Parallel Ant Colony Optimization Algorithm For Vehicle Routing Problem

Ya Li, Dong Wang, Yuewu Yang, Yan Zhou, and Chen Wu
dept. Computer
School of Electrical&Information Engineering,Foshan University
Foshan Guangdong, China
e-mail: li-bh@163.com

***Abstract*—this paper presents a parallel ant colony algorithm based on Message Passing Interface (MPI) for vehicle routing problem (VRP). At the beginning of the algorithm, a sub-colony is simulated in each processor which independently computes to ensure population diversity. In order to overcome the shortcoming of the ant colony algorithm that is easy to fall into local optimum, a method is designed to judge weather the parallel algorithm get into premature stage. Once premature phenomenon turns up in the whole colony, the local search strategy is used to help the algorithm search the better solution. When the iterative process is over, the optimal solution of all ant colony computed is improved by the neighborhood exchange strategy. Experimental results show that the algorithm is effective and feasible especially for the VRP of large scale cities.**

Keywords- parallel; ant colony algorithm; premature stage

I. INTRODUCTION

Vehicle routing problem is an important segment of modern logistics, as the economy continues to grow, the distribution size of the vehicle increases, the efficiency of the vehicle scheduling system is also need to rise, so the research about vehicle routing problem has important theoretical and practical significance[1]. VRP refers to in known the locations of the distribution center and customers,design a path to get the shortest distance or the least transportation cost as much as possible to achieve the maximum economic benefits of enterprise under the premise of meeting the customer demand and the maximum load of vehicle[2].

VRP is a typical NP-hard; it is difficult to obtain a satisfactory global optimal solution. Currently, heuristic algorithms become the main algorithms for VRP, they are: genetic algorithm, ant colony algorithm, particle swarm optimization, simulated annealing algorithm [3, 4]. Ant colony algorithm (ACA) is proposed by the Italian scholars inspired by the collective behavior of real ant colony [5]. In dealing with large and complex problem, the serial ant colony optimization algorithm is slow, local convergence, while the ant colony algorithm has a feature of parallelism, so the parallel ant colony algorithm becomes an important area of research. Parallel computing has the ability of process large scale data and can improve the efficiency and speed the operation.

In parallel ant colony algorithm, information exchange patterns and cycles is an important factor, which will affect the performance of the algorithm [6]. This paper presents a parallel ant colony algorithm based on Message Passing Interface (MPI), at the beginning of the algorithm; a sub-colony is simulated in each processor which independently computes to ensure population diversity. A method is designed to judge weather the algorithm get local optimum. Once premature phenomenon turns up in whole colony, the optimal solution computed by each sub-colony is gathered and the minimum value among them is selected. Both the minimum value and the pheromone matrix of the sub-colony getting the minimum value are broadcasted to the whole ant colony,based on the minimum value and the pheromone matrix, the iterative process of each sub-colony continues until some criterion for convergence is met. When the iterative process is over, the optimal solution is selected among the sub-colony, which will be improved by the neighborhood exchange strategy in order to get a better solution.

II. THE MATHEMATICS MODEL OF VEHICLE ROUTING PROBLEM

Suppose there are N cities, the city numbered 0 is the distribution center, No. 1 ~ N-1 of the cities are the customers waiting for delivery; Position of customer j is (x_j, y_j), the demand of customer j is $demand_j$; There are M cars, maximum carrying capacity of each car is $Maxload$, the distance between city i and city j is d_{ij},a cargo delivery missions is required to meet a minimum total traveling distance.

Path marking function $path_{ij}^{k}$ =

1, the vehicle k through the $path(i,j)$ in this cycle
0,otherwise

City marking function $route_{j}^{k}$ =

1, the vehicle k service for the city j in this cycle
0,otherwise

The departure load of car k is $setload_k$;

The load of car k in the $path$ (i, j) is $load_{ij}^{k}$;

The total distance of all cars traveled:

$$Y = \sum_{k=1}^{M} y_k \tag{1}$$

In Equation 1,y_k is a total distance of the car k traveled:

$$y_k = \sum_{i=0}^{N-1}\sum_{j=0}^{N-1} path_{ij}^k \times d_{ij} \tag{2}$$

The demand of distribution Center : $demand_0 = 0$

The distance between the city *i* and *j*:

$$d_{ij} = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2} \tag{3}$$

Constraints:

$$Maxload >= \sum_{j=1}^{N-1} demand_j \cdot route_j^k \tag{4}$$

k∈1,2,…M

Ensure that each car's actual loading must not exceed its maximum load.

$$\sum_{k=1}^{M} path_{ij}^k = 1 \quad , i \neq j \tag{5}$$

i∈0,1, …N-1, j∈0,1, …N-1

Ensure that each path is only serviced once by a vehicle.

$$\sum_{k=1}^{M} route_j^k = 1 \tag{6}$$

j∈1,2, …N-1

Ensure that all customers are only serviced once by a vehicle[7].

III. MPI PROFILE

MPI is composed of a group of library functions, the tasks between parallel programs communicate through these functions. These functions include the elementary functions (used to construct the environment and related issues), the basic point messaging functions and collective messaging functions. The first function called in MPI program must be MPI_ Init (), this function can be called only once to initialize MPI programming environment. After all function called, the code must be over in MPI_Finalize (), it makes the program exit MPI programming environment. After MPI_ Init () called, its default communicator property is set to MPI_COMM _WORLD, this attribute includes all the tasks, used to determine the number of tasks and each task's id. MPI_Comm _size ( ) is used to get the number of tasks which need MPI_COMM _WORLD as parameter[8].

IV. THE PRINCIPLE AND IMPLEMENTATION OF PARALLEL ANT COLONY ALGORITHM BASED ON MPI

The algorithm mainly includes the following improvements: construct policy according the whole colony to determine whether the algorithm going into premature stage; Once the algorithm going into this stage, the pheromones of each sub-colony is redistributed in accordance with pre-determined strategy to help the algorithm escape from local optima solution to improve search speed; Finally, the neighborhood exchange strategy is used on the optimal solution to further improve the search accuracy.

A. Premature Judgment Policy

In the traditional serial ant colony algorithm, through the positive feedback of information, the amount of information on the better path is gradually increased. With the increase in the number of iterations, the amount of information on the better path is significantly higher than the other paths. But this usually can not obtain the optimal solution, but better solution existing near the optimal solution. At this time, the algorithm goes into premature stage. For parallel ant colony algorithm, there is the same problem.

The algorithm presented by this paper simulates a sub-colony in each processor. At the early stages of the algorithm, each processor independently calculates. Therefore, the time of each sub-colony going into premature stage is various. In order to measure the time of the whole parallel algorithm going into premature, a constant *N* is set. After each sub-colony finish *N* time's iteration, premature judgment policy starts. Because ant colony algorithm is an evolving process, it usually goes into premature stage in the late of iterations, the constant *N* is usually between the maximum number of iterations *NcMax* and *NcMax* divided by 2. Since the change of the optimal solution of each sub-colony is unchanged or smaller, so computing the average of each sub-colony's optimal solution in the same iteration can reflect the entire state of parallel algorithms. If the average of the optimal solution in the continuous *NbestMax* time's iterations remains unchanged (*NbestMax* is the predefined value), it means the parallel algorithms going into the premature stage.

So after each sub-colony finishes the iterations for *N* times, premature judgment policy starts. In the next iterations, the optimal solution of each sub-colony is gathered and the average of them is calculated. If the averages of the optimal solutions in the continuous *NbestMax* time's iterations remain unchanged, local search strategy is started, else iteration continues. A integer *random* is randomly generated between $N + i* NbestMax + 1$ and *NcMax*, here, *i* is an integer that represents the times of starting premature judgment policy. The initial value of *i* is 0, after each premature judgment policy is started, *i* is incremented. When the number of iteration is *random*, the premature judgment policy starts. Repeat the process until the local search strategy starts or the cycle is end directly into the local search strategy.

B. Local Search Strategy

Before the algorithm goes into premature stage, the sub-colony in each processor take use of the advantages of parallel algorithms, each processor on each sub-colony calculate independently to maintain the diversity of solutions. Once the whole colony goes into premature stage, the sub-colony in each processor requires the exchange of information between them in order to help the algorithm to enhance the local search ability, get rid of local optimal solution, open up new paths to get a better solution near the optimum solution.

Specific strategy is that when the whole colony into the premature stage, the optimum solutions calculated by each sub-colony are gathered and the minimum value is selected. The minimum value and pheromone of the sub-colony get the minimum value are broadcasted to each sub-colony. On the basis of these, each sub-colony continues to iterate following the steps of the ant colony algorithm until it reaches the predetermined times. Finally the optimal solutions calculated by each sub-colony are re-collected, the minimum value is selected to further optimize. Since each sub-colony continues iteration based on the optimal solution the whole colony calculating, the convergence speed of the algorithm will be greatly accelerated. In addition, each sub-ant modify its pheromone according the sub-colony get the minimum value, which can guarantee in subsequent iterations of the algorithm, the search is near the optimal solutions, the advantages of parallel algorithms is used to help the algorithm get rid of local optimal solution, and improve search accuracy.

C. Neighborhood Exchange Strategy

When the Ant colony algorithm calculates the state transition probability, an important factor is involved: $\eta_{ij}(t) = \frac{1}{d_{ij}}$ heuristic function , β is the desired heuristic factor, reflecting the degree heuristic information is valued during the movement of ants. The greater the value is, the similarer the algorithm is to the greedy rule [9].

Therefore the ant colony algorithm has the shortcoming of randomness which is also the shortcoming of the greedy rule. When the algorithm structures a solution, it does not care the "solution" is how to compose and will only concern "solution components." In other words, the "solution components" that the greedy rule get are generally all the components of the optimal solution. But the greedy rule can not combine the "solution components" into the optimal solutions. The greedy rule can only combine the "solution components" randomly, but the optimal solution is generally unique.

In the actual calculation process, the constitute of the shortest path can often be obtained, but not the shortest path.

For example, in Fig. 1, Fig. 2, the vertex is the composition of solution (cities); the connection is the combination mode (vehicle routing route). Figure 1 is the theoretical optimum solution; Fig. 2 is one set of the approximate solutions algorithms find. It is not difficult to find the components of two solutions are the same, except that the combination mode, and as long as the two sides exchanged in Fig. 2, the optimal solution can get. This approach is the neighborhood exchange: use more rational k edges replace the original k edges [10].

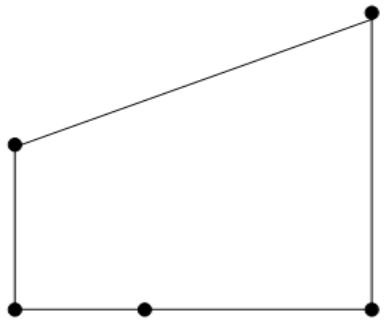

Figure 1. the theoretical optimum solution

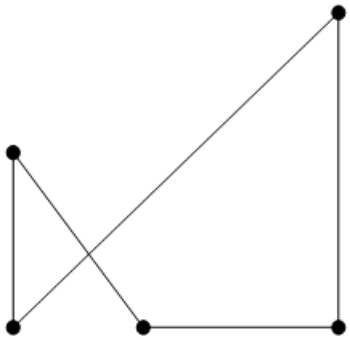

Figure 2. the approximate solution

To eliminate the cross of the path, the proposed algorithm uses the neighborhood exchange strategy for further optimization the optimum solution and rearranged the optimal path.

The adjustment procedure for the optimum route of each vehicle is as follows:

- The cities that the vehicle traveled are sorted by abscissa *x*, If the abscissa x of two cities are the same, the two cities are sorted by the longitudinal distance to the distribution center. After sorted, the minimum distance is adjusted (if better solution appears);
- Then the cities that the vehicle traveled are sorted by ordinate *y*, If the ordinate y of two cities are the same, the two cities are sorted by the lateral distance to the distribution center.After sorted, the minimum distance is adjusted (if better solution appears).

D. Specific Algorithm Steps

- Step 1:No. 0 process reads the size *N* of city, each city's point coordinates (x_j,y_j), the demand of each city ($demand_j$) and vehicle's maximum load(*Maxload*). The distance between cities are calculated according to the formula (3). All of this

information will be broadcasted to the other processes.

- Step 2: Parameter initialization. *NcMax*, *NbestMax* and *N* are set in each sub-colony. The other parameters the ant colony algorithm involved are set for the best configuration values.
- Step 3: The sub-ant colony algorithm calculates in accordance with the max-min ant colony algorithm.
- Step 4: When the number of iterations of each sub-colony have reached *N* times (achieved by the synchronization of MPI_Barrier), the algorithm is judged whether to go into premature stage according to premature judgment policy. If the algorithm goes into premature stage, step 5 is executed, if not, the iteration continues until the algorithm goes into the premature stage or the end of the iteration. If the end of the iteration, step 6 is executed.
- Step 5: The local search is started near the optimal solution according to local search strategy.
- Step 6: The optimal solutions that the sub-ant colony obtained are collected, one of the minimal solution is selected, and the minimal solution are further optimized according to neighborhood exchange strategy.

V. SIMULATION AND ANALYSIS

The experimental environment is: 4 computers(Pentium (R) 4 2.8GHz CPU, 512M RAM);100M switch; Windows XP operating system, MPICH2 libraries, Microsoft VC + +6.0 programming software are installed on each computer.

In order to verify and compare the effect of algorithm, this paper selects two typical examples from internationally VRP problem libraries (Solomon's instances) as the test cases. The parameters of the algorithm involved in two cases: $\alpha = 1, \beta = 3, \rho = 0.8, Q = 60$, *NcMax* = 10000, *NbestMax* = 5, *N* = 8000.

A. Medium-Scale Test

This paper selects TSPLIB, eil51 VRP problem, namely 51 cities vehicle routing problem, as the medium-scale test case. The algorithm runs in two hosts is compared with the standard algorithm of ant colony of stand-alone version, the parameters of the standard algorithm of ant colony are same with the algorithm of this paper. Figure 3 shows the optimal solutions of the two algorithms obtained by 10 experiments. The solid lines represent the optimal solution of the standard algorithm of ant colony of stand-alone version, the dash-dotted line represent the optimal solution of the algorithm of this paper.

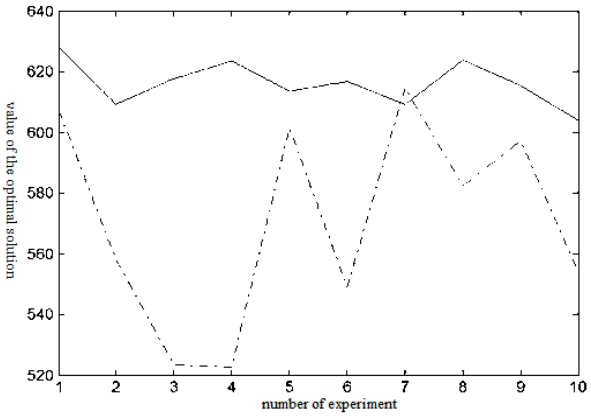

Figure 3. Values of the optimal solution in 10 experiments for eil51 VRP problem

Table 1 gives the number of vehicles, the average solution, the optimal solution, the worst solution and the average time of computation (the average time of the end last process) that the two algorithms obtained in 10 experiments.

TABLE I. ANALYSIS OF EXPERIMENTAL RESULTS FOR EIL51 VRP PROBLEM

algorithm	**vehicle number**	**the average solution**	**the optimal solution**	**the worst solution**	**the average time of computation (/s)**
the standard algorithm of ant colony	5	616.05	603.81	627.86	4.01
the algorithm of this paper	5	570.94	522.65	614.66	14.72

B. Large-Scale Test

This paper also selects TSPLIB, eil101 VRP problem, namely 101 cities vehicle routing problem, as the large-scale test case. The algorithms in two hosts and four hosts are compared with the standard algorithm of ant colony of stand-alone version. Figure 4 shows the optimal solutions of the three conditions obtained by 10 experiments. The solid lines represent the optimal solution of the standard algorithm of ant colony of stand-alone version, the dotted line represents the optimal solution of the algorithm of this paper in two hosts, the dash-dotted line represents in four hosts.

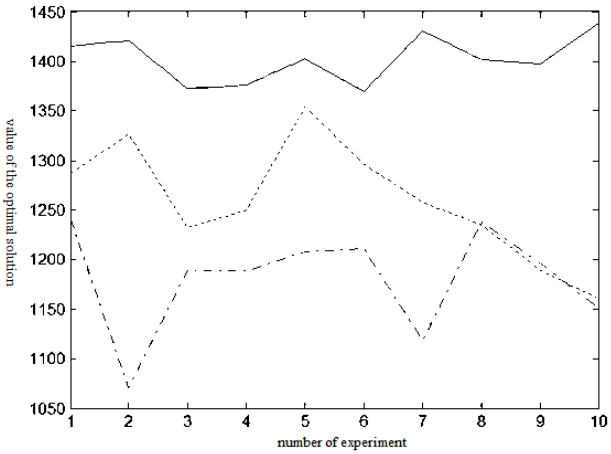

Figure 4. Values of the optimal solution in 10 experiments for eil101 VRP problem

Table 2 gives the number of vehicles, the average solution, the optimal solution, the worst solution and the average time of computation (the average time of the end of the last process) that the three conditions obtained in 10 experiments.

TABLE II. ANALYSIS OF EXPERIMENTAL RESULTS FOR EIL101 VRP PROBLEM

algorithm	vehicle number	the average solution	the optimal solution	the worst solution	the average time of computation (/s)
the standard algorithm of ant colony	14	1402.84	1369.37	1438.61	14.46
the algorithm of this paper (2 host)	14	1259.22	1160.71	1327.20	51.49
the algorithm of this paper (4 host)	14	1181.26	1070.72	1240.25	51.59

C. Analysis Of Experimental Results

From Figure 3, Figure 4, Table 1 and Table 2, it can be seen, both for medium scale or large scale cities, the optimal solutions obtained by the algorithm of this paper are significantly better than the standard algorithm of ant colony of stand-alone version. Especially to the large scale cities, the more the number of hosts running the algorithm, the better the optimal solution obtained. This is due to the more detailed local search near the optimal solution after the algorithm going into premature stage, the higher the quality of the optimal solution.

In terms of time, the time-consuming of this algorithm are slightly more than the standard algorithm of ant colony of stand-alone version, but this extra time is only used to exchange the optimal solution and the pheromone of sub-ant colony after the algorithm entering the premature stage, so the extra time is not a lot. For the large scale cities, with the increase in the number of hosts, the time-consuming of the algorithm are not increased significantly, this is because the larger the size of the city, the smaller proportion of these extra time in the whole computer time. Therefore, For the large scale cities, when more host involved in compute, the algorithm can get better result in a short period of time.

VI. CONCLUSIONS

This paper presents an improved parallel ant colony algorithm for vehicle routing problem. The improved algorithm can effectively avoid the problem of the local optimal solution, get higher quality solution; especially for the problem of larger scale cities, when more hosts involved in compute, the algorithm can get better results in a short period of time. Experimental results show that the algorithm is effective and feasible.

ACKNOWLEDGMENT

This paper is supported by Guangdong Province scientific and technological project (2012B040301032) and Guangdong Province outstanding and innovative young personnel training project (2012LYM_0132).

REFERENCES

[1] JIANG Qi-wei,CHEN Zhi-ya, “On Dynamic Programming Method in the Shortest Route of Logistics Delivery, ”Systems Engineering, vol. 25,Jan. 2007, pp. 27-29.

[2] Wu Jianjun,Liu Jun, “MIXED ANTS ALGORITHM OF ROUTING PROBLEM FOR LOGISTICS DISTRIBUTGION, ” China Civil Engineering Journal, vol. 37,Aug. 2004, pp. 98-101.

[3] WU Jie-ming, “Vehicle Routing Optimization Problem of Logistics Distribution, ”Computer Simulation, vol. 28,Jul. 2011, pp.357-361.

[4] CHEN Yin,XU Hong-mei, “Application of Hybrid Algorithm in Vehicle Routing Problem, ”Computer Simulation, vol. 29,May. 2012, pp.356.

[5] Colorni A, Dorigo M and Maniezzo V, “An investigation of some properties of an ant algorithm, ” Proc. of Nature Conference (PPSN’92), Brussels, Belgium: Elsevier Publishing,1992,pp. 509-520.

[6] ELLABIB I, CALAMAI P andBASIR O, “Exchange strategies for multiple ant colony system, ” Information Sciences, vol. 177,May. 2012, pp.1248-1264.

[7] LI Ya,WANG Dong, “Ant colony optimization algorithm based on chaotic disturbance and neighborhood exchange for vehicle routing problem ,” Journal of Computer Applications, vol. 32,Feb. 2012, pp.444.

[8] Liu Dong,Chang Jing,Wei Wen-hong and Zhao Jie, “Research and Implementation of Parallel Ant Colony Optimization Algorithm Based on MPI, ” Journal of Guangdong University of Technology, vol. 25,Jan. 2008, pp.39.

[9] Duan Hai-bin, ant colony algorithms:theory and applications, Beijing: Science Press,2005.

[10] Marco Dorigo, Thomas Stützle, Ant Colony Optimization, Beijing: Science Press,2007.