reaches the limit times, go to step m).

m) Algorithm is over.

V. EXPERIMENTS AND RESULTS

Several benchmark problems are used to evaluate the performance of ABCA and PABCA. The benchmark problems are used in this paper including ch130, ch150 and pr107. All benchmark problems are obtained from International standard database of tsp database. We implemented the PABCA with C language and MPI function library. And PABCA is executed on an experimental platform which composes of four computers. Each computer has one processor of 3.0 GHz dominant frequency and 1.93GB memory.

Parameters should be set in the experiments and calculation time of PABCA is showed in table Ⅰ. From this we can get a conclusion that time of calculation decline as the number of computers increases. The multiple of reducing time is equal to the multiple of increasing processor nodes basically. However, the improvement of time consuming descends and the multiple relationship between time and processor nodes is not corresponding with each other strictly, because time in communication increases as the computer which engaged in calculation increases. Three curves which describe the relationship between time of PABCA and processor number is showed in Fig. 1.

To evaluate the quality of obtained solution in TSP, we compare the solution of PABCA with known best solution, the solution of Genetic Algorithm and the solution of Particle Swarm Optimization. Minimum value of TSP obtained by these different methods is displayed in Table Ⅱ. It is obviously showed in Table Ⅱ that PABCA with multiple processor nodes has approximately same quality solution in comparison with serial ABCA and has much better performance in contrast with the solution which obtained by Genetic Algorithm and Particle Swarm Optimization. However, PABCA is still trapping into a local best solution with limited time in comparison with the known best solution.

VI. CONCLUSION AND DISCUSSION

Artificial bee colony algorithm is a novel heuristic algorithm which excels at searching for a global solution. TSP is a famous combinatorial optimization problem and a NP-hard problem. In the paper, we solve TSP by using PABCA. Three benchmark problems are used to evaluate the performance of ABCA and PABCA. And the performance of PABCA in TSP is compared with GA and PSO. The experimental results demonstrate that PABCA

© the authors

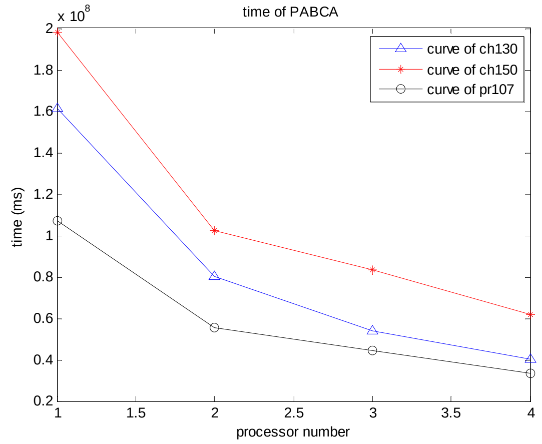

Figure 1. Time of PABCA in different TSP problems.

can improve the speed of calculation and ensure the quality of solution in comparison with serial ABCA and has much better performance in solution in contrast with GA and PSO.

In the future, we mainly have two assignments need to do. On the one hand, the PABCA also traps into a local best solution in comparison with known best solution. So we need to improve the PABCA to make sure that it can find a global best solution. On the other hand, PABCA allocates the calculation resource equally in this paper. In the next step, we will propose a strategy to allocate the calculation resource dynamically according the current calculation load of each processor node.

ACKNOWLEDGMENT

This work is supported by the National Natural Science Foundation of Shandong Province of China under Grant No.ZR2010FM040 and Special Funding Project for Independent Innovation Achievements transform of Shandong Province under Grant No.2009ZHZX1A0108, No.2010ZHZX1A1001.

REFERENCES

[1] J.R.L. Fournier , S.Pierre, "Assigning cells to switches in mobile networks using an ant colony optimization heuristic, " Computer Communications., vol. 28, pp. 65-73, 2005.

[2] J.E. Bella, P.R. McMullen, "Ant colony optimization techniques for the vehicle routing problem," Advanced Engineering Informatics., vol. 18, pp. 41-48, 2004.

[3] T.P. Bagchi, IN.D. Gupta, C. Sriskandarajah, "A review of TSP based approaches for flowshop scheduling," European Journal of Operational Research, pp. 816-854, 2006.

[4] C.A. Silvaa, J.M.C. Sousaa, T.A. Runkler, "Rescheduling and optimization of logistic processes using GA and ACO," Engineering Applications of Artificial Intelligence, vol. 21, pp. 343-352, 2008.

[5] Jiang Mingyan, Yuan Dongfeng, "Artificial Fish Swarm Algorithm and its applications". Science Press, Beijing, China, 2012.

[6] Jiang-wei Zhang,Wen-jian Si, "Improved Enhanced Self-Tentative PSO algorithm for TSP," Proceedings of the Sixth International Conference on Natural Computation, pp. 2638 – 2641, 2010

[7] Wei-xin Ling, Huan-ping Luo, "An Adaptive Parameter Control Strategy for Ant Colony Optimization," Proceedings of the 2007 International Conference on Computational Intelligence and Security, pp. 142 – 146, 2007

[8] Kun Xu, Mingyan Jiang, "Optimal Tuning of Robust Controller Based on Artificial Bee Colony Algorithm," Advanced Materials Research Vols. 562-564, pp. 1668-1672, 30 April. 2012.

[9] Dervi Karaboga and Bahriye Akay, "Artificial Bee Colony (ABC) Algorithm on Training Artificial Neural Networks," Signal Processing and Communications Applications, 2007. SIU 2007. IEEE 15th, Eskişehir, pp. 1-4, 11-13 June. 2007.

[10] Tirimula Rao Benala and Sree Durga Jampala, "A novel Approach to Image Edge Enhancement Using Artificial Bee Colony Optimization Algorithm for Hybridized Smoothening Filters," 2009 World Congress on Nature & Biologically Inspired Computing (NaBIC 2009), coimbatore, India, pp. 1071-1076, 2-4 Spet.2009.

[11] A Grama, A Gupta, G. Karypis, V. Kumar, "Introduction to parallel computing," New York: Addison-Wesley, 2003.

[12] http://www.mcs.anl.gov/research/projects/mpi/