Journal of

Software Engineering

ISSN 1819-4311

Academic
Journals Inc.

www.academicjournals.com

OPEN ACCESS Journal of Software Engineering

ISSN 1819-4311
DOI: 10.3923/jse.2016.253.262

Research Article
A Routing Optimization Adaptive Hybrid Genetic Algorithm Research

Zhongping Chen

Hunan Engineering Polytechnic, 410151 Changsha, China

Abstract

In order to meet the needs of goods delivery and pick-up, a mixed integer programming model with simultaneous delivery and pick-up requirement (VRPSDP) was presented for vehicle routing problem. An adaptive hybrid genetic algorithm was developed to solve VRPSDP, a special optimal splitting procedure was used to get the fitness values and a local search was taken as the mutation operator. A novel crossover and a population replacement scheme were designed. Measurement of population diversity and the adjustment rules of mutation probability were also defined. Numerical computations indicated that it was an effective algorithm for solving VRPSDP and VRP.

Key words: Vehicle routing problem, delivery and pick-up, hybrid genetic algorithm, adaptation, local search

Received: October 18, 2015 Accepted: May 09, 2016 Published: June 15, 2016

Citation: Zhongping Chen, 2016. A routing optimization adaptive hybrid genetic algorithm research. J. Software Eng., 10: 253-262.

Corresponding Author: Zhongping Chen, Hunan Engineering Polytechnic, 410151 Changsha, China

Competing Interest: The author has declared that no competing interest exists.

Data Availability: All relevant data are within the paper and its supporting information files.

INTRODUCTION

Vehicle routing problem with simultaneous delivery and pick-up (VRPSDP) is an important extension of classic Vehicle Routing Problem (VRP). Delivery requirement is that goods are delivered to the customer as customer demand from the logistics centerin, pick-up requirement is that vehicle collects goods and returns logistics center from the customer point. The VRPSDP goal is generally that vehicle travels with the shortest total distance, the requirements of each customer is serviced only once, VRPSDP is widespread in the actual logistics. For example, many logistics companies in the port logistics park have dual function of both logistics center and forwarder, when the container terminals are transported to the logistics enterprises, the goods inside the container will be divided into individual customer goods, the goods are shipped to customers through corporate vehicles; these vehicles will also gather customer goods at the same time, these goods are shipped back to the district for consolidation and shipment. Moreover, VRPSDP is also an important study direction of VRP reverse logistics. The VRP reverse logistics was usually considered separately forward logistics and reverse logistics in past study, vehicle idling was serious, it is not conducive to environmental protection. Customers may have both the distribution and recovery needs in many practical cases. For example, soft drinks are distributed for the grocery store, the owner's empty bottles may be recycled at the same time.

The VRPSDP was first proposed in the late 1980s (Min, 1989) there were no published results about these issues in the next 10 years. In recent years, the problem once again aroused widespread interest from scholars, the existing study of the issue is to design approximation algorithms. Insertion algorithm was proposed based on load capacity (Salhi and Nagy, 1999), a customer not only was inserted every time. A plurality of customers can also be inserted at the same time, experiments showed that inserting multiple customers may get better results. On the basis of the definition of VRPSDP strong viable and weak viable state, the heuristic method was proposed based on neighborhood search (Nagy and Salhi, 2005), the overall effect was better than the plug-in algorithm. The corresponding mathematical model was established to study VRPSDP from the perspective of reverse logistics (Dethloff, 2001) and insertion algorithm was given based on the remaining load capacity. The Tabu search was combined with variable neighborhood search algorithm (Crispim and Brandao, 2005) hybrid algorithm was proposed to solve the problem. Because the algorithm was used to solve a packing problem in the construction process of the initial solution, its algorithm can reduce the number of vehicles, but there was a substantial increase in the total distance. After the initial solution was generated by inserting rules, record-to-record travel method was used as a framework and neighborhood search was combined with Tabu table, a heuristic algorithm was presented to solve this problem (Chen and Wu, 2006) experiments showed that the algorithm can be the most optimal solution in small-scale examples. The VRPSDP Tabu search algorithm was designed (Montane and Galvao, 2006) four different neighborhood structure was used in the algorithm, focus search and dispersion search mechanisms were increased based on Tabu frequency. A four-stage heuristic algorithm CLOVES (Cluster nodes, orients nodes along a route, assign vehicles to the routes, using GA to improve the search) was proposed (Ganesh and Narendran, 2007), it was used to solve various types of customer demand VRP including VRPSDP. The construction method, neighborhood search method and Tabu search were used for solving VRP with mixed or delivery requirements simultaneously (Bianchessi and Righini, 2007) experimental results showed that the complex structure and variable neighborhood search algorithm were better. The genetic algorithm was improved for solving the machine scheduling problem and it was used to solve VRPSDP (Vural, 2003).

MATERIALS AND METHODS

Symbol definition: The H for the customer set, $H = \{1, 2,..., n\}$, H_0 is the customer and logistics center set (logistics center point 0), $H_0 = H \cup \{0\}$, K is the maximum number of vehicles in the logistics center that can be used, c_{ij} is the distance between the customer i and j, $c_{ij} = c_{ji,}\ c_{ik}+c_{kj}>c_{ij}$, $i, j = 1, 2,..., n$, q_i is the collection demand for customer i, $i = 1, 2,..., n$; d_i is delivery demand of customer i, $i = 1, 2,..., n$, Q is the maximum capacity of each vehicle, x_{ij} is decision variables and only when the vehicle passes arc (i, j), $x_{ij} = 1$, otherwise $x_{ij} = 0$, $j = 1, 2,..., n$, $i \neq j$, y_{ij} is the total amount of the loaded and collected goods when the vehicle passed the arc (i, j), $i, j = 0, 1,..., n$, $i \neq j$ and z_{ij} is total amount of no-distribution goods on board when the vehicle passed the arc (i, j), $i, j = 0, 1,..., n$, $i \neq j$.

Mixed integer programming model: The VRPSDP goal is to design vehicle path with the shortest total distance, it satisfies the constraint: (1) Each path begins from a logistics center and ends at the logistics center, (2) Each path (each car) maximum cargo capacity must be less than or equal to Q at any moment and (3) Every customer is served only once.

For VRPSDP, three subscript vehicle path models are given in this study (Montane and Galvao, 2006), the label indicates with the serial number of the vehicle are removed, the following more compact two subscript mixed integer programming model is given in Eq. 1-9, respectively:

$$\min \sum_{i=0}^{n}\sum_{j=0}^{n} c_{ij}x_{ij} \tag{1}$$

s.t.

$$\sum_{j=0}^{n} x_{ij} = 1 \tag{2}$$

$$\sum_{j=0}^{n} x_{ij} = \sum_{j=0}^{n} x_{ji},\ i = 1, 2, \cdots, n \tag{3}$$

$$\sum_{j=0}^{n} x_{0j} \le K \tag{4}$$

$$\sum_{j=0}^{n} y_{ij} - \sum_{j=0}^{n} y_{ji} = q_i,\ i = 1, 2, \cdots, n \tag{5}$$

$$\sum_{j=0}^{n} z_{ij} - \sum_{j=0}^{n} z_{ji} = d_i,\ i = 1, 2, \cdots, n \tag{6}$$

$$y_{ij}+z_{ij} \le Qx_{ij},\ i, j = 0, 1,..., n,\ i \neq j \tag{7}$$

$$x_{ij} \in \{0, 1\},\ i, j = 0, 1,..., n,\ i \neq j \tag{8}$$

$$y_{ij}, z_{ij} \ge 0,\ i, j = 0, 1,..., n,\ i \neq j \tag{9}$$

Equation 1 is minimizing the minimizing objective of the total distance, Eq. 2 and 3 ensure that every customer is accessing the same vehicle and is accessed only once, Eq. 4 is up to limit which vehicle can be used, Eq. 5 and 6 are traffic demand constraint for the distribution and collection, two requirements meet for each customer's needs, Eq. 7 ensures that cargo does not exceed vehicle capacity in this path and Eq. 8 and 9 are variable value constraints.

VRPSDP's adaptive hybrid genetic algorithm: Many scholars used genetic algorithms to solve different types VRP, the corresponding study was obtained (Zhao *et al*., 2004; Jiang and Wang, 2006; Zou *et al*., 2004; Prins, 2004). In these genetic algorithm for solving VRP, clients are mostly used as a chromosome coding arrangement, then each vehicle is assigned sequentially by number of customers, until the customer has been allocated so far to meet various constraints. This decoding method is simple but the best solution is lost in the chromosome coding and the genetic algorithm often appears premature convergence in solving VRP, problem is easy to fall into local extreme points, the improvement of the standard genetic algorithm also need to be considered in population diversity and genetic operators. Adaptive hybrid genetic algorithm is designed to solve VRPSDP in this article.

Chromosome structure: The problem solution vector is encoded into a permutation of the n natural numbers {1, 2,..., n} in this study, that is H = (g_1, g_2,..., g_n).

Chromosome decoding method: The chromosome structure may form a large path from a logistics center g_0, n customers are accessed in g_1, g_2,..., g_n order and it ultimately back to the logistics center g_0, as is shown in Fig. 1a. In this study, optimal partitioning method (optimal splitting procedure) of dynamic programming is used to split the shortest path to this big paths, customers will be divided into different sub-path, so that the total length is made the shortest paths to obtain the solution VRPSDP and it is as chromosome fitness value F (H).

Chromosome H = (g_1, g_2,..., g_n) can form a simple directed graph G = (V, A) where, V set is composed of the n+1 nodes (g_0, g_1,..., g_n). The (g_0, g_1, g_{i+1},..., g_j, g_0) can form a path (meet capacity constraints), the arc (g_i, g_j) is feasible, i.e., the arc (g_i, g_j)∈A, i≤j. The right value on the arc defines in Eq. 10:

$$w_{g_i g_j} = \begin{cases} c_{g_0 g_i} + \sum_{k=i}^{j-1} c_{g_k, g_{k+1}} + c_{g_j g_0},\ i < j \\ c_{g_0 g_i} + c_{g_j g_0},\ i = j \end{cases} \tag{10}$$

Figure 1 is an example of the optimal partitioning method for decoding chromosome. Wherein the chromosome structure is H = (g_1, g_2,..., g_5), vehicle capacity Q = 5, the two type demands of each customer indicated in parentheses in the Fig. 1a, the first requirement is for the collection and the second is for the distribution requirements, the digital on line is the distance between two points. Figure 1a is a large path, which is formed by the chromosome structure, Fig. 1b is a directed graph of all possible arc, which is gived by the structure of chromosome H, g_1 g_2: 70 on the arc represents the path (g_0, g_1, g_2, g_0) with the 70 length and to meet the capacity constraints. Thick line part is the shortest path with

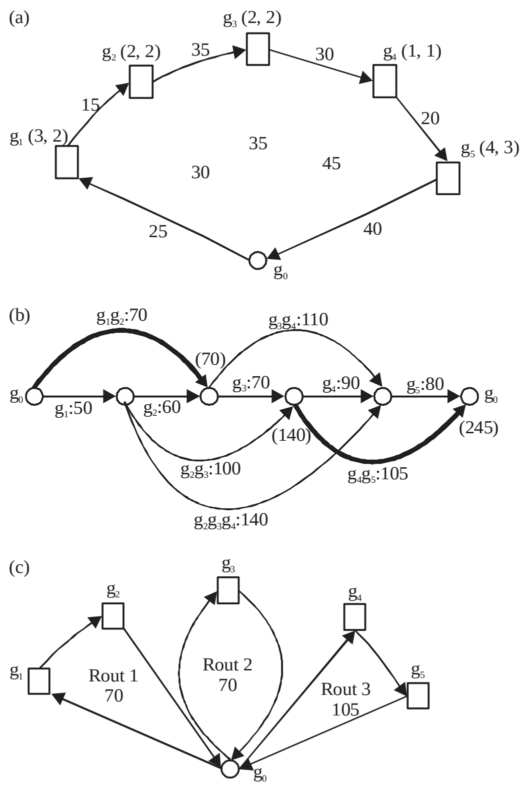

Fig. 1(a-c): Chromosome decoding method example

resource constraints on the directed graph. According to the shortest path in Fig. 1b, customers will be divided into three sub-path, VRPSDP solution is obtained, as shown in Fig. 1c, the optimal objective function value is 70+70+105 = 245.

Since, VRPSDP involves two needs of customers, two resources π and δ need be defined to determine, whether the path is feasible. The π at the customer represents the continue collected capacity after the vehicle passes j. In the initial stage of the iterative determining path feasibility, π is the maximum capacity of each vehicle (π = Q), π decreases with the increase in collect demand. Thus, through arc (i, j), π new value is $\pi = \pi\text{-}q_j$. Similarly, δ at the customer j represents the distribution capacity after the vehicle passes j, the initial stage δ is the maximum capacity of the vehicle (δ = Q). The δ is different to π, δ not only reduce, while distribution demand increases but also it decreases with the increase in collect demand, after the vehicle passes customer j, maximum amount of distribution is not greater than the maximum amount of the collection. Therefore, through the arc (i, j), δ new value $\delta = \min\{\delta\text{-}d_j, \pi\text{-}q_j\}$. The π and δ resources decreases with the iteration, when π<0 or δ<0 a car need to be re-send and the initial value Q are re-set for π and δ the next iteration is done. These are shortest path problems with resource constraints.

Since G does not contain any sub-circle, G shortest path can be obtained by the dynamic programming method, while the H is divided into multiple sub-paths. The algorithm is as follows:

Step 1 : According chromosome $H = (g_1, g_2,..., g_n)$, all possible arcs (g_i, g_j) and arc weights $w_{g_ig_j}$ are obtained as previously described method and the Eq. 10, i.e., digraph G is obtained as shown in Fig. 1b

Step 2 : Dynamic programming method of the solving shortest path is used to obtain the shortest path of a directed graph G from the start point to the end

Step 3 : The various sub-path is obtained according to the shortest path, which is VRPS-DP solution

Chromosome cross-replication: A new cross-way is introduced in this study, the information can better retained in a path of the parent, even when two parents are the same individual, new individuals still be produced, thereby the standard genetic algorithm "Premature convergence" shortcomings are effectively avoided, this is that a conventional crossover are not available. Figure 2 illustrates its operation: Customers which belong to a path are randomly selected from P_1 and P_2, respectively as a mating area. For example, the clients in a path of P_1 are 6 7 8, the client in a path of P_2 is 5 2 1 8, the selected customers are added to the front of each other, the second portion is shown in Fig. 2. Finally, the same natural number is erased in the mating region, the final progeny C_1 and C_2 are obtained.

The achieved cross way requires to record each client belong paths, chromosome in this study consists of two parts: (1) Vehicle routing order and (2) Customer belong path. After each cross, optimal partitioning method is called again the new generation chromosomes were decoded to determine the path to your customers and adapted value is obtained.

Mutation operator: For VRPSDP, neighborhood search (local search) is used as a method of mutation operator to constitute hybrid genetic algorithm (hybrid genetic algorithm). Here the following three methods are mainly used as neighborhood structure:

- **Or-opt:** A customer point i or two consecutive clients point (i, j), which belong to the same path are removed from the original location, they are inserted into point k. Where k can be customer point and can also be a starting point on any path, k and i can be on the same path or on

$$\begin{array}{l} P_1\text{: } 1\ 2\ 3\ 4\ 5\ \mid 6\ 7\ 8\ \mid 9 \\ P_2\text{: } 4 \mid 5\ 2\ 1\ 8 \mid 7\ 6\ 9\ 3 \end{array} \Rightarrow \begin{array}{l} P'_1\text{: } 5\ 2\ 1\ 8 \mid 1\ 2\ 3\ 4\ 5\ 6\ 7\ 8\ 9 \\ P'_2\text{: } 6\ 7\ 8\ \mid 4\ 5\ 2\ 1\ 8\ 7\ 6\ 9\ 3 \end{array} \Rightarrow \begin{array}{l} C_1\text{: } 52\ 1\ 8\ 3\ 4\ 6\ 7\ 9 \\ C_2\text{: } 67\ 8\ 4\ 5\ 2\ 1\ 9\ 3 \end{array}$$

Fig. 2: Cross-way example

different paths. The relative positions of the two customer points (i, j) can be unchanged, they can be swaped from a certain probability

- **2-opt:** Any two client's points (i, j) are taken, which belong to the same path, their locations are interchanged
- **2-exchange:** 2-exchange is a 2-opt extension, client positions of two points i and j are selected respectively and swaped from the two different paths r_1 and r_2

The sub-individual C is formed after cross, neighborhood search will be done in mutation probability pm. In each cycle, possible combinations (i, j), i, j = 1, ..., n, i ≠ j will in turn be searched, each combination (i, j) are sequentially searched by these three neighborhood structure, after an improved solution obtained, the original solution is modified, the no-search neighborhood structures no longer have be searched, the cycle is end, the next cycle is done on the basis of the modified solution and the combination (i, j), which has not been searched modified, possible combinations (i', j') is re-searched, the above neighborhood search method is repeated continuously until solutions can not be any improvement and termination is made. Because (i, j) and (j, i) are two different combinations, the combination number is up to n^2.

Population structure and generating initial population: Because each variability will improve chromosome solution after cross, the diversity of population could be undermined, premature convergence is leaded. In this study, it is design a new group structure, premature convergence generation is prevented with the aforementioned crossover together.

For the population Ω with a scale pop, according to fitness ascending order, the chromosome is sorted to obtain (H_1, H_2,..., H_{pop}). In the initial population generation and group's update process, it is not allowed to generate "clone" (identical) individuals, the population adiversity need to be checked for each new individual. Fitness is taken to characterize the diversity of the population. Specifically, the definition of a constant $\lambda>0$, for a new generation of individuals H'_k, only to meet $\forall H_i \in \Omega$, if $|F(H_i)-F(H'_k)|\geq\lambda$, H'_k can be added to Ω.

If the new generation individual H'_k does not satisfy the diversity of the population, their fitness is better than the best individual H_1 of the present population, H'_k diversity of the population is no longer check, it is directly allowed to replace an individual in the original population. This method is similar to defy guidelines to ensure that the best solutions from the update process will not be discarded, thereby the convergence rate is speeded up.

Initial population is randomly generated. When a new individual H_k is generated, if H_k diversity of the population does not meet the requirements, they need be regenerated. This study stipulates that the regenerated number of single individuals do not exceed σ times, σ = 30. If the requirements of H_k still can be meet inside σ times, the initial generation process of population is stopped, the population size is reduced to k-1, (i.e., pop = k-1). This situation will happen only when the problem of relatively small-scale or set of pop too large.

Adaptive mutation probability: Adaptive mutation probability of genetic algorithm has significant improvements in maintaining population diversity and optimization capabilities. Adaptive genetic algorithm with the original mutation probability is directly determined by the fitness value, it is different from mutation probability of this study, this mutation probability is decided by number k of individual fitness value from small to large order, the individual is generated by the cross in the original population. It is defined as follows in Eq. 11:

$$p_m(k) = \begin{cases} k/(5\times \text{pop}), & 0<k\leq[\text{pop}/2] \\ 0.10, & [\text{pop}/2]<k\leq \text{pop} \end{cases} \tag{11}$$

Adjustment rule: In the process of updating population, if the current number of times that the best solution remains unchanged is more than half a predetermined maximum number of algorithms, the probability of individual variability need to be increased, a new mutation probability is taken $p'_m(k) = 1.5\ p_m$; once the current best solution is improved, the mutation probability will be restored into $p'_m(k) = p_m(k)$.

Algorithm steps: The α_{max} is maximum number of crosses, β_{max} is maximum number of the current unchanged best solution, they are as the algorithm termination conditions.

Step 1 (Initialization): Generated scale is pop, it meets the initial population 0 of the population diversity, the fitness value is calculated for each individual and the fitness values are sorted from small to large order; α, β are assigned to 0.

Step 2 (Select): Father individuals (P_1, P_2) are generated by tournament selection method, two individuals are selected randomly from population, each parent individual is the merit between two individuals.

Step 3 (Cross): Father individuals (P_1, P_2) are intersected and a individual C is randomly selected from the two sub-cross individuals, it is used as the variability and the updated object.

Step 4 (Updated individuals are selected): A individual H_k is randomly selected from the second half of population, $k \geq [pop/2]$, it is as the individual that need to be updated in n individuals of the original population, thus the best individuals are retained.

Step 5 (Adaptive mutation probability): According to adaptation value F (C) ranking position of offspring C in Ω, mutation probability p_m is calculated by Eq. 11, if $\beta > 0.5\ \beta_{max}$, $p_m = 1.5\ p_m$ is obtained by the adjustment rules.

Step 6 (Variation): According to the mutation probability p_m or p'_m, C is mutated with a neighborhood search method to get new chromosome M and only when chromosome M meets diversity of the population (refers to the original population, in which H_k was removed), C is replaced with M.

Step 7 (Contempt guidelines): If $F(C) < F(H_k)$, H_k is replaced with C, the best solution remains the same frequency $\beta = 0$, cross number α plus 1, the population is reordered to go to Step 9, otherwise continue.

Step 8 (Update): If the chromosome C satisfy the diversity of population, H_k is placed by chromosome C, population is reordered α, β, respectively, plus 1, go to Step 9; otherwise C does not satisfy the diversity of the population directly to Step 9.

Step 9 (Termination or transfer): If $\alpha \geq \alpha_{max}$ or $\beta \geq \beta_{max}$, algorithm ends, otherwise go to step 2.

RESULTS AND DISCUSSION

To test the effectiveness of the algorithm, VRP and VRPSDP typical examples are used to test in this study, they are compared with different optimization algorithms.

To test the algorithm on the sensitivity and effectiveness in solving various parameters, different values of the parameters were set as follows: $pop = 30, 50, \lambda = 0.3, 0.5, 0.8;$ $p^l_m(k)$ and $p^h_m(k)$ are given by the Eq. 12 and 13, $p^l_m(k)$, $p_m(k)$ and $p^h_m(k)$ represent respectively low, middle and high three cases of the mutation probability; $\alpha_{max} = 20000$, $\beta_{max} = 10000$ are unchanged.

Experiment: The various combinations of the above parameters are tested for instance the size of 100 VRPSDP (CMT3X) (Salhi and Nagy, 1999) and they run 10 times at random for each combination of parameters, the results are shown in Table 1. Wherein, after algorithms runn for 10 times, value and time was respectively average calculation time and average value of the resulting solution, the time unit is second.

$$p^l_m(k) = \begin{cases} k/(10 \times pop), & 0 < k \leq [pop/2] \\ 0.05, & [pop/2] < k \leq pop \end{cases} \quad (12)$$

$$p^h_m(k) = \begin{cases} 3k/(10 \times pop), & 0 < k \leq [pop/2] \\ 0.15, & [pop/2] < k \leq pop \end{cases} \quad (13)$$

Analysis: Table 1 shows that, in the same pop and λ, computing time will grow significantly with increasing the mutation probability, due to the number of neighborhood search is increased, the average search quality of algorithm has also been improved. At the same mutation probability and λ except a pair of data, λ = 0.3 and mutation probability p, average search quality will be improved with the increase of population size, computing time will increase, which is due to increased algorithmic search space. The λ impact of changes is not clear on the quality of the search algorithm, in most cases the computing time will increase with the increases. In addition, after the population size, mutation probability, λ and other parameters are fixed, the results of other experiments show that α_{max} and β_{max} role in the algorithm is similar to evolution generation number in the standard genetic algorithm.

After the search quality and computation time are weighed, the algorithm parameters are taken as pop = 50,

Table 1: Calculated results of AHGA algorithm solving CMT3X

		$p^l_m(k)$		$p_m(k)$		$p^h_m(k)$	
pop	λ	Value	Time	Value	Time	Value	Time
50	0.3	734.27	158.52	728.11	316.46	724.67	601.62
	0.5	731.45	173.43	727.35	323.77	726.32	614.85
	0.8	733.81	173.95	729.06	361.75	727.11	717.21
30	0.3	732.73	144.81	731.24	289.18	728.21	526.11
	0.5	735.16	156.65	728.87	273.24	727.96	521.94
	0.8	736.63	164.12	730.79	323.97	727.35	550.07

Table 2: Comparison of several algorithm’s optimal values for E-n51-k5 VRP

Algorithm	Optimal value	Algorithm	Optimal value
Saving search algorithm	585.00	Scan search algorithm	532.00
Allocation algorithm	524.61	Site selection algorithm	524.90
Two phase algorithm	547.00	Tree search algorithm	534.00
Tabu search algorithm	524.61	Simulated annealing algorithm	528.00
Neighborhood search algorithm	524.61	Genetic algorithm	524.80
Ant colony algorithm	524.61	Predatory search algorithm	524.61
our AHGA	524.61		

Table 3: Compared effect with other heuristics optimal algorithm

Algorithm	L = 100	L = 150	L = 199
Saving search algorithm	866.00	1204.00	1540.00
Scan search algorithm	851.00	1079.00	1389.00
Allocation algorithm	833.00	1014.00	1386.00
Site selection algorithm	832.90	1088.60	1461.20
Tabu search algorithm	835.00	1052.00	1354.00
Genetic algorithm	840.70	1032.50	1375.40
Ant colony algorithm	840.50	1060.90	1461.20
Predatory search algorithm	841.70	1061.80	1370.30
Our AHGA	826.41	1031.63	1327.78

$\lambda = 0.5$, $\alpha_{max} = 20000$ and $\beta_{max} = 10000$ in the following experiments. Mutation probability p_m is determined by the Eq. 11.

Experiment example 1-VRP: When VRPSDP collected demand of all customers are 0:00, it can be translated into a general VRP, therefore our algorithm can also be used to solve general VRP. For E-n51-k5 VRP test case was proposed by Christofides and Eilon (Salhi and Nagy, 1999), the examples are used to test our AHGA algorithm, Predatory search algorithm (Jiang and Wang, 2006), as well as the other optimization algorithms, which were summarized in literature (Jiang and Wang, 2006). Table 2 shows the results were obtained after these algorithms randomly run 10 times. Data in the Table 2 shows that the obtained optimal value reached the best known solution in the proposed algorithm, the corresponding five vehicle path was exactly the same with the literature (Jiang and Wang, 2006).

In order to verify the effectiveness of our AHGA algorithm for solving large-scale VRP, VRP were calculated with size L, respectively 100, 150 and 199 which Christofides Mingozzi and Toth proposed (separately calculated 10 times), our AHGA results were compared with the literature summary optimization algorithms (Jiang and Wang, 2006). From Table 3 shows that the compared results with other algorithms, this algorithm can be morethan other algorithms in best solution, it fully shows the effectiveness of our AHGA algorithm. Because there is no summary of the relevant algorithm computation time in the literature (Jiang and Wang, 2006), comparison of computation time is not done. Even when the VRP problem with scale 199 is solved by our AHGA algorithm, maximum calculation time is also within 22 min, it is acceptable.

Experiment example 2-VRPSDP: Christofides, examples of the VRP were about transformation, it was converted into VRPSDP instance, the scale was 50, 75, 100, 120, 150 and 199 (Salhi and Nagy, 1999). The "Y" type problem is instances of interchangeable collection and delivery demands in "X" type problem. Since then, intelligent optimization algorithms, which is mostly related to solving VRPSDP were tested by this instance, the existing best results were made on different scale (Chen and Wu, 2006; Montane and Galvao, 2006). Our AHGA algorithm were compared with the results of the literature (Chen and Wu, 2006; Montane and Galvao, 2006), it was also calculated to 10 times and the best solution was obtained, the compared the results are shown in Table 4. Where, RTR is record-to-record travel proposed algorithm (Chen and Wu, 2006), TS is Tabu search algorithm (Montane and Galvao, 2006), AHGA is proposed in this study. Value was total path length which corresponding algorithms obtained the best solution, veh wasthe number of the used vehicles, time was the required time of the algorithm running, the time unit is second(s). "Total" line represents sum of the best solutions which each algorithm obtained in all instances.

Due to the randomness of the proposed algorithm, the stability of the algorithm is tested, Table 5 shows the obtained solution average of 10 times calculation, the worst value and standard deviation, two columns under average were solving mean and mean time, the two columns under worst were the worst value and maximum computing time to solve. For ease of comparison, Table 5 once again shows the best solution, which were given in the literature (Chen and Wu, 2006; Montane and Galvao, 2006).

Comparative analysis: Table 4 shows that the proposed algorithm results exceed the best available solution in 10 instances of 14 instances VRPSDP and AHGA results in example CMT12X are basically the same with the best available solution. Only in these three instances of CMT2Y, CMT3Y and CMT4Y, algorithm does not meet the best known solution, but the gap is very small between the AHGA result and the best known solution, the relative error is less than 0.5%. In comparison the number of the used vehicles is more than the best known solution except the number of the used vehicles in example CMT5Y, AHGA results in the other instances already have reached the number of the used vehicles in the best known solution, especially for instance

Table 4: Comparison of the best existing solutions in VRPSDP examples

		RTR			TS			AHGA		
Examples	Scale	Vehicles	Value	Time	Vehicles	Value	Time	Vehicles	Value	Time
CMT1X	50	3	478359.00	7.74	3	472	3.73	3	466.77	54.57
CMT1Y	50	3	480.78	7.81	3	470	4.37	3	469.80	68.81
CMT2X	75	6	688.51	24.86	7	695	6.91	7	687.95	158.58
CM2TY	75	6	679.44	12.02	7	700	7.61	6	682.39	194.97
CMT3X	100	5	744.77	94.06	5	721	11.04	5	720.08	329.51
CMT3Y	100	5	723.88	120.66	5	719	12.01	5	720.91	296.39
CMT4X	150	7	887.00	501.95	7	880	24.60	7	862.11	799.03
CMT4Y	150	7	852.35	406.32	7	878	29.07	7	854.75	844.52
CMT5X	199	10	1089.22	1055.83	11	1098	51.50	10	1075.21	1364.58
CMT5Y	199	10	1084.27	771.71	10	1083	56.21	11	1051.71	1469.73
CMT11X	120	4	858.57	321.08	4	900	18.17	4	844.17	710.16
CMT11Y	120	5	859.77	230.72	5	910	18.04	4	847.78	632.02
CMT12X	100	6	678.46	46.83	6	675	12.23	6	675.37	226.79
CMT12Y	100	6	676.23	56.35	6	689	12.80	6	672.97	355.24
Total			10781.84			10890			10631.97	

Table 5: AHGA stability

						AHGA				
		RTR		TS		Average		Worst		
Examples	Scale	Value	Time	Value	Time	Value	Time	Value	Time	Standard deviation
CMT1X	50	478.59	7.74	472	3.73	469.49	50.66	469.80	75.52	2.44
CMT1Y	50	480.78	7.81	470	4.37	469.83	65.64	471.53	86.74	2.05
CMT2X	75	688.51	24.86	695	6.91	695.09	155.77	702.52	207.52	5.41
CMT2Y	75	679.44	12.02	700	7.61	693.01	154.72	701.99	201.28	5.61
CMT3X	100	744.77	94.06	721	11.04	727.35	323.77	730.77	352.65	2.52
CMT3Y	100	723.88	120.66	719	12.01	730.61	344.81	736.96	365.18	4.03
CMT4X	150	887.00	501.95	880	24.60	869.10	863.47	880.15	902.11	6.44
CMT4Y	150	852.35	406.32	878	29.07	874.52	833.26	886.09	874.12	12.36
CMT5X	199	1089.22	1055.83	1098	51.50	1078.22	1312.11	1085.27	1473.14	10.12
CMT5Y	199	1084.27	771.71	1083	56.21	1065.47	1379.33	1080.18	1525.17	13.35
CMT11X	120	858.57	321.08	900	18.17	865.60	582.89	879.46	726.77	12.56
CMT11Y	120	859.77	230.72	910	18.04	867.75	555.96	882.89	747.21	13.30
CMT12X	100	678.46	46.83	675	12.23	677.50	312.50	683.51	331.18	4.97
CMT12Y	100	676.23	56.35	689	12.80	678.17	283.87	681.77	96.52	3.97
Total		10781.84		10890		10761.71		10871.16		

CMTl1Y, our algorithm vehicles are fewer than the best known solution. The sum of all best solutions in AHGA algorithm is also less than other algorithms in 14 instances, it increases by 1.4 and 2.4%, respectively than RTR and TS. In the running time, although computation time in AHGA algorithm is longer than TR and TS, it is still within the acceptable range.

Table 5 shows that the average value and the worst value of our algorithm solution b for 10 times still be better than or to achieve the best available solution in these 4 instances of CMT1X, CMT4X, CMT5X and CMT5Y, AHGA average is better than the best available solution for solving instances CMT1Y. In the rest of the practice and the gaps of the average and the worst value are small between AHGA algorithm solution and the best known solution, the relative error is less than 3 and 4%. Apart CMT3Y, the average of AHGA algorithm solution is better than the RTR algorithm and TS algorithm in other instances. With increasing instance's size, standard deviation of solutions has tended to increase, but also the largest standard deviation is within 14, it has described that AHGA is stable in solving VRPSDP. In addition, the average sum of our algorithm solutions is also less than other algorithms for solving the 14 instances, it is better 0.2 and 1.2%, respectively than RTR and TS. Table 5 shows the average computation time and maximum computing time, the computation time of the algorithm is still long than RTR and TS, but even the longest computing time is also acceptable range.

Figure 3 shows the optimal evolution processes of AHGA algorithm to seek the best solution on the CMT3X and CMT4X. Since this algorithm does not "Evolution generation number" concept when the optimal evolution diagrams are drawed,

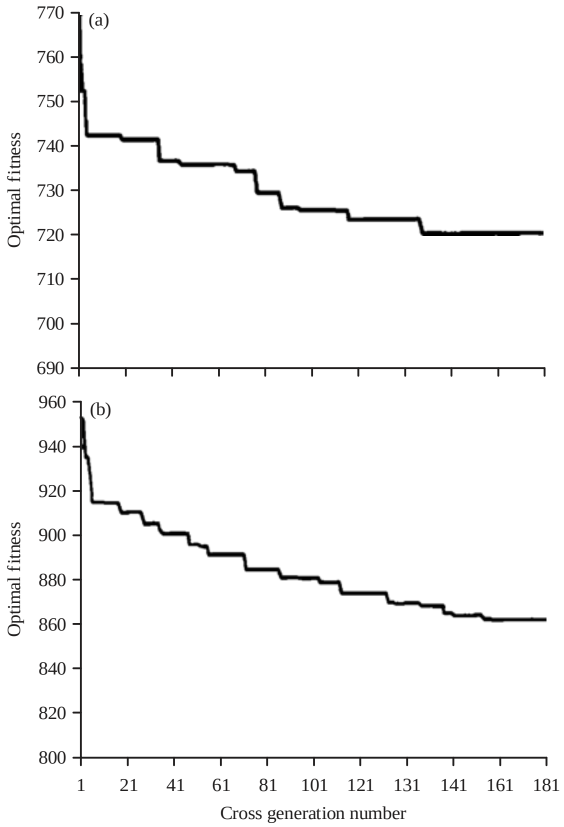

Fig. 3(a-b): Optimal evolution figure of AHGA solving (a) CMT3X and (b) CMT4X

cross for every 100 generations is as a generation, the best value of the current population is taken as the best value of this generation, the best evolution is showed in Fig. 3. As can be seen from Figure 3 that the adaptive hybrid genetic algorithm in this paper has a faster convergence rate.

During the experiment, if the optimal partitioning method is not used for decoding, only according to natural number arrangement of chromosomes and the vehicle capacity limits to decoded, it will greatly reduce the quality of the algorithm. If a simple two exchange variation is used to substitute neighborhood search algorithm as the mutation operator, computation time is reduced but the quality of the solution has been significantly reduced. In addition, some of the more difficult or large-scale problems are solved, adaptive changes of mutation probability may increase the computation time but it can improve the quality of the solution.

CONCLUSION

Adaptive Hybrid Genetic Algorithm (AHGA) is designed to solve VRPSDP in this study. Replacement ranks are used as chromosome structure in algorithm, it is convert into a simple directed figure, dynamic programming method for solving the shortest path is used to get the VRPSDP best solution at this chromosome structure. A mutation operator is obtained by neighborhood search algorithm, which is composed of a variety of neighborhood structures, such as 2-opt, 2-exchange and or-opt other components, the probability of individual variation adaptively changes in ascending order of number k of its fitness value and it is combined with innovative crossover and update policies, as well as the structure of population diversity and many other technologies, the effect of genetic algorithms are greatly improved to solve. The VRP and VRPSDP two sets of experiments are compared with existing optimization algorithms in their result of the comparison example, in most instances, the proposed algorithm outperforms the existing methods, it shows that the algorithm is efficient algorithm for solving the VRP and VRPSDP.

This algorithm still needs to continue to improve in order to be applicable to other types of VRP, such as VRPSDP and VRP with time windows, mileage and vehicle restrictions. In addition, our algorithm did not get comparative advantage in the calculation of time and therefore, the effect of the solution need to be maintained and improved continuely to shorten the calculation time.

REFERENCES

Bianchessi, N. and G. Righini, 2007. Heuristic algorithms for the vehicle routing problem with simultaneous pick-up and delivery. Comput. Operat. Res., 34: 578-594.

Chen, J.F. and T.H. Wu, 2006. Vehicle routing problem with simultaneous deliveries and pickups. J. Operat. Res. Soc., 57: 579-587.

Crispim, J. and J. Brandao, 2005. Metaheuristics applied to mixed and simultaneous extensions of vehicle routing problems with backhauls. J. Operat. Res. Soc., 56: 1296-1302.

Dethloff, J., 2001. Vehicle routing and reverse logistics: the vehicle routing problem with simultaneous delivery and pick-up. OR-Spektrum, 23: 79-96.

Ganesh, K. and T.T. Narendran, 2007. CLOVES: A cluster-and-search heuristic to solve the vehicle routing problem with delivery and pick-up. Eur. J. Operat. Res., 178: 699-717.

Jiang, Z. and D. Wang, 2006. Predatory search algorithm for vehicle routing problem. Comput. Integr. Manufact. Syst., 12: 1899-1902,1908.

Min, H., 1989. The multiple vehicle routing problem with simultaneous delivery and pick-up points. Transp. Res. Part A: Gen., 23: 377-386.

Montane, F.A.T. and R.D. Galvao, 2006. A tabu search algorithm for the vehicle routing problem with simultaneous pick-up and delivery service. J. Comput. Operat. Res., 33: 595-619.

Nagy, G. and S. Salhi, 2005. Heuristic algorithms for single and multiple depot vehicle routing problems with pickups and deliveries. Euro. J. Operat. Res., 162: 126-141.

Prins, C., 2004. A simple and effective evolutionary algorithm for the vehicle routing problem. Comput. Operat. Res., 31: 1985-2002.

Salhi, S. and G. Nagy, 1999. A cluster insertion heuristic for single and multiple depot vehicle routing problems with backhauling. J. Operat. Res. Soc., 50: 1034-1042.

Vural, A.R., 2003. A GA based meta-heuristic for capacitated vehicle routing problem with simultaneous pick-up and deliveries. Master's Thesis, Sabanci University, Istanbul, Turkey.

Zhao, Y.W., B. Wu, L. Jiang, H.Z. Dong and W.L. Wang, 2004. Double populations genetic algorithm for vehicle routing problem. Comput. Integr. Manufact. Syst., 10: 303-306.

Zou, T., N. Li and D. Sun, 2004. Genetic algorithm for variable fleet vehicle routing problem with time window. Syst. Eng. Theory Pract., 24: 134-138.