Research Article

Vehicle Routing Problem with Time Windows and Simultaneous Delivery and Pick-Up Service Based on MCPSO

Xiaobing Gan, Yan Wang, Shuhai Li, and Ben Niu

College of Management, Shenzhen University, Shenzhen 518060, China

Correspondence should be addressed to Xiaobing Gan, ganxb@szu.edu.cn

This paper considers two additional factors of the widely researched vehicle routing problem with time windows (VRPTW). The two factors, which are very common characteristics in realworld, are uncertain number of vehicles and simultaneous delivery and pick-up service. Using minimization of the total transport costs as the objective of the extension VRPTW, a mathematic model is constructed. To solve the problem, an efficient multiswarm cooperative particle swarm optimization (MCPSO) algorithm is applied. And a new encoding method is proposed for the extension VRPTW. Finally, comparing with genetic algorithm (GA) and particle swarm optimization (PSO) algorithm, the MCPSO algorithm performs best for solving this problem.

1. Introduction

Vehicle routing problem with time windows (VRPTW) is an important issue in logistics system which has been researched widely in recent years. The problem can be described as choosing routes for limited number of vehicles to serve a group of customers in the time windows. Each vehicle has a limited capacity. It starts from the depot and terminates at the depot. Each customer should be served exactly once. The objective of the VRPTW is to minimize the total transport costs. Many researchers have contribute to the problem. In 1981, Schrage [1] proposed vehicle routing and scheduling problem with time window constraints as an important area for progress in handling realistic complications and generalizations of the basic routing model. Feng [2] divide the VRPTW with delivery and pickup into five categories. Wang and Lang [3] proposes multiperiod vehicle routing problem with recurring dynamic time windows.

However, in reality, two factors need to be considered, the uncertain number of vehicles and simultaneous delivery and pick-up. This situation is very common in transport activity, because reduction of the number of the vehicles can save costs, while simultaneous delivery and pick-up can improve transport efficient. VRPTW with simultaneous delivery and pick-up service (VRPTW-SDP) is an extension of VRPTW, where customers require simultaneous delivery and pick-up.

Many researchers have made great effort in solving the VRPTW in recent years. Fisher [4] proposed k-trees to solve the VRP and VRPTW. Kolen et al. [5] described a branch-and-bound method. However, those classical approaches less efficient in solving complex problems. Thus evolutionary algorithm (EA) is proposed to deal with optimization problems. Among those algorithms, particle swarm optimization (PSO) has turned out to be an efficient algorithm in dealing with many complex optimization problems. While PSO also has some disadvantages. It sometimes immerses the local optimal value, thus the accuracy is limited. Then the multiswarm cooperative particle swarm optimization (MCPSO) was proposed as an improved PSO in [6]. It takes a multi-swarm cooperative evolutionary strategy where the master swarms change their particles based on their own knowledge and the knowledge of the particles in the slave swarms, while the slave swarms carry out PSO independently.

In this paper, we study a case of the VRPTW both with uncertain number of vehicles and simultaneous delivery and pickup service. The objective of the proposed VRPTW-SDP is minimizing the transport costs. In order to solve the VRPTW-SDP, we set the objective of VRPTW-SDP as the fitness function of MCPSO. In MCPSO, each particle should contain two aspects of the customers, the order of served by which vehicle. Thus two dimensions encoding methods in MCPSO are proposed and used for the VRPTW-SDP.

The rest of this paper is arranged as follows. Section 2 describes the VRPTW-SDP, Section 3 describes the MCPSO for the proposed problem, Section 4 presents experiment study, and Section 5 of the paper contains the conclusion.

2. Description of VRPTW-SDP

Vehicle routing problem with time windows (VRPTW) can be defined as choosing routes for limited number of vehicles to serve a group of customers in the time windows. Each vehicle has a limited capacity. It starts from the depot and terminates at the depot. Each customer should be served exactly once. If vehicles arrive before the time window "opens" or after the time window "closes," there will be waiting cost and late cost. Assume that there are K vehicles in depot 0. N customers are waiting to be served and each of the customers has a demands of g_i $(i = 1, 2, \ldots, N)$ units. The distance between customer i and j is d_{ij}. Each vehicle has a capacity of q_k $(k = 1, 2, \ldots, K)$ units. That is, the total demands of customers served by each vehicle cannot exceed q_k units. Therefore, the vehicle has to periodically returned to the depot for reloading or a new vehicle needed to be arranged for delivery. Besides, each customer must be visited once and only once by exactly one vehicle. s_i represents service time needed by customer i. Therefore, the vehicle has to stay at the location of customer i for a time interval at least s_i $(s_0 = 0$ is associated with the depot 0) for service. A time window $[ET_i, LT_i]$ is considered. Therefore, if a vehicle arrives at customer i before ET_i, it has to wait until the beginning of the time window to serve the customer. Thus there is a cost e for waiting. On the other hand, if a vehicle cannot arrive at i before LT_i, there will be a cost f for late. The velocity of each vehicle is v_k. t_i represents the moment when the vehicle arrives i from the depot. And the unit freight of each vehicle is C_k. This paper considers the

condition that vehicles of the depot are the same, that is, the vehicles have the same velocity v, the same capacity q and the same unit freight C.

Define variable

$$x_{ijk} = \begin{cases} 1 & \text{if the vehicle } k \text{ travels from } i \text{ to } j, \\ 0 & \text{else.} \end{cases} \tag{2.1}$$

The goal of VRPTW is

$$\min z = \sum_{i=0}^{N}\sum_{j=0}^{N}\sum_{k=1}^{K} C * x_{ijk} * d_{ij} + \sum_{i=1}^{N} \max\{e * (ET_i - t_i); 0; f * (t_i - LT_i)\}, \tag{2.2}$$

where

$$t_{ijk} = \sum x_{ijk}\left(t_i + \frac{d_{ij}}{v} + s_i\right) \quad (t_0 = 0, \; s_0 = 0), \tag{2.3}$$

s.t.

$$\sum_{j=1}^{N}\sum_{k=1}^{K} x_{jik} = \sum_{j=1}^{N}\sum_{k=1}^{K} x_{ijk} = K \quad (i = 0), \tag{2.4}$$

$$\sum_{j=0}^{N}\sum_{k=1}^{K} x_{ijk} = 1 \quad (i \in N), \tag{2.5}$$

$$\sum_{i=0}^{N}\sum_{k=1}^{K} x_{ijk} = 1 \quad (i \in N), \tag{2.6}$$

$$\sum_{j=1}^{N} x_{ijk} = \sum_{j=1}^{N} x_{jik} = 1 \quad (i = 0, \; k \in K), \tag{2.7}$$

$$\sum_{i=0}^{N}\sum_{j=0}^{N} x_{ijk} * g_i \leq q. \tag{2.8}$$

In the model, formula $\sum_{i=1}^{N} \max\{e * (ET_i - t_i); 0; f * (t_i - LT_i)\}$ defines the time window constraint, where t_i is the time when the vehicle arrives at customer i, $ET_i - t_i$ is the waiting time of a vehicle at the customer i. From formula (2.3), s_i is the service time, and t_{ijk} is the traveling time of vehicle k between customer i and j. When $i = 0$, the vehicle is at the depot, $s_0 = t_0 = 0$.

Constraint (2.4) represents the number of the vehicles which start from the depot and go back to depot is k. Constraint (2.5) and constraint (2.6) stand for each customer can only be served by one vehicle. Constraint (2.7) means all the vehicles which start from the depot go back to the depot. Constraint (2.8) denotes the quantity of goods that each vehicle carries could not exceed the capacity q.

However, two extra aspects are involved. Firstly, what if the number of the vehicles is uncertain? Zhang et al. [7] proposed that the number of the vehicles can be calculate by $\sum q_i/Q$, where q_i is the demands of the customer i, Q is the vehicle's capacity. This is quite simple, but it may not meet the requirement of the time window. When the number of the vehicles is uncertain, it is not necessary to use all of the k vehicles. The reduction of the vehicles may contribute to reducing the total costs. Consider the constraint (2.4), where k can be replaced by m ($m \leq k$). Secondly, simultaneous delivery and pickup is involved to the vehicles. Under this condition, two variables are related to the customer, demands (delivery) g_i and supplies (pickup) p_i.

Define parameters: C is unit freight of the vehicle, K is the total number of the vehicles, N is the total number of the customer, d_{ij} represents the distance between customer i and j, v is the velocity of each vehicle, q_i is the demands of customer i, p_i is the supplies of customer i, s_i is the service time of customer i, $[ET_i, LT_i]$ represents the time window, e is the cost for waiting, f is the cost for late, q is the specified capacity of the vehicle, t_i is the time when the vehicle arrives customer i, y_{ijk} represents the capacity of the vehicle k when it travels from customer i to j, z_{ijk} represents the pick-ups of the vehicle k from customer i when it travels from customer i to j,

$$x_{ijk} = \begin{cases} 1 & \text{if the vehicle } k \text{ travels from } i \text{ to } j, \\ 0 & \text{else.} \end{cases} \tag{2.9}$$

When the vehicle starts from the depot, its total cargo quantity should be equal to the sum of all the customers' demands, who are on the route of the vehicle travels along, so

$$\sum_{j=1}^{N} y_{ojk} = \sum_{i=0}^{N} \sum_{j=0}^{N} x_{ijk} * g_i \quad (k \in K, \ i \neq j). \tag{2.10}$$

While, when the vehicle k returns to the depot, the total pick-ups quantities are the sum of each customer's supplies. That is

$$\sum_{i=1}^{N} z_{i0k} = \sum_{i=0}^{N} \sum_{j=0}^{N} x_{ijk} * p_i \quad (k \in K, \ i \neq j). \tag{2.11}$$

Then the model is as follows:

$$\min z \sum_{i=0}^{N} \sum_{j=0}^{N} \sum_{k=1}^{K} C * x_{ijk} \sum_{i=1}^{N} \max\{e * (a_i - t_i); \ 0; f * (t_i - b_i)\} + m * c, \tag{2.12}$$

s.t.

$$t_{ijk} = x_{ijk}\left(t_i + \frac{d_{ij}}{v}\right), \tag{2.13}$$

$$\sum_{j=1}^{N} x_{ijk} = \sum_{j=1}^{N} x_{jik} \le 1, \tag{2.14}$$

$$\sum_{i=0}^{N} \sum_{k=1}^{K} x_{ijk} = 1, \tag{2.15}$$

$$\sum_{j=0}^{N} \sum_{k=1}^{K} x_{ijk} = 1, \tag{2.16}$$

$$\sum_{j=1}^{N} y_{ojk} = \sum_{i=0}^{N} \sum_{j=0}^{N} x_{ijk} * g_i \quad (k \in K, \ i \ne j), \tag{2.17}$$

$$\sum_{i=1}^{N} z_{i0k} = \sum_{i=0}^{N} \sum_{j=0}^{N} x_{ijk} * p_i \quad (k \in K, \ i \ne j), \tag{2.18}$$

$$\sum_{j=1}^{N} y_{0jk} \le q, \tag{2.19}$$

$$\sum_{j=0}^{N} z_{0jk} \le 1, \tag{2.20}$$

$$\sum_{i=0}^{N} y_{ijk} + (p_i - q_i) \sum_{i=0}^{N} x_{ijk} = \sum_{i=1}^{N} y_{ijk}, \tag{2.21}$$

$$\sum_{j=0}^{N} \sum_{k=1}^{K} x_{0jk} = \sum_{j=0}^{N} \sum_{k=1}^{K} x_{j0k} = m \le K, \tag{2.22}$$

$$y_{ijk} \le q * x_{ijk}, \tag{2.23}$$

$$y_{ijk} > 0, \tag{2.24}$$

$$z_{ijk} > 0. \tag{2.25}$$

In the model, formula (2.12) is the goal of the problem, that is minimum the cost. Formula (2.13) calculates the time when the vehicle k travels from customer i to j. Constrain (2.14) means that all the vehicles start from the depot and terminate at the depot. Constrain (2.15) and (2.16) represent that each customer can only be served by one vehicle. Constrain (2.17) and (2.19) stand for the capacity of each vehicle cannot exceed the specified capacity. Constrain (2.18) and (2.20) stand for the sum pickups of each vehicle cannot exceed the specified capacity. Constrain (2.21) represents the capacity of the vehicle after it has serviced
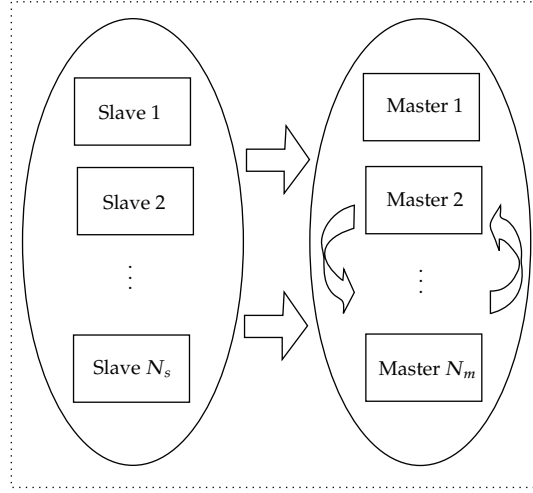
Figure 1: Master-slave model.

the customer i, but before it services customer j. Constrain (2.22) means the number of the vehicles arranged should not exceed k. Constrain (2.23) represents that no matter where the vehicle is, the capacity cannot exceed the vehicle's specified capacity. Constrain (2.24) and (2.25) mean that the capacity and pickups should not lower 0.

3. MCPSO Algorithm for VRPTW-SDP

3.1. Description of MCPSO Algorithm

The particle swarm optimization (PSO) was proposed by Kennedy and Eberhart in 1995. It was inspired by the social behavior of animals, such as bird flocking and fish schooling. The mechanism of PSO is as follows. Each member of PSO is seen as a particle, and each particle is a potential solution to the problem. The particle keeps memory of its previous locations, and evolves from one generation to another. The experiences of all the particles are transmitted to the global best particle. By using the communication mechanism, PSO optimum search is speeded up.

However, PSO also has disadvantages. It sometimes converges to undesired local solution, thus the accuracy of the algorithm can achieve is limited. MultiSwarm Cooperative Particle Swarm Optimizer (MCPSO) proposed in [4] is a variant of PSO. It takes a multiswarm cooperative evolutionary strategy where the master swarms change their particles based on their own knowledge and the knowledge of the particles in the slave swarms, while the slave swarms carry out PSO independently to get the diversity of particles. The mechanism of information transfer for MCPSO is showed in Figure 1 (the arrows stand for information transfer).

Suppose that there are N swarms, among which there are N_m ($N_m \in [1, N]$) main swarms, the remains are slave swarms N_s ($N_m + N_s = N$). Each slave swarm evolves according to the PSO. When all the slave swarms are ready with the next new generation, each slave swarm then sends the best local individual to one of the master swarms. The master

```
Algorithm MCPSO
Begin
  Initialize the population // master swarm and slave swarm
  Evaluate the fitness value of each particle in the population
  Repeat
    Do in parallel
      Swarm i, 1 ≤ i ≤ Ns
    End Do in parallel
    Barrier synchronization // wait for all processes to finish
    Select the fittest global individual (pᵍˢ or pᵍᴹ) from all the swarms
    Do in parallel
      Swarm j, 1 ≤ j ≤ Nm
    End Do in parallel
    Barrier synchronization
    Evolve the mast swarm // update the velocity and position using Eqs. (3.1)
    Evaluate the fitness value of each particle
  Until a terminate-condition is met
End
```

Algorithm 1: Pseudocode for MCPSO algorithm.

swarm chooses the best of all received individuals and other main swarms' experiences and evolves according to the following equations:

$$
v_{id}^{M}(t+1) = w v_{id}^{M}(t) + c_1 * r_1 * \left(p_{id}^{M}(t) - x_{id}^{M}(t) \right) + c_2 * r_2 * \left(p_g^{M}(t) - x_{id}^{M}(t) \right)
$$
$$
+ c_3 * r_3 * \left(p_g^{Q}(t) - x_{id}^{M}(t) \right),
$$
$$
x_{id}^{M}(t+1) = x_{id}^{M}(t) + v_{id}^{M}(t+1),
$$
(3.1)

where M represents one of the master swarm, Q represents the other master swarms, c_1 and c_2 are accelerating constant, w is inertial weight, c_3 is migration coefficient, while r_1, r_2, and r_3 are uniform random sequences in the range $[0, 1]$. Note that the particle's velocity update in the master swarm is associated with three factors

$p_{id}^{M}(t)$: previous best position of the master swarm in dth;

$p_g^{M}(t)$: best global of the master swarm in dth;

$p_g^{Q}(t)$: best global of the other master swarms in dth.

In what follows we briefly sketch out the multiswarm cooperative particle swarm optimizer (MCPSO) step by step. See Algorithm 1.

3.2. Particle Encoding Scheme

For our proposed vehicle route problem with time windows, the appropriate expression of particles in MCPSO algorithm is a key issue. After all, the essence of VRPTW is to determine the best route of vehicle from a series of potential solutions. The object is to minimize the

Table 1: The supplies and demands and time window of each customer.

Customer i	1	2	3	4	5	6	7	8
Demands g_i	0.8	1.5	0.8	0.7	1.4	1.5	0.6	0.8
Supplies p_i	0.6	1	1	1	0.8	1.2	0.8	0.8
Time window	$[1,3]$	$[2,4]$	$[1,2]$	$[3.5,7]$	$[3.5,5]$	$[2,5]$	$[2,5]$	$[1,4.5]$

sum of all costs. Based on the uncertain number of vehicles in this paper, for the VRPTW with N demand points, $2N$ dimensions encoding scheme is applied. For each demand point, two dimensions are involved. The first k is the serial number of the vehicle that serviced the demand point. The second r is the serial of the demand point that was serviced by k. Conveniently, each particle corresponds to a matrix of $(2, N)$. The first dimension represents the serial number of vehicles, and the second dimension represents the serial number of customers.

3.3. Design of Fitness Function

In this paper, the capacity of vehicles is calculated by

$$\sum_{i=0}^{N} y_{ijk} + (p_i - q_i) \sum_{i=0}^{N} x_{ijk} = \sum_{i=0}^{N} y_{jik} \quad (k \in K, \ i \neq j). \tag{3.2}$$

Both sides of the equation have unknown quantities. It is complicated to convert the constraint by penalty function method. Hence the constraint is dealt with when we program for the VRPTW-SDP. Thus the objective function corresponds to the fitness function:

$$\text{fitness} = \min z. \tag{3.3}$$

4. Experimental Study

We consider the following situatoion. 8 customers need served by one depot. The depot has 5 vehicles. Each of the vehicles has a capacity of 3 units and velocity of 10 units. The distance between each customer and the demands supplies of each customer is shown in Table 1. The distance between depot and each customer is shown in Table 2. The service time for each customer is 0.

Base on Matlab 7.0, we apply MCPSO algorithm to solve this VRPTW-SDP. In MCPSO algorithm, the number of swarms is $n = 4$. The 4 swarms conclude 3 slave swarms and 1 master swarm. The migration coefficient of the master swarm is $c_1 = c_2 = c_3 = 1.333$. The migration coefficient of the slave swarm is $c_1 = c_2 = 2$. To validate its performance, the MCPSO algorithm is also compared with GA and PSO.

For fair comparison, in all cases the population size num of GA/PSO and MCPSO was set at 100 (all the swarms of MCPSO include the same particles) and a fixed number of maximum iterations 100 is set in GA/PSO and MCPSO. In the fitness function, we set four parameters as follows: $a = 1$, $b = 5$, $s_1 = 1$, $s_2 = 2$. The results are shown in Tables 3 and 4. To serve the 8 customers, 3 vehicles are arranged. The first vehicle serves customer 2, 8 and 4 in sequence. The second vehicle serves customer 3 and 6 in turn. The last one serves

Table 2: The distance between depot and each customer.

d_{ij}	0	1	2	3	4	5	6	7	8
0	0	10	9	7	8	8	8	3	4
1		0	4	9	14	18	18	13	14
2			0	5	10	14	17	12	13
3				0	5	9	15	10	11
4					0	6	13	11	12
5						0	7	10	12
6							0	6	8
7								0	2
8									0

Table 3: The route and fees.

Vehicle	Route of vehicle	Fee
1	$0 \rightarrow 2 \rightarrow 8 \rightarrow 4 \rightarrow 0$	
2	$0 \rightarrow 3 \rightarrow 6 \rightarrow 0$	549
3	$0 \rightarrow 1 \rightarrow 5 \rightarrow 7 \rightarrow 0$	

Table 4: The time the vehicle arrives at each customer.

Customer	1	2	3	4	5	6	7	8
Time	1	0.9	0.7	3.4	2.8	2.2	3.8	2.2

Table 5: Summary result.

Algorithm	Maximum	Minimum	Mean	Variance
PSO	642	522	571.2	1296
GA	708	631	676.4	557.6
MCPSO	604	500	551.5	456.5

customer 1, 5, and 7 successively. The total fees are 549 units. From Table 4, we can see that the time points when vehicles arrive at customers are within the time windows. Figure 2 shows MCPSO algorithm, which converge on the minimum point quickly.

The ten fitness values are obtained in ten runtimes in GA algorithm, PSO algorithm and MCPSO algorithm. The final result including maximum, minimum, mean, and variance is shown in Table 5. It is apparently that the maximum, minimum and mean of MCPSO are smaller than PSO, whose are smaller than GA. Although the difference on maximum, minimum, and mean between PSO and MCPSO are small, the gap on variances is big. This means that the MCPSO is more stability. The convergence graph for MCPSO, PSO and GA can be seen in Figure 3. MCPSO achieves a minimal solution 549, which is lower than 570 (PSO) and 675 (GA). It is obvious that MCPSO is more suitable at least for our proposed vehicle route problem with time windows.
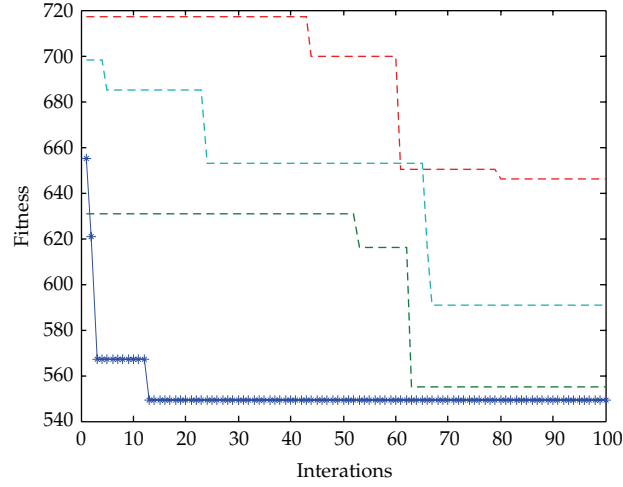
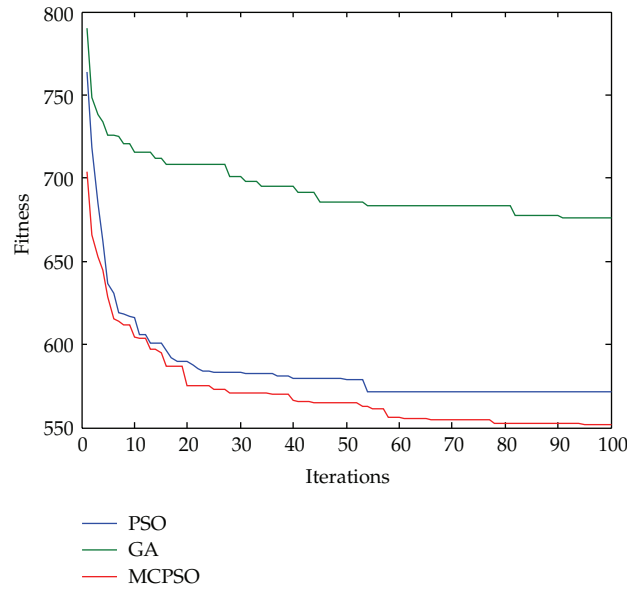**Figure 2:** The Convergence graph for MCPSO.

Figure 3: The convergence for MCPSO, PSO, and GA.

5. Conclusions

The VRPTW-SDP studied in this paper is an extension of VRPTW, where the number of vehicles with simultaneous delivery and pickup is uncertain. The proposed problem is meaningful for transport companies in real world. To solve the proposed problem, we used MCPSO algorithm. A new encoding method is proposed, which may be somewhat benefit to deal with other problems. Furthermore, with comparison to PSO algorithm and GA algorithm, the results show that the simple and robust MCPSO algorithm is more effective for the proposed VRPTW-SDP.

Acknowledgments

References

[1] L. Schrage, "Formulation and structure of more complex/realistic routing and scheduling problems," *Networks*, vol. 11, no. 2, pp. 229–232, 1981.

[2] Z. Feng, "Multi-period vehicle routing problem with recurring dynamic time windows," in *Proceedings of the 8th International Conference on Digtial Object Identifier*, pp. 1–6, 2011.

[3] Y. Wang and M. Lang, "Study on the model and tabu search algorithm for delivery and pickup vehicle routing problem with time windows," in *Proceedings of the IEEE International Conference on Service Operations and Logistics, and Informatics (SOLI '08)*, pp. 1464–1469, October 2008.

[4] M. L. Fisher, "Optimal solution of vehicle routing problems using minimum K-trees," *Operations Research*, vol. 42, no. 4, pp. 626–642, 1994.

[5] A. W. J. Kolen, A. H. G. R. Kan, and H. W. J. M. Trienekens, "Vehicle routing with time windows," *Operations Research*, vol. 35, no. 2, pp. 266–273, 1987.

[6] B. Niu, Y. Zhu, X. He, and H. Wu, "MCPSO: a multi-swarm cooperative particle swarm optimizer," *Applied Mathematics and Computation*, vol. 185, no. 2, pp. 1050–1062, 2007.

[7] T. Zhang, Y. J. Zhang, and M. G. Wang, "Model and hybrid algorithm for vehicle routing problem with uncertain vehicle number," *System Engineering—Theory Methodology Applications*, vol. 11, no. 2, pp. 121–130, 2010.