Proceedings of the 26th Chinese Control Conference
July 26-31, 2007, Zhangjiajie, Hunan, China

A Hybrid Intelligent Algorithm for Vehicle Pick-up and Delivery Problem with Time Windows*

Liao Li, Wu Yaohua, Hu Hongchun, Xiao Jiwei
The Logistics Institute of Shandong University, P. R. China
E-mail: liaoli@sdu.edu.cn

Abstract: In this paper a hybrid intelligent algorithm is presented to solve the pick-up and delivery problem with time windows (PDPTW), which combines cluster algorithm, genetic algorithm and tabu search. At first, the multi-vehicle PDPTW problem is transformed into several single-vehicle PDPTW problems by using clustering algorithm, then initial solution for tabu search is constructed by arranging the best solutions of single-vehicle PDPTW problems which obtained by using genetic algorithm, at last the optimal solution is obtained by performing tabu search. In addition, a gradually increase strategy of the number of cluster from the minimal feasible number is proposed, to guarantee the minimization of fleet. The computational results demonstrate the effectiveness of the proposed method to solve the PDPTW problem.
Key Words: Pick-up and delivery problem with time windows (PDPTW), Cluster, Genetic algorithm, Tabu search

1 INTRODUCTION

The optimization of vehicle routing and scheduling is very important to the logistical system, and it is beneficial for the company to reduce costs and strengthen the competitive power. In the last two decades, many researchers have worked on it and significant achievements have been achieved. However, the reference on the pick-up and delivery problem is not abundant, and there is rather poor of contributions on such problem with time windows[1].

This paper focus on the pick-up and delivery problem with time windows (PDPTW), which is the following problem: a set of clients is located on a transportation network, each client requires either a delivery or a pick-up operation. The service is provided by a fleet of vehicles, the goal is to find a set of vehicle routes (which vehicles should visit which customers and in what order exactly) to minimize the overall distance, cost, time, or vehicles, etc., subject to some constraints (time windows constraints, vehicle capacity constraints, driver distance constraints, driver time constraints and so on).

As one variation of the classical vehicle routing problem, PDPTW is a NP-hard problem. In the existing literature, there are mainly two methods to solve PDPTW problem, exact optimization methods and artificial intelligence methods. The most successful exact optimization algorithms are K-tree and Lagrangian relaxation, and the exact methods are difficult to find the optimal solution for practical applications of large scale problems in an acceptable execution time. The top representatives of intelligent methods for PDPTW problem are simulated annealing (SA), genetic algorithm (GA) and tabu search (TS)[2-7].

In this paper we further investigate and develop a hybrid intelligent method for PDPTW, which takes advantage of the dimensionality degrading of cluster algorithm, self-adaptability of genetic algorithm and neighborhoods searching of tabu algorithm. The hybrid intelligent algorithm transform the multi-vehicle PDPTW problem into several single-vehicle PDPTW problems at first by using clustering algorithm at first, then construct initial solution for tabu search by arranging the best solutions of single-vehicle PDPTW problems which gained by using genetic algorithm, and at last get the optimal solution by performing tabu search. In addition, a method to gain the minimal number of cluster is provided, from which the number of cluster is gradually increase. So our algorithm guarantees the minimization of fleet, although we do not explicitly consider the minimization of the number of vehicles as an objective, which is practically important. Furthermore, several cases are created and the test results are discussed in the final part of the paper.

2 MATHEMATICAL FORMULATION

The character of exclusive multi-vehicle vehicle PDPTW problem discussed in this paper is described as follows:

(1) The vehicles are homogeneous capacitated, and each vehicle starts at the depot and returns to the depot after serving all the clients on the route it visits.

(2) Each client can be visited only once by one of the vehicles, and each vehicle is required to go directly from the pick-up location to the delivery location of the same client.

(3) The vehicles should arrive at the clients within the time window, if they arrive earlier than the earliest arrival time, waiting occurs.

(4) Some data is predefined, including the cost and traveling time between nodes of the clients, time windows of demands, maximum travel time of each vehicle, etc.

Considering that the total cost for loading travel is fixed and can be ignored, the objective is to find a set of vehicle routes to minimizing the total cost for unloading travel, waiting, and delay of the vehicles, subject to the above constraints.

Notation

N set of actual clients

* This work is supported by National Nature Science Foundation under Grant No. 50175064.

V set of extensive clients: $V=\{0\}\bigcup N$, which define the initial depot 0 to be a virtual client

M set of vehicles

n the total number of clients

t_{ij} travel time between the drop-off location of client i and the pick-up location of client j, $i,j\in V$

τ_i travel time between the pick-up location and the drop-off location of client i, $i\in N$

c_{ij} cost for traveling from the drop-off location of client i to the pick-up location of client j, $i,j\in V$

ET_i earliest pick-up time of client i, $i\in N$

LT_i latest pick-up time of client i, $i\in N$

c_l delay cost in time units at client i, $i\in N$

c_w waiting cost in time units at client i, $i\in N$

t_i^{kw} waiting time of vehicle k at client i, $i\in N$

t_i^{kl} delay time of vehicle k at client i, $i\in N$

MT maximum total travel time allowed for any vehicle

Decision variables

$$x_{ij}^k=\begin{cases}1, \text{ if vehicle } k \text{ serve client } i \text{ followed } by \text{ client } j\\ 0, \text{ otherwise}\end{cases}$$

$$i\neq j,\quad \forall i,j\in V$$

T_i^k arrival time of vehicle k at client i, $i\in N$

The corresponding mathematical formulation is as follows:

Minimize

$$\sum_{k\in M}\sum_{i\in V}\sum_{j\in V}c_{ij}t_{ij}x_{ij}^k+\sum_{k\in M}\sum_{i\in N}c_w t_i^{kw}+\sum_{k\in M}\sum_{i\in N}c_l t_i^{kl} \tag{1}$$

Subject to

$$\sum_{k\in M}\sum_{j\in N}x_{ij}^k=1,\quad \forall i\in N \tag{2}$$

$$\sum_{j\in N}x_{ij}^k\leqslant 1,\quad \forall k\in M,i\in N \tag{3}$$

$$\sum_{j\in V}x_{ij}^k-\sum_{j\in V}x_{ji}^k=0,\quad \forall k\in M,i\in V \tag{4}$$

$$\sum_{i\in N}x_{0i}^k=1,\quad \forall k\in M \tag{5}$$

$$\sum_{j\in N}x_{j0}^k=1,\quad \forall k\in M \tag{6}$$

$$t_i^{kw}=\max\left\{ET_i-T_i^k,0\right\},\quad \forall k\in M,i\in N \tag{7}$$

$$t_i^{kl}=\max\left\{T_i^k-LT_i,0\right\},\quad \forall k\in M,i\in N \tag{8}$$

$$\max\left\{T_i^k+t_i^{kw}+\tau_i+t_{i0}\right\}\leqslant MT,i\in N,\forall k\in M \tag{9}$$

$$x_{ij}^k=1\Rightarrow T_i^k+t_i^{kw}+\tau_i+t_{ij}\leqslant T_j^k,\quad \forall k\in M,\forall i,j\in N \tag{10}$$

$$x_{ij}^k=\{0,1\},\quad \forall i,j\in V,k\in M \tag{11}$$

The objective function seeks to minimize total cost. Constraints (2) and (3) ensure that each client is visited by exactly one vehicle. Constraints (4) ensure the equation of flow between clients. Constraints (5) and (6) guarantee that the vehicles departure the depot at their starting time and return to the depot at the end of the planning period. Constraints (7) and (8) define the delay time and waiting time at client i, respectively. Restrictions (9) are maximum travel time constraints. Finally, Restrictions (10) and (11) define the nature of the decision variables.

3 SOLUTION: A HYBRID INTELLIGENT ALGORITHM

The hybrid intelligent algorithm proposed in this paper takes advantage of the dimensionality degrading of cluster analysis, self-adaptability of genetic algorithm and neighborhoods searching of tabu algorithm. The main idea is that we decompose the problem into several single-vehicle PDPTW problems at first by using clustering algorithm, and construct initial solution for tabu search by arranging the best solutions of single-vehicle PDPTW problems which gained by using genetic algorithm, at last get the optimal solution by performing tabu search.

3.1 Clustering Analysis

Clustering aims to partition a set of data into some non-overlapping subsets according to the similarity of statistical samples so that the objects in one subset are similar with each other whereas different from ones in other sets. According to the characteristic, we divide the clients into K clusters by using K-means and each cluster is performed by a vehicle.

In many practical applications, the primary objective is to minimize the total number of vehicles required, especially when it represents the actual number of drivers to employ. In such case the overall cost must be minimized after the minimum number of required vehicles or clusters has been computed.

According to the relation between the maximal value of driving hours, average unloading travel hours of vehicles and total service time, the minimum number of clusters $K_{\min}$ can be obtained from the following equation:

$$K_{\min}=\left[\sum_{i\in N}\tau_i\Big/MT-\overline{t}_{ij}\right] \tag{12}$$

Where $\overline{t}_{ij}=\sum_{i\in V}\sum_{\substack{j\in V\\ j\neq i}}t_{ij}\Big/(n+1)$,

$[\cdot]$ denotes the integer.

The K-means clustering algorithm we used is described as follows:

Step 1 Initial the number of clusters: K=$K_{\min}$, $K_{\min}$ can be obtained from (12).

Step 2 Select k clients x^i, i=1, 2, …, n randomly, and define y^i=x^i, i=1, 2, …, k to be the initial centers of K clusters H_1, H_2, …, H_k

Step 3 Assign all clients to the k clusters according to following criterion:

$$x^i\in H_r,\quad \text{if } D(x^i,y^r)=\min D(x^j,y^r)$$

$$x^i\in i,j\in N,\ r=1,2,\cdots,K$$

Where D is Euclidean distance.

Step 4 After every point is assigned, recount the new

clustering center $y^r = \frac{\sum_{x^i \in Hr} x^i}{n_r}$, $r = 1,2,\cdots,K$, where n_r is the total number of clients in H_j.

Step 5 If not satisfy the terminate condition, go to Step3. Otherwise terminate and output the cluster of clients.

The terminate condition is that the total distance $P(x)$ between clients should be the smallest and the total travel time to serve any cluster of clients should not be longer than the maximum time MT, which are described as follows:

$$P(x) = \min \sum_{r=1}^{K} \sum_{\substack{i,j \in H_r \\ i \neq j}} D^2\left(x^i, x^j\right) \tag{13}$$

$$\sum_{i \in H_r} \tau_i + \overline{t}_r * \left(\left|H_r\right| + 1\right) \leqslant MT \tag{14}$$

$$\max\left(LT_i + \tau_i + t_{i0}\right) \leqslant MT \tag{15}$$

3.2 GA for Single Vehicle PDPTW Problem

In view of the exclusive character of tasks, the pick-up location and delivery location of a client can be integrated into a node. Therefore, the sing le-vehicle PDPTW problem is essentially a traveling salesman problem (TSP) with windows.

Genetic algorithm (GA) is a search technique based on mechanics of natural selection and natural genetics, and has proved to be highly successful in solving optimization problems. In this paper, GA is adopted to solve single vehicle PDPTW problem.

3.2.1 Chromosome Representation and Creation of Initial Population

Natural number code is adopted which denotes the corresponding client number. Thus the chromosome is represented as a string of client numbers. Defining P_i and D_i to be the pick-up location and delivery location respectively, and the set (P_i, D_i) closely follows client number in chromosome, the sequence of the genes in the chromosome is the order of visiting these clients. For example, one chromosome is 0—3(5, 1)—4(1, 3)—2(6, 10)—0, where 0 stands for the depot.

Taking diversity of genes into consideration, Solomon insertion algorithm[8] is used to create the initial population. Two practical solutions are put into the initial population, which are obtained by the farthest insertion method and the closest insertion method respectively. Different from Solomon's random selection of clients, the client farthest from the depot takes precedence in the farthest insertion method to get better initial solutions, while the client nearest from the depot would be inserted before others in the closest insertion method.

3.2.2 Selection

The selection process determines which chromosomes in the current generation participate in reproduction for the next generation population, according to their fitness values in the current population. We adopt a tournament selection mechanism in lecture [2].

In tournament selection, the population P of size N is arbitrarily ranked at first, then each pair of adjacent chromosomes in the ranked population are compared, the one with smaller fitness value is duplicated for the next generation. After comparing all the pairs in P, we get $N/2$ chromosomes. Arbitrarily ranking the population P again and repeating the process, we get other $N/2$ chromosomes. Subsequently, totally there are N chromosomes chosen. By tournament selection operation, the superior chromosomes are given priority in reproduction and the average have some chance to be selected too.

3.2.3 Crossover

The crossover operation is used to exchange information between randomly selected parent chromosomes for producing better offspring. Considering the importance of geographic locations and time sequence, heuristic crossover and merge crossover operation proposed in literature [2] are adopted here.

The heuristic crossover will make a cut on two parents randomly, and from the node after the cut point, choose the shorter gene between the two edges leaving the node. The process is continued until all nodes in the chromosomes have been considered. For example, consider two chromosomes from a 9 client problem:

Parent1: 3-6-5-9-1-2-4-8-7

Parent2: 8-7-4-1-5-6-3-9-2

Assume we choose 4 to be the first gene of the offspring, we have to first swap 4 and 3 in parent 2.After swapping, if the distance between 4 and 8 is shorter than that between 4 and 9, we then choose 8 to be the next node and swap 8 and 9 in parent1 to avoid reproduction later.

The merge crossover process is similar to heuristic crossover, except that merge crossover operated on basis of the predefined time precedence. This time precedence is defined according to the latest arrival time of each node.

In the same example above, a random cut point is selected and a first gene 4 is chosen randomly from parent1 and swapping 4 and 3 is done to parent2. The node 8, which comes earlier in the time precedence becomes the next gene in the new chromosome, and is deleted in parent1 to avoid duplication later.

From a pair of parents, both heuristic crossover and merge crossover will produce one offspring. So that we can get two children from two parent.

3.2.4 Mutation

The mutation operator is used to add variability to the population obtained from the above crossover process. There is several mutation operators proposed in the literatures. Because of the fixed length of chromosomes for single-vehicle PDPTW, only swap point and swap sequence mutation are used here.

The swap point mutation will select two genes randomly, and their positions are interchanged. For example, from the chromosome L we select two points, 5 and 4. Interchanging 5 and 4, we obtained the offspring L'.

L: 3-6-5-9-1-2-4-8-7

L': 3-6-4-9-1-2-5-8-7

The swap sequence mutation will choose two cut sites within a chromosome at random, and then the order of the substring between the two positions is inverted. For example, from the chromosome L we select two points, 5 and 4. By swap sequence mutation, we obtained the offspring L'.

L: 3-6-5-9-1-2-4-8-7

L': 3-6-8-7-5-9-1-2-4

Crossover ratio p_c and mutation ratio p_m follow Adaptive Genetic Algorithm put forward by M.Srinivas, which increase when the adaptability of gene populations is comparatively concentrated and decrease when the adaptability is comparatively dispersive.

$$p_c = \begin{cases} K_1 \dfrac{(f_{\max} - f')}{(f_{\max} - f_{\text{mean}})}, & f' \geqslant f_{\text{mean}} \\ K_2, & f' < f_{\text{mean}} \end{cases} \tag{16}$$

$$p_m = \begin{cases} K_3 \dfrac{(f_{\max} - f)}{(f_{\max} - f_{\text{mean}})}, & f \geqslant f_{\text{mean}} \\ K_4, & f < f_{\text{mean}} \end{cases} \tag{17}$$

Where K_i(i=1, 2, 3, 4)is constant below 0; $f_{\max}$ is the maximal fitness value, f_{mean} is the average fitness value; f' is the better fitness value of the two offspring brought.

3.3 Tabu Search

Tabu search is a procedure that uses an initial solution as a starting basis for seeking improved solutions by searching different neighborhoods. We define the movement as neighborhoods that a client is moved from one route and included in another route. We use SPI in literature [9] for the tabu search. A quick tabu search algorithm in literature [10] is also used in this paper, which defines a cost function.

If a client is removed from route r_1 and inserted into route r_2, there will generate two routes r_1' and r_2' .Defining f(r) to be the fitness value of route r, the cost function of SPI operation for one time is defined as follows:

$$C_{SPI} = f(r_1') + f(r_2') - f(r_1) - f(r_2)$$

3.4 General Structure of the Hybrid Intelligent Algorithm

The cluster number obtained from (12) is not always optimal because of time windows neglect, so it is necessary to increase the cluster number gradually and compare the optimal solutions with each other, until the fitness value of optimal solution decrease no longer. The algorithm procedure is as follows:

Step 1 Initialize the cluster number k from (12), define the maximum times h for tabu search, let the fitness value of solution S_{k-1} to be ∞, i.e. $f(S_{k-1}) = \infty$.

Step 2 Assign the clients to cluster k by using the cluster algorithm, generate the initial solution S_k which consists of the best solutions of k routes obtained by running the genetic algorithm individually.

Step 3 Choose the best SPI operations for tabu or non-tabu, which lead to the minimize value of C_{SPI}.

Step 4 If the best SPI operation for tabu leads to $C_{\text{SPI}} \leqslant 0$, execute the SPI operation, then go to Step 3

Step 5 If the best SPI operation for non-tabu leads to $C_{\text{SPI}} < 0$, the non-tabu status is overridden. Then execute the SPI operation, go to Step 3

Step 6 If the best SPI operation for non-tabu led to $C_{\text{SPI}} \geqslant 0$, optimize each route by using genetic algorithm and gain the best solution S_k' . Let $S_k = S_k'$, if $f(S')_k \leqslant f(S_k)$. Then go to Step 3.

Step 7 If the best SPI operation for either tabu or non-tabu exists, execute it and go to Step 3.

Step 8 Repeat Steps 3–7 for h times.

Step 9 Let k=k+1, repeat Steps 2–8 until $f(S_k) > f(S_{k-1})$.

Step10 Output the global optimal solution S_{k-1}.

4 COMPUTIONAL RESULTS

Several cases are created to test the performance of the algorithm proposed in this paper. All experiments are performed on a Pentium 4 personal computer with 2.0 G MHz and the Windows 2000 environment. The programming language is Borland C++. Geographical data is based on the electronic map of Jinan city, and the distance between any two nodes are predetermined by Dijkstra algorithm. The maximum travel time is 240 minutes. The travel cost, delay cost and waiting cost in time units are 1.0, 2.0 and 0.1, respectively.

Cases 1 and Case 2 are created to test the algorithm performance on different conditions. The total number of clients n is constant, while total number of geographical nodes m is different. For Case 1, the geographical distributing of client is sparse and the time window constrains is not regular. While the geographical distributing is dense and the time windows are loose for most clients and close for a few in Case 2. Tab.1 shows the test results.

Tab.1 The Test Results of Case1 and Case2

	n, m	K	C	execution-time
Case 1	50, 101	5	278	14s
Case 2	50, 10	4	63	22s

K: Total number of required vehicles.

C: Total cost (the objective value of optimal solution).

From tab.1, we see that the hybrid intelligent algorithm is effective and can generate the optimal solution in an acceptable short time. The difference in the number of vehicles results from the difference in geographical distributing and time windows constraints of clients. The reason why the total cost of Case 1 is larger than that of Case 2 is that more vehicles required in Case 1 led to more unloading cost.

Furthermore, to explain the advantage of the hybrid algorithm, we compare it with the heuristic algorithm in literature [2] on condition of different client scale. Each instance has been tested for 20 times, and Tab.2 shows the test results.

Tab.2 Comparison of Test Result from Different Methods

N	hybrid algorithm		heuristic algorithm	
	C	T_A/ms	C	T_A/ms
20	58	4 952	58	4 291
50	261	16 237	274	20 371
100	354	48 975	371	49 121
200	491	108 563	522	154 153
500	689	323 741	721	576 984
1000	937	755 741	987	1341 646

T_A: Average execution-time.
C: Average total cost (the objective value of optimal solution).

On one hand, in terms of execution-time, we can see from table 2 that the time of both algorithms is almost equivalent for small-scale clients, and very different from each other for large-scale clients. The execution-time of heuristic algorithm increased much faster than that of the proposed hybrid algorithm. On the other hand, in terms of the total cost, we can see from Tab. 2 that the total cost of the hybrid algorithm is better than that of the heuristic algorithm at the same client scale.

The difference between two algorithms mainly result from that the hybrid algorithm can greatly degrade the dimensionality of search space by cluster algorithm based on the minimal fleet, and effectively find the optimal solution by using genetic algorithm and tabu search.

So the comparison results show the proposed method's comprehensive superiority, and indicate that the hybrid intelligent algorithm is efficient and quick in solving PDPTW problem.

5 CONCLUSION

As a special vehicle routing and scheduling problem, the optimization of PDPTW is an important piece of logistical system. In this paper a hybrid intelligent algorithm is proposed to solve PDPTW, which combine cluster algorithm, genetic algorithm and tabu algorithm. At the same time, the minimization of fleet is obtained by using the gradually increase strategy of the number of cluster from the minimal feasible number. The computational results demonstrate the effectiveness of the proposed method to solve the PDPTW problem.

REFERENCES

[1] Fermín AlfredoTang Montané, Roberto Diéguez Galvao. A tabu search algorithm for the vehicle routing problem with simultaneous pick-up and delivery service[J]. Computers &Operations Research, 2006, 33: 595-619.

[2] Tan K C, Lee L H, Zhu Q L, Ou K. Heuristic methods for vehicle routing problems with time windows[J]. Artificial Intelligence in Engineering, 2001, 15: 281-295.

[3] William P Nanry, Wesley J Barnes. Solving the pickup and delivery problem with time windows using reactive abu search [J]. Transportation Research: PartB, 2000, 34: 107-121.

[4] Li H, Lim A. A metaheuristic for the pickup and delivery problem with time windows[A]. Proc of the 13th IEEE Int Conf on Tools with Artificial Intelligience [C]. Dallas: IEEE Computer Society, 2001: 160-167.

[5] Sexton T, Bodin L. Optimizing single vehicle many-to-many operations with desired delivery times: II. Routing[J]. Transportation Science, 1985, 19: 411-435.

[6] Sexton T, Choi Y. Pickup and delivery of partial loads with time windows[J]. American Journal of Mathematics and Management Science, 1986, 6: 369-398.

[7] Van der Bruggen LJJ, Lenstra JK, Schuur PC. Variable-depth search for the single-vehicle pickup and delivery problem with time windows [J]. Transportation Science, 1993, 27: 298-311.

[8] Solomon M. Algorithms for the vehicle routing and scheduling problems with time window constraints [J].Operations Research, 1987, 35(2): 254-265.

[9] LauH C, Liang Z. Pickup and delivery with time windows: Algorithms and test case generation [A]. Proc of the13th IEEE Int Conf on Tools with Artificial Intelligence [C]. Dallas: IEEE Computer Society, 2001: 333-340.

[10] Jia Yongji, Gu Hanyu, XI Yugeng. Analysis and Algorithm of Single-Vehicle Exclusive Pickup and Delivery Problem with Time Windows [J]. Journal of Shanghai Jiaotong university, 2005, 39(3): 409-412. (in Chinese)