An Improved Adaptive Genetic Algorithm for the Multi-depot Vehicle Routing Problem with Time Window

Chun-Ying Liu
Department of Computer and Information Engineering, Heze University, Heze, China
E-mail: lcy810204@163.com

Abstract—In order to improve the efficiency of vehicle objective, the paper addresses the problem of multi-depot vehicle routing with time window. An adaptive genetic algorithm based on the artificial bee colony algorithm is developed for the solution process of the multi-depot vehicle routing problem. The new algorithm provides not only with the strong global search capability, but also the strong local search capability. Give the multiple depots vehicle scheduling model and the coding method of the vehicle route. On the one hand, in order to increase the accuracy of optimization and reduce the probability of trapping in local optimum, adjust adaptively the ratio of the crossover and mutation. On the other hand, the acceptance operators are treated by the simulated annealing. The fitness function with the adaptive penalty coefficient is designed. The simulation results demonstrate that the solving result of the fusion algorithm is more excellent than the other algorithms, and it improves the performance in searching speed and increases the global astringency compared with simple genetic algorithm.

Index Terms—multi-depot vehicle routing problem, genetic algorithm, artificial bee colony algorithm, adaptive optimization

I. INTRODUCTION

The logistics distribution is getting increasingly important in playing role of constructing country economy. How to increase distribution efficiently and reduce its cost is becoming a hot research topic for many foreign corporations and researchers of science. The business logistics and distribution environment are increasingly complex, facing the logistics task more arduous, and many companies have established the multiple distribution centers [1]. For decades, a lot of variants of the vehicle routing problem have been derived, such as multi-depot vehicle routing problem, vehicle routing problem with time windows, open vehicle routing problem, periodic vehicle routing problem, vehicle routing problem with pickup and delivery and so on, a great deal of research results have been achieved. The multi-distribution center vehicle routing problem (MDVRP) has become intelligent logistics scheduling of research hot spots, and it improves the vehicle utilization and reduce the total transport distance. Vehicle routing problem is a very important link in multiple depot logistics scheduling. This problem effectively resolved can improve logistics scheduling scientific level lower transport cost, and increase economic effectively [2]. As a NP-hard problem, the multiple distribution routing plans of multi-depots vehicle scheduling problem will increase exponentially with the adding customers. So it becomes an important studying trend to solve the vehicle scheduling problem with heuristic algorithm [3]. There are two methods for solving the multi-depot vehicle routing problem. One is the exact algorithm, and the other is the heuristic algorithm. The exact algorithm includes the branch and bound method, the cutting plane method, the network flow algorithm, the dynamic programming, etc. The computation complexity increases exponentially with the size of the researched problem solved using the exact algorithm, so its application is limited [4]. The literature defines the new problem and develops an integer programming-based heuristic for the multi-depot vehicle routing [5]. The multi-depot vehicle routing problem is then solved in a sequential and iterative manner by the simulated annealing algorithm embedded in the general framework for the problem-solving procedure [6]. The experts focus mainly on the construction quality of heuristic algorithm. Three hybrid heuristics are presented to solve the multi-depot vehicle routing problem [7]. A multi-level composite heuristic is proposed and two reduction tests are designed to enhance its efficiency [8].

The genetic algorithm (GA) was first promulgated by J. Holland in 1975, and he was inspired by the evolution theory [3]. Using the nature rule of superior win and the inferior being washed out, the genetic algorithm can count the initial solution in the beginning, and get the better solution and feasible solution in the counting process. GA is a commonly used optimization algorithm, and its encoding technique and genetic operations is reasonably easy. The optimal result of GA is independent of the restrictive conditions and the implicit parallelism and global searching are two characteristics of GA. Two hybrid genetic algorithms are developed in for improving the multi-depot vehicle routing problem [9]. A genetic algorithm based on this clustering technique is developed for the solution process of the multi-depot vehicle routing problem [10]. The standard Genetic Algorithm is applied into the vehicle routing problem with common defects of early convergence, and it easily falls into the local
In the meantime, based on the behavior of honey bees all bees were developed intelligent algorithm, the optimization is also a hot research field. Bees algorithm includes the basic bee algorithm, the artificial bee colony algorithm, the bee colony optimization algorithm. The artificial Bee Colony algorithm (ABC) has the global optimal and convergence speed. Therefore, ABC is applied to the multi-depot vehicle routing problem. The using of artificial Bee Colony algorithm provides a new method for MDVRP and prevents it from the local best solution. On the basis of building the model of multi-depots vehicle scheduling problem, this paper studies to solve the problem with the artificial Bee Colony and genetic algorithm, and the new stochastic approach is proposed to solve the vehicle routing problems. By making full use of the locally searching powerful capability of the artificial Bee Colony method, the new algorithm avoids effectively the common defects of the early convergence.

II. MATHEMATICAL MODEL OF MULTI-LOGISTICS CENTER DISTRIBUTION

In the study on the multi-logistics center distribution, the network nodes have the parking lots, distribution centers and the users. The cost of the station the yard, the distribution sites, and the users is the weight of any two vertices. \( H \), \( M \) is respectively the number of the distribution centers and sites, and \( q_{ij} \) is the customer’s demand quantity. \( [a_i, b_i] \) is the time range that the customers demand. \( d_{ij}, \tau_{ij} \) is respectively the distance and the travel time between the points, and \( s_i \) is the time that the vehicle reach. \( \tau_i' \) is the unloading time, and \( K \) is the total number of the delivery vehicles. \( Q_k \), \( D_k \) represents respectively the maximum freight volume and distance, and \( n_i \) is the customer number. \( R_i = \{r_{i1}, r_{i2}, ..., r_{in_i}\} \) is the path set, and \( r_{ie} \) is the consumer sequence. The mathematical model of the multi-logistics center distribution is as follows.

\[
\min Z = \sum_{k=1}^{K} \sum_{i=1}^{n_k} d_{r_{ike}r_{i(e+1)}}
\]

\[
R_i = \begin{cases} 
    \{r_{ie} \in (M + 1, M + 2, ..., M + H), i=1,n_i > 2 \} \\
    \{r_{ie} \in (1, 2, ..., M), i \in (2, 3, ..., n_{i-1}) \} \\
    r_{si} = r_{1i}, i = n_i, n_i > 2 \\
    \{R_i - r_{1i} - r_{ni} \} \cap \{R_i - r_{ki} - r_{ni} \} = \emptyset \quad \forall k \neq i, 0 \leq n_i \leq M + 2 \text{ and } n_i \neq 1, 2 \\
    q_k = \begin{cases} 
    0, & i = (M + 1, M + 2, ..., M + H) \\
    \sum_{j=1}^{k} q_{r_{ij}r_{j(e+1)}} \leq Q_k \\
    \sum_{j=1}^{k} d_{r_{ij}r_{j(e+1)}} \leq D_k
\end{cases}
\]

\[
S_{n(i-1) +} + \tau_{r_{ij}r_{j}} + \tau_{r_{ij}r_{j}} + \tau_{r_{ij}r_{j}} = S_{n_i} 
\]

\[
\begin{cases} 
    a_i = a_i, b_i = b_i; i = (1, 2, ..., M) \\
    a_i = 0, b_i = +\alpha; i = (M + 1, M + 2, ..., M + H) \\
    s_i = s_i, \tau_i' = 1; i = (1, 2, ..., M) \\
    s_i = 0, \tau_i' = 0; i = (M + 1, M + 2, ..., M + H) \\
    \tau_i' = \max(a_i - s_i, 0); i = 1, 2, ..., M \\
    0; i = (M + 1, M + 2, ..., M + H) 
\end{cases}
\]

Soft Time Windows:

\[
d_j = d_j + c \times \max(aj - sj, 0) + d \times \max(s_j - bj, 0) \\
i = 1, 2, ..., M, M + 1, ..., M + H; \\
j = 1, 2, ..., M
\]

Hard Time Windows:

\[
d_j = d_j + c \times \max(aj - sj, 0) + d \times \max(s_j - bj, 0) \\
i = 1, 2, ..., M, M + 1, ..., M + H; \\
j = 1, 2, ..., M
\]

The formula (1) is the total objective function of the multi-logistics center, and the shortest path is the final goal. The formula (2) shows the path stats from the multi-logistics center and return to the multi-logistics center throughout the distribution sites. All paths aren’t mixed and the path number doesn’t exceed the clients form the formula (3) and (4). The formula (5) gives that the multi-logistics centers haven’t the assignment. The formula (6) means the customer demand doesn’t exceed the load. The travel distance in logistics distribution isn’t more than the own maximum by the formula (7), and the customers number of car distribution is equal to the existing from the formula (8). The time of the current distribution center is formed by the arrival, the waiting, the unloading time. The shortest path is the final goal. The formula (10) and (11) said the distribution centers haven’t the time limit. \( c, d \) is the penalty coefficient.

III. PROPOSED SCHEME

A. Encoding Method

The chromosome structure is the most important in genetic algorithm. There are the two encoding methods, the vehicle routing sequence, the customers in multiple depots vehicle routing. In this paper, the two encoding methods are comprehensive and modified. The chromosome is \((G_1, G_2, ..., G_r)\), and \(G_i\) includes four parts. \(\text{start\_depot\_num}\) is the initial station numbers, and \(\text{vehicle\_num}\) is vehicle numbers. \(\text{order\_num}\) is serial number, and \(\text{end\_depot\_num}\) is the final station numbers. \(G_i\) shows that the customers’ vehicles start from \(\text{start\_depot\_num}\), and end at \(\text{end\_depot\_num}\).
order_num is the order of the vehicle transportation path.

For example, the structure Chromosome is $S(A13BB13AA11BA12BB12AB11A)$. The vehicle No. 1 from A station travels A-3-4-1-B and returns to B, and the customers 3, 4, 1 are visited. The vehicle No. 1 from B station travels B-6-5-2-A and returns to A, and the customer 6, 5, 2 is visited. The chromosome’s encoding methods indicate appropriately the open vehicle routing scheduling problem.

B. Population Initialization Rule

Because the model multiple depots vehicle routing contains the constraint conditions, it isn’t feasible for initializing chromosomes by the randomized method. For example, the vehicles must be the same from a starting station to a returning yard, and every car corresponds to a starting and termination station, etc. Set especially the following rules for the chromosome initialization [11].

Rule 1: start_depot_num can be all the station numbers, but its number is the same with end_depot_num. Ensure each station vehicle number unchanged.

Rule 2: Each vehicle can only correspond to one a initial and termination station. If the first and second of every four gene is certain, the fourth is also determined.

Rule 3: The first and the fourth in the $N = (1, 2,..., N)$ is random for every four genes series, and the second belongs to $K_n (n = 1, 2,...,N)$. In addition, the first and fourth should also follow the rule 1 and rule 2.

C. Artificial Bee Colony Algorithm

In order to solve the multivariable function optimization, Karaboga proposed Artificial Bee Colony algorithm (ABC) based on the intelligent foraging behavior of honey bee swarm, and it is a swarm intelligent optimization algorithm. The artificial bee is composed by three bees, which are the leader bee, the follower bee, the scout bee. The amount of leader bee and follower bee are all equal to the food source [12].

The food source is corresponding to the practical problem, and the nectar quantity in food source represents the fitness of the solution. The food source is given up by the leader and scout bee, and the leader bee becomes automatically the scouting [13]. ABC produces firstly the population with $SN$ food source, and $x_i (i = 1, 2,...,SN)$ is the D dimension vector. The leader and follower bee search circularly the food source with the number $MCN$. The leader bee returns the nest after searching and communicating the corresponding food information to the following by dance. According to the food quality offered by the leader bee, the follower bee select randomly the food source by the probability $P_r$, as follows.

$$P_r = \frac{f_i}{\sum_{n=1}^{SN} f_n}$$

$f_i$ is the fitness of the food source. The searching formula of the leader and following bee is as follows.

$$v_k = x_i + R_k (x_j - x_i)$$

$k \in \{1, 2,...,SN\}, j \in \{1, 2,...,D\}, k \neq i, k, j$ is random selection. $R_k \in [-1, 1]$. If the corresponding solution of the food source doesn’t evolve by Limit circulation, the leader bee becomes the scout that initializes it. The new solution $x_j$ is randomly produced by the scouting bee, as follows [14].

$$x_j = x_j^{\prime} + rand (0,1) (x_{\text{max}} - x_j^{\prime})$$

$i \in \{1, 2,...,SN\}, j \in \{1, 2,...,d\}$ is the dimensionality.

D. Adaptability of Crossover and Mutation

- Crossover operation

GA is an optimization method with the advantages by high degree of parallel, the random and self-adapting, which is based on the “Survival of the fittest”. It represents the problem solving as the evolution of chromosomes. The optimization result is searched by selection, cross and mutation and other genetic operations over the generations [15]. GA is a commonly used optimization algorithm, and its encoding technique and genetic operations is reasonably easy. In the genetic operation, the crossover operation is the main method of producing the new individual, and it determines the global search ability of GA. The mutation is the auxiliary method of producing the new individual, and it determines the local search ability of GA. The crossover and mutation cooperate, and complete the all search space. The crossover is the following ways [16].

$$A' = (1-\alpha) \times A + \beta \times B$$

$$B' = (1-\beta) \times B + \alpha \times A$$

$$A'\left(B'\right) < L, \ \text{the new} \ A'\left(B'\right) = L$$

$$A'\left(B'\right) > R, \ \text{the new} \ A'\left(B'\right) = R$$

$A, B, A', B'$ is respectively the individual of the father and offspring, and $\alpha, \beta$ is the random number of $(0, r)$ with uniform distribution. $r$ is the cross coefficient, and its size can control cross range. $L, R$ is the boundary of the optimal parameters. If the cross offspring goes beyond the optimal boundary, and the operation repeats. If the repetitive operation maximizes, the offspring can't still meet the boundary constraint, and the boundary is amended.

- Mutation operation

The mutation operation is as follows.

$$C' = \begin{cases} C + k \times \gamma \times (R - C) & U(0,1) = 0 \\ C - k \times \gamma \times (C - L) & U(0,1) = 1 \end{cases}$$

© 2013 ACADEMY PUBLISHER
makes the individual be eliminated. At the same time, the values of the r and k decreases with the optimization, and GA’s local search capability strengthen gradually. The adaptive strategy makes the GA to maintain the diversity of the population, at the same time, the convergence is certain [19, 20].

E. Annealing Treatment of the Acceptance Operator

The earlier evolution of the standard genetic algorithm has more differences in the individual fitness. The roulette wheel algorithm is easy to fall into the local optimal solution, and the individuals in the population tend to the optimal value in the later. In 1982, Kirkparrick introduced Metropolis criterion into the combinatorial optimization area, and put forward the Simulated Annealing Algorithm for solving the large-scale combinatorial optimization problem. Metropolis discovered that the important sampling method assumed that the new state was accepted by probability in 1953 [21]. In other word, the current state produce the new, and the both energy is \( E_i \) and \( E_j \). If \( E_j < E_i \) and \( P_a = \exp\left(\frac{E_i - E_j}{kT}\right) \) is more than the random number of [0, 1], the new state is the current. Otherwise, the current state is kept [22].

The individuals with good solution are accepted only after crossover and mutation. Though those have strong ability to grasp the overall search process and the search direction is the global optimal solution, it is easy to fall into the local optimal solution. In the other word, if the acceptance operator is treated for annealing, it accepts not only the good solution, but also the deterioration with a certain probability in the population evolution. The individual diversity increases and the search is returned by the above operation, and these cause to jump out of the local optimal solution and form effectively the global optimal method. Therefore, this paper takes acceptance operator for annealing treatment, namely the annealing process is added after the crossover and mutation in the genetic algorithm. If the adaptive value of the offspring is higher than the fathers, and it alternates the fathers after the crossover and mutation. Otherwise, the offspring are accepted by Metropolis Criterion. The probability of acceptance is as follows [23].

\[
P_a = \begin{cases} 
1, & f_{new} \geq f_{old} \\
\exp\left(\frac{f_{old} - f_{new}}{T}\right), & f_{new} < f_{old} 
\end{cases}
\]
generation and improves the local search ability, and the search is the optimal [24].

F. Fitness Function

In the multi-depot vehicle routing problem with time window, when the requirement for time isn’t met, the punishment is implemented. If the customer arrives with time window \([a_i, b_i]\), the vehicle reaches before the time \(a_i\), and the service can’t allow. At the same time, the vehicle wastes time for waiting. If the vehicle reaches after the time \(b_i\), the service is delayed, and the punishment should be implemented. The punishment function is set as follows.

\[
p_i(t) = \begin{cases} 
\alpha_i (a_i - t_i), & t_i < a_i \\
0, & a_i \leq t_i \leq b_i \\
\beta_i (t_i - b_i), & t_i > b_i 
\end{cases}
\]

(25)

t_i is the time that the customer arrives, \(\alpha_i\), \(\beta_i\) is punishment coefficients. The fitness function with the adaptive penalty coefficient is designed. The fitness function is as follows.

\[
f_j(x) = \frac{1}{\sum_{i=1}^{n} d_{c_k-c_h} + \sum_{i=1}^{n} p_i(t) + c_g \times n_k}
\]

(26)
c_g is the fixed cost of the assigned vehicle. The fitness function value is more, and the fitness of the individual is better. By adding a delivery vehicle fixed cost to the fitness function, the contradictory between the number of vehicles and driving distance at the same time is solved effectively.

G. Realization of the Algorithm

The adaptive genetic algorithm with the artificial bee colony method is applied to the optimization of the multiple depots vehicle routing, and the annealing treatment is executed for the acceptance operator. The concrete steps are as follows.

Step 1: Set the initial parameters.

\(M\) is the population size, and \(T_{\text{max}}\) is the largest genetic algebra. \(N_{\text{max}}\) is the receiving times of the new minimum, and \(C_{\text{max}}\) is the maximum times of the inner cycle. The initial population \(G,(1,2,...,M)\) is generated randomly.

Step 2: Calculate the fitness.

Calculate all fitness value in the population, and select the largest individual for the queen.

Step 3: The selection operation.

Select the individual by the roulette and replicate the chromosomes. Calculate the adaptive value \(f_i\) of each chromosome. \(F = \sum_{k=1}^{n} f_k\), \(p_k = f_k / F\), \(k = 1,2,...,n\). \(p_k\) is the selection probability for each chromosome.

Step 4: The crossover operation.

Execute the adaptive crossover operation according to formula (17), (18), (20), (21).

Step 5: Execute the annealing treatment for the acceptance operator after the crossover by Metropolis criterion

Step 6: The mutation operation.

Execute the adaptive mutation operation according to formula (19), (22), (23) for the individual.

Step 7: Execute annealing treatment for the acceptance operator after the mutation by Metropolis criterion.

Step 8: Calculate all fitness value in the new population, and select the largest individual for the best queen.

Step 9: Compare the best in new population and the queen, and select the higher fitness as the new queen. Delete any individuals in offspring population, and replaced the optimal individual.

Step 9: If \(T < T_{\text{max}}\), and return to step (2). Otherwise, the optimization process ends.

In a word, the steps of the multi-depot vehicle routing problem are the hybrid algorithm. The acceptance operators with the near probability produce the offspring by the annealing treatment, and avoid the early good individual to congest the entire population. The outstanding individuals the genetic algorithm produces are more excellent. At the same time, the artificial Bee Colony algorithm adopts the optimal individual which couldn’t be destroyed, and guarantee the convergence.

VI. SIMULATION RESULTS

A. Effectiveness of the Algorithm

In order to verify the effectiveness of the algorithm, use VC+ + 6.0 to design the program, and select the standard testing dataset of Cordeau with 20 examples from Pr01 to Pr20 [25]. All examples can get from http://neo.lcc.uma.es/radi-aeb/webvrp/. Each dataset includes the distribution centers, the customers, the vehicles. The measure for the length adopts the Euclidean distance.

The parameters of the multiple depots vehicle routing optimization based on the adaptive genetic algorithm with the artificial Bee Colony (AGABC) is as follow. \(M = 100, T_{\text{max}} = 600\). The punishment coefficient is the same with the literature [27], \(\alpha_i = 100\), \(\beta_i = 100\). Execute AGABC in 200 times for each example, and the distributes of AGABC’s solution are shown in table I.

The percent of deviation is that the current optimal solution deviates from AGABC’s [26]. From the table I, the best solutions in Pr06, Pr10, Pr13, Pr15, Pr16, Pr19 are superior to the current optimal. The best solutions in Pr01, Pr02, Pr3, Pr5, Pr7, Pr9, Pr11, Pr12, and Pr20 are equated with the current optimal. The best solutions in Pr04, Pr08, Pr14, Pr17, and Pr18 are inferior to the current optimal, but they are close to the optimal. The deviation of the worst and average solutions is less than 2%. AGABC combines the adaptive genetic and the artificial Bee Colony algorithm, so it has the better performance and stability.
There are four distribution centers in Pr05, and each depot has six vehicles. The Pr15 has also four distribution centers, and each depot has five vehicles. The algorithms in common use have TS [27], VNS [28], CNVNS [29]. Execute AGABC and the other algorithms in 200 times in Pr01 and Pr15. The results in solving the path length are shown in table II.

TABLE I. THE DISTRIBUTE OF AGABC’S SOLUTION

<table>
<thead>
<tr>
<th>Set</th>
<th>AGABC’s solution</th>
<th>Current optimum</th>
<th>Relative deviation (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Max</td>
<td>Average</td>
<td>Min</td>
</tr>
<tr>
<td>Pr01</td>
<td>1082.48</td>
<td>1077.12</td>
<td>1074.12</td>
</tr>
<tr>
<td>Pr02</td>
<td>1778.76</td>
<td>1769.24</td>
<td>1762.21</td>
</tr>
<tr>
<td>Pr03</td>
<td>2384.98</td>
<td>2379.61</td>
<td>2373.65</td>
</tr>
<tr>
<td>Pr04</td>
<td>2835.69</td>
<td>2820.42</td>
<td>2818.24</td>
</tr>
<tr>
<td>Pr05</td>
<td>2986.94</td>
<td>2970.56</td>
<td>2965.18</td>
</tr>
<tr>
<td>Pr06</td>
<td>3628.57</td>
<td>3609.24</td>
<td>3589.35</td>
</tr>
<tr>
<td>Pr07</td>
<td>1441.87</td>
<td>1425.86</td>
<td>1418.22</td>
</tr>
<tr>
<td>Pr08</td>
<td>2114.94</td>
<td>2101.34</td>
<td>2099.64</td>
</tr>
<tr>
<td>Pr09</td>
<td>2734.28</td>
<td>2729.65</td>
<td>2724.90</td>
</tr>
<tr>
<td>Pr10</td>
<td>3491.28</td>
<td>3480.36</td>
<td>3468.94</td>
</tr>
<tr>
<td>Pr11</td>
<td>1019.56</td>
<td>1013.84</td>
<td>1005.73</td>
</tr>
<tr>
<td>Pr12</td>
<td>1482.38</td>
<td>1470.32</td>
<td>1464.30</td>
</tr>
<tr>
<td>Pr13</td>
<td>2019.36</td>
<td>2007.69</td>
<td>1994.14</td>
</tr>
<tr>
<td>Pr14</td>
<td>2211.97</td>
<td>2201.35</td>
<td>2195.53</td>
</tr>
<tr>
<td>Pr15</td>
<td>2478.67</td>
<td>2461.23</td>
<td>2449.64</td>
</tr>
<tr>
<td>Pr16</td>
<td>2862.78</td>
<td>2850.32</td>
<td>2837.58</td>
</tr>
<tr>
<td>Pr17</td>
<td>1347.65</td>
<td>1243.23</td>
<td>1242.38</td>
</tr>
<tr>
<td>Pr18</td>
<td>1806.87</td>
<td>1798.64</td>
<td>1793.25</td>
</tr>
<tr>
<td>Pr19</td>
<td>2282.41</td>
<td>2270.36</td>
<td>2260.68</td>
</tr>
<tr>
<td>Pr20</td>
<td>3015.57</td>
<td>2997.68</td>
<td>2991.79</td>
</tr>
</tbody>
</table>

TABLE II. AGABC AND THE OTHER ALGORITHM IN SOLVING THE PATH LENGTH IN PR01

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Pr05</th>
<th>Current optimum</th>
<th>Relative deviation (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Max</td>
<td>Average</td>
<td>Min</td>
</tr>
<tr>
<td>TS</td>
<td>3031.68</td>
<td>3042.46</td>
<td>3038.87</td>
</tr>
<tr>
<td>VNS</td>
<td>2994.45</td>
<td>3008.12</td>
<td>2999.76</td>
</tr>
<tr>
<td>CAVNS</td>
<td>2969.65</td>
<td>2978.87</td>
<td>2972.35</td>
</tr>
<tr>
<td>AGABC</td>
<td>2965.18</td>
<td>2986.94</td>
<td>2970.56</td>
</tr>
</tbody>
</table>

TABLE III. AGABC AND THE OTHER ALGORITHM IN SOLVING THE PATH LENGTH IN PR02

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Pr15</th>
<th>Current optimum</th>
<th>Relative deviation (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Max</td>
<td>Average</td>
<td>Min</td>
</tr>
<tr>
<td>TS</td>
<td>2498.18</td>
<td>2515.39</td>
<td>2502.93</td>
</tr>
<tr>
<td>VNS</td>
<td>2468.76</td>
<td>2481.98</td>
<td>2474.68</td>
</tr>
<tr>
<td>CAVNS</td>
<td>2455.12</td>
<td>2473.34</td>
<td>2462.67</td>
</tr>
<tr>
<td>AGABC</td>
<td>2449.64</td>
<td>2478.67</td>
<td>2461.23</td>
</tr>
</tbody>
</table>

From table II and III, the best, the average, and the worst solution of AGABC in solving the multi-depot vehicle routing are better than the other algorithms. On the one hand, the genetic algorithm with the artificial Bee Colony is introduced, which is well suited for improving the multi-depot vehicle routing problem. On the other hand, the adaptive strategy is designed to reproduce the successful properties. The combination with the heuristic algorithms in AGABC can achieve the optimal solution more than other algorithms.

B. Influence of Adaptability and Artificial Bee Colony Algorithm

In order to verify the influence of the adaptability and the artificial Bee Colony in AGABC, remove them for getting the standard genetic algorithm SGA. Abandon the adaptability for getting the artificial Bee Colony genetic algorithm ABCG, and remove the artificial Bee Colony algorithm for the adaptive genetic algorithm AGA. Test the above algorithms in 200 times and iterate 180 generation. The algorithms set the same parameters and adopt the consistent data. The performance of the algorithms is shown in table IV and V.

TABLE IV. PERFORMANCE OF THE ALGORITHMS IN PR01

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Pr05</th>
<th>Times of the global optimal</th>
<th>Times of the local optimal</th>
<th>Average convergence algebra</th>
<th>Average solution (km)</th>
</tr>
</thead>
<tbody>
<tr>
<td>SGA</td>
<td>27</td>
<td>31</td>
<td>84</td>
<td>3509.78</td>
<td></td>
</tr>
<tr>
<td>ABCG</td>
<td>35</td>
<td>26</td>
<td>117</td>
<td>3456.67</td>
<td></td>
</tr>
<tr>
<td>AGA</td>
<td>41</td>
<td>17</td>
<td>109</td>
<td>3278.78</td>
<td></td>
</tr>
<tr>
<td>AGABC</td>
<td>47</td>
<td>13</td>
<td>71</td>
<td>2970.56</td>
<td></td>
</tr>
</tbody>
</table>

TABLE V. PERFORMANCE OF THE ALGORITHMS IN PR02

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Pr15</th>
<th>Times of the global optimal</th>
<th>Times of the local optimal</th>
<th>Average convergence algebra</th>
<th>Average solution (km)</th>
</tr>
</thead>
<tbody>
<tr>
<td>SGA</td>
<td>26</td>
<td>39</td>
<td>92</td>
<td>2876.68</td>
<td></td>
</tr>
<tr>
<td>ABCG</td>
<td>32</td>
<td>31</td>
<td>120</td>
<td>2674.49</td>
<td></td>
</tr>
<tr>
<td>AGA</td>
<td>48</td>
<td>14</td>
<td>112</td>
<td>2512.78</td>
<td></td>
</tr>
<tr>
<td>AGABC</td>
<td>51</td>
<td>8</td>
<td>67</td>
<td>2461.23</td>
<td></td>
</tr>
</tbody>
</table>

The times of the global optimal convergence is the most in AGABC from the table IV and V, and the Pr05 is 47 times, the Pr15 is 51 times. We can conclude AGABC has the better convergence for adopting the improved adaptive genetic algorithms. On the other hand, the local optimal convergence is the least in AGABC, and the Pr05 has 13 times, the Pr15 has 8 times. The average solution of AGABC is better than the other algorithms. The average convergence algebra reflects the convergence speed, and AGABC has the more convergence speed and the less average route for adding the adaptability and the artificial Bee Colony algorithm. The simulation results indicated that AGABC can improve the convergence speed under the same iterations because it use the optimum individual as a queen in population for the parent select. On the other hand, AGABC has introduced a random population in order to extend search ability and maintain the population diversity.

V. CONCLUSIONS

The distribution of finished products from depots to customers is a practical and challenging problem in logistics management. In this paper, in order to improve the efficiency of the multiple depots vehicle routing, put
forward a new algorithm with the adaptive artificial Bee Colony genetic algorithm. On the one hand, in order to increase the accuracy of optimization and reduce the probability of trapping in local optimum, adjust adaptively the ratio of the crossover and mutation. On the other hand, the acceptance operators are treated by the simulated annealing. The experimental results show that AGABC adding the adaptability and the artificial Bee Colony improves the speed and global convergence under the condition of the invariable convergence result.

ACKNOWLEDGMENT

This work was financially supported by the science and technology projects of the Shandong province universities (J12L055).

REFERENCES

Chunying Liu received the Master of Engineering in Computer Science from the Shan Dong University of Science and Technology in 2008. She is now a lecturer at the Department of Computer and Information Engineering, Heze University. His research interests include the computer network, data-mining, Intelligence Calculation.