

Research Article

Dynamic Vehicle Routing Using an Improved Variable Neighborhood Search Algorithm

Yingcheng Xu,^{1,2} Li Wang,³ and Yuexiang Yang¹

¹Branch of Quality Management, China National Institute of Standardization, Beijing 100088, China

²Department of Industrial Engineering, Tsinghua University, Beijing 100084, China

³School of Economics and Management, Beihang University, Beijing 100191, China

Correspondence should be addressed to Yuexiang Yang; yxyangcnis@163.com

Received 14 September 2012; Revised 5 December 2012; Accepted 25 December 2012

Academic Editor: QiuHong Zhao

Copyright © 2013 Yingcheng Xu et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

In order to effectively solve the dynamic vehicle routing problem with time windows, the mathematical model is established and an improved variable neighborhood search algorithm is proposed. In the algorithm, allocation customers and planning routes for the initial solution are completed by the clustering method. Hybrid operators of insert and exchange are used to achieve the shaking process, the later optimization process is presented to improve the solution space, and the best-improvement strategy is adopted, which make the algorithm can achieve a better balance in the solution quality and running time. The idea of simulated annealing is introduced to take control of the acceptance of new solutions, and the influences of arrival time, distribution of geographical location, and time window range on route selection are analyzed. In the experiment, the proposed algorithm is applied to solve the different sizes' problems of DVRP. Comparing to other algorithms on the results shows that the algorithm is effective and feasible.

1. Introduction

Dynamic Vehicle Routing Problem (DVRP) is a variant of the Vehicle Routing Problems (VRPs) that has arisen due to recent advances in real-time communication and information technologies. The VRP is a nondeterministic polynomial hard (NP-hard) problem that calls for the determination of the optimal set of routes to be performed by a fleet of vehicles to serve a given set of customers. However, a majority of the information is unpredictable before path optimization, such as the customer's geographic position, customer demand, and vehicle travel and service time. This information is dynamic and new information may appear or existing information changes, and so forth. Many different factors must be considered when a decision about the allocation and scheduling of a new request is taken: the current location of each vehicle, their current planned route and schedule, characteristics of the new request, travel times between the service points, characteristics of the underlying road network, service policy of the company, and other related constraints. The DVRP is a complex problem compared to

the classic VRP, and variable neighborhood search (VNS) is proposed as a means to effectively and efficiently tackle the dynamic problem and optimize the planned routes between the occurrences of new events. VNS was initially proposed by Hansen and Mladenović [1, 2] for solving combinatorial and global optimization problems. The main reasoning of this metaheuristic is based on the idea of a systematic change of neighborhoods within a local search method.

This paper has two main contributions. First, according to the characteristic of DVRP, we gave the graph expression and formulated the mathematical model. Second, the proposed algorithm is improved in initial solution, shaking and local search based on classic VNS. The later optimization process is presented, and the results show that the algorithm is feasible and competitive with other existing algorithms. The remainder of the paper is organized as follows: literature reviews are illustrated in Section 2 and the mathematical formulation problem is discussed in Section 3. Section 4 introduces the main ideas of the improved variable neighborhood search, while computational results are presented and discussed in Section 5. Section 6 concludes the paper.

2. Literature Reviews

The DVRP problem is closely related to the actual production and life. In recent years, DVRP gradually becomes a hot topic. Both domestic and foreign scholars mainly focus on the construction of different scheduling models and the designing of simple and efficient heuristic algorithms. A survey of results achieved on the different types of DVRPs can be found in Gendreau and Potvin [3]. The dynamic full-truckload pickup and delivery problem has been studied by Fleischmann et al. [4] and Yang et al. [5]. Montemanni et al. [6] proposed the vehicle scheduling model with random dynamic demand and solved it using an ant colony algorithm. In Gendreau et al. [7] neighborhood search heuristics for dial-a-ride problems are finally presented. Goel and Gruhn [8] studied the mathematical model of dynamic vehicle routing problems under the conditions of the real-life information. Schönberger and Kopfer [9] studied the real-time decision-making and autonomous decision-making of DVRP. Novoa and Storer [10] introduced the approximate solution algorithm with stochastic demands. Branchini et al. [11] presented local search algorithm for a dynamic vehicle routing problem. Müller [12] studied DVRP with time window and analyzed the algorithm for the model suboptimal solution.

In the past ten decades, a tremendous amount of work in the field of vehicle routing problems has been published, especially literature based on VNS. The Bräysy [13] gave the internal design of the VND and RVNS algorithm in detail, analyzed the VRPTW problem, and indicated the VND algorithm as one of the most effective ways to solve VRPTW problems. Polacek et al. [14] designed VNS to solve the multidepot vehicle routing problem with time windows MDVRPTW. His algorithm used the neighborhood structure of swap and cross to do shaking operation for the current solution, to do local search with a constrained 3-opt operator to accept the part of the poor solution to avoid getting into a local optimum for the algorithm by Threshold Accepting. Kytöjoki et al. [15] designed the guided VNS algorithm to handle the 32 existing large-scale VRP problems and compared it to the TS algorithm. The result showed that the VNS algorithm was more effective than TS algorithm in solving time. Goel and Gruhn [16] introduced the RVNS to solve the general VRP problem including time windows, vehicle constraints, path constraints, travel departure time constraints, capacity constraints, the order models compatibility constraints, multisupplier point of the orders, and transport and service position constraints. Hemmelmayr et al. [17] proposed the VNS algorithm for periodical VRP problem, adopted the saving algorithm for the construction of the initial solution, designed the move and cross neighborhood, used 3-opt operator as local search strategies, and contrasted it with other research results. Fleszar et al. [18] adopted VNS algorithm to solve the open-loop VRP problem and tested 16 benchmark problems.

Due to the complexity of the problem, the current solving quality and efficiency for the DVRP are far from the practical requirements. So, there are many problems need to make in-depth research such as how to seek feasible solutions, how to prevent falling into local optimum, and how to control

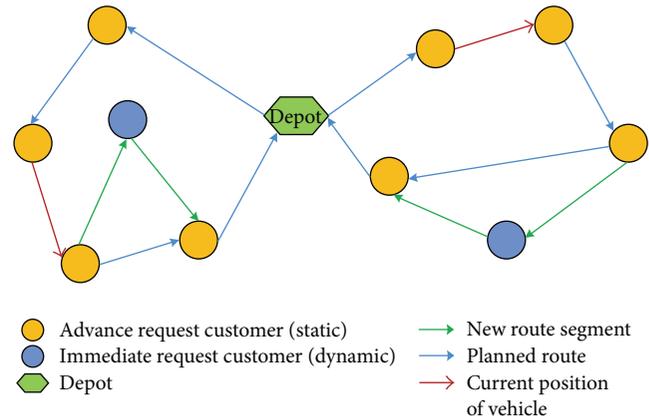


FIGURE 1: An example for dynamic vehicle routing problem.

the solution within the acceptable range. This paper presents an improved variable neighborhood search algorithm to solve DVRP; it integrates local search operator, optimization process, and the simulated annealing algorithm into the VNS algorithm framework. Through the comparison with other algorithms, it shows that the proposed algorithm can get the better solution.

3. Problem Descriptions

3.1. Problem Definition. Larsen [19] defined DVRP with two aspects: not all information relevant to the planning of the routes is known by the planner when the routing process begins; information can change after the initial routes have been constructed. And he illustrated a dynamic vehicle routing situation. The simple example is shown in Figure 1. As seen from it, two uncapacitated vehicles must service both advance and immediate request customers without time windows. The advance request customers are represented by yellow nodes, while those that are immediate requests are depicted by black nodes. The solid blue lines represent the two routes the dispatcher has planned prior to the vehicles leaving the depot. The two thick arcs indicate the vehicle positions at the time the dynamic requests are received. Ideally, the new customers should be inserted into the already planned routes without the order of the nonvisited customers being changed and with minimal delay. This is the case depicted on the right-hand side route. However, in practice, the insertion of new customers will usually be a much more complicated task and will imply a replanning of the nonvisited part of the route system. This is illustrated by the left-hand side route where servicing the new customer creates a large detour.

We can also describe the DVRP with a mathematic approach. The problem is defined on a complete graph $G = (V, E)$, where $V = \{v_0, \dots, v_L\}$ is the vertex set and $E = \{(v_i, v_j) : v_i, v_j \in V, i \neq j\}$ is the arc set. The vertices' set $D = \{v_0\}$ corresponds to the depots. Each vertex $v_i \in V$ has several nonnegative weights associated with it, namely, a demand d_i , a arrival time a_i , the waiting time w_i , service time s_i , and an earliest e_i and latest l_i possible start time for the service, which define a time window $[e_i, l_i]$. C_{ij}^t is the transportation cost

from customer i to customer j in the period t . Each vehicle k has associated with a nonnegative capacity q_k . T refers to the number of time periods, c_f is the fixed cost for a vehicle, c_t is the traveling cost per unit time, and M is a very large number.

The variables are defined as follows:

$$y_{ik} = \begin{cases} 1, & \text{if customer } i \text{ is delivered by vehicle } k, \\ 0, & \text{other,} \end{cases}$$

$$x_{ij}^t = \begin{cases} 1, & \text{if the vehicle visits the customer } j \text{ from} \\ & \text{customer } i \text{ in the } t \text{ period,} \\ 0, & \text{other.} \end{cases} \quad (1)$$

3.2. Mathematical Formulation

3.2.1. Objective Function. The dynamic vehicle routing problem with time windows is formulated in this section as a mixed integer linear programming problem. The objective of the formulation is to minimize the total cost that consists of the fixed costs of used vehicles and the routing costs:

$$\min F = c_f \times \sum_{i \in V} \sum_{j \in V \setminus \{V_0\}} \sum_{t=1}^T x_{ij}^t + c_t \times \sum_{i \in V} \sum_{j \in V \setminus \{V_0\}} \sum_{t=1}^T (c_{ij}^t \times x_{ij}^t). \quad (2)$$

3.2.2. Problem Constraints. The constraints of the problem consist of vehicle constraints, demand constraints, routing constraints, and other constraints.

Assignment of Nodes to Vehicles. Equation (3) ensure that each customer has a vehicle service for only one time and the vehicle does not return to the yard:

$$\sum_{i=0, i \neq j}^L \sum_{t=1}^T x_{ij}^t = 1, \quad j = 1, 2, \dots, L;$$

$$\sum_{j=1, j \neq i}^L \sum_{t=1}^T x_{ij}^t = 1, \quad i = 1, 2, \dots, L; \quad (3)$$

Relationship between the Vehicle and Depot. Constraint (4) ensures that the number of vehicles departed from the yard does not exceed the maximum number K of vehicles belonging to the distribution center:

$$\sum_{j=1}^L \sum_{t=1}^T x_{0j}^t \leq K. \quad (4)$$

Relationship between the Vehicle and Route. Constraint (5) ensures that customers on the same route are delivered by the same vehicle:

$$\sum_{k=1}^K k (y_{ik} - y_{jk}) \geq M \left(\sum_{t=1}^T x_{ij}^t - 1 \right), \quad \forall i, j, i \neq j,$$

$$\sum_{k=1}^K k (y_{ik} - y_{jk}) \leq M \left(1 - \sum_{t=1}^T x_{ij}^t \right), \quad \forall i, j, i \neq j. \quad (5)$$

Assignment of Nodes to Vehicles. Equation (6) states that every customer node must be serviced by a single vehicle:

$$\sum_{k=1}^K y_{ik} = 1, \quad \forall i = 1, 2, \dots, L \quad (6)$$

Capacity Constraints. Constraint (7) states that the overall load to deliver to customer sites serviced by a used vehicle v should never exceed its capacity:

$$\sum_{i=1}^L d_i y_{ik} \leq q_i, \quad \forall k = 1, 2, \dots, K. \quad (7)$$

Subcircuit Constraints. Equation (8) ensures to eliminate sub-circuit:

$$\sum_{i, j \in S \times S, i \neq j} \sum_{t=1}^T x_{ij}^t \leq |S| - 1, \quad S \in \{1, 2, \dots, L\}, \quad \forall i, j, i \neq j. \quad (8)$$

Time Constraint Violations due to Early/Late Services at Customer Sites. One has

$$a_i + w_i + s_i + c_{ij}^t - M(1 - x_{ij}^t) \leq a_j,$$

$$a_j \leq l_j, \quad (9)$$

$$w_i = \max \{e_i - a_i, 0\}.$$

Other Constraints. One has

$$x_{ij}^t = 0, 1, \quad \forall i, j, k,$$

$$y_{ik} = 0, 1, \quad \forall i, k. \quad (10)$$

4. An Improved Variable Neighborhood Search Algorithm

VNS is a metaheuristic for solving combinatorial and global optimization problems proposed by Hansen and Mladenović [1, 2]. Starting from any initial solution, a so-called shaking step is performed by randomly selecting a solution from the first neighborhood. This is followed by applying an iterative improvement algorithm. This procedure is repeated as long as a new incumbent solution is found. If not, one switches to the next neighborhood (which is typically larger) and performs a shaking step followed by the iterative improvement. If a new incumbent solution is found, one starts with the first neighborhood; otherwise one proceeds with the next

```

Main: Clarke And Wright savings algorithm
Input: the number of customers and vehicles, the capacity of vehicles
Output: set of initial solution
Begin
  for each day do
    while number of routes > number of vehicles do
      shortest route:= find route with fewest number of customers
      for each customer  $\in$  shortest route do
        delete current route
        insert in cheapest position of the remaining routes
      end for
    end while
  end for
end Begin

```

ALGORITHM 1: The initial solution based on CW.

neighborhood, and so forth. The description consists of the building of an initial solution, the shaking phase, the local search method, and the acceptance decision. The flow of VNS is shown in Figure 2.

4.1. Initial Solution. Using variable neighborhood search algorithm, first, it needs to build one or more initial feasible solution; a clustering algorithm for an initial feasible solution mainly completes two tasks: customer allocation and path planning. For obtaining an initial solution, each customer is assigned a visit day combination randomly. Routes are constructed by solving a vehicle routing problem for each day using the Clarke and Wright savings' algorithm [20]. The Clarke and Wright savings' algorithm terminates when no two routes can feasibly be merged; that is, no two routes can be merged without violating the route duration or capacity constraints shown in Algorithm 1. As a result, the number of routes may exceed the number of available vehicles. In that case, a route with the fewest customers is identified and the customers in this route are moved to other routes (minimizing the increase in costs). Note that this may result in routes that no longer satisfy the duration or capacity constraints. This step is repeated until the number of routes is equal to the number of vehicles. Since the initial solution may not be feasible, the VNS needs to incorporate techniques that drive the search to a feasible solution.

The initial solution obtained by the above method can basically meet the needs of the follow-up work, building the foundation to achieve optimal feasible solution in the following algorithm.

4.2. Shaking. Shaking is a key process in the variable neighborhood search algorithm design. The main purpose of the shaking process is to extend the current solution search space, to reduce the possibility that the algorithm falls into the local optimal solution in the follow-solving process, and to get the better solution. The set of neighborhood structures used for shaking is the core of the VNS. The primary difficulty is to find a balance between effectiveness and the chance to get out of local optimal. In the shaking execution, it first selects

a neighborhood structure N_k from the set of neighborhood structures of current solution x ; then according to the definition of N_k , x corresponds to change and generate a new solution x^* .

There are two neighborhood structures to achieve the shaking: insert and exchange. Insert operator denotes a certain period of consecutive nodes moving from the current path to another path; exchange operator refers to interchange the two-stage continuous nodes belonging to different paths. The insert and exchange operators are shown in Figure 3. The cross-exchange operator was developed by Taillard et al. [21]. The main idea of this exchange is to take two segments of different routes and exchange them. Compared with the VNS by Polacek et al. [14], the selection criterion is slightly changed. Now it is possible to select the same route twice. This allows exploring more customer visit combinations within one route. An extension to the CROSS exchange operator is introduced by Bräysy [13]; this operator is called improved cross-exchange—*i*Cross exchange for short. Both operators are used to define a set of neighborhood structures for the improved VNS.

In each neighborhood, the insert operator is applied with a probability p_{insert} to both routes to further increase the extent of the perturbation; then the probability of the exchange operator is $1 - p_{\text{insert}}$. IVNS selects randomly an exchange operator to change path for each shaking execution. The shaking process is somewhat similar to the crossover operation of the genetic algorithm. When the process is finished, the only two paths have the exchange of information; most of the features of the current solution will be preserved, to speed the convergence of the algorithm.

4.3. Local Search. In a VNS algorithm, local search procedures will search the neighborhood of a new solution space obtained through shaking in order to achieve a locally optimal solution. Local search is the most time-consuming part in the entire VNS algorithm framework and decides the final solution quality, so computational efficiency must be considered in the design process of local search algorithm. Two main aspects are considered in the design of local search

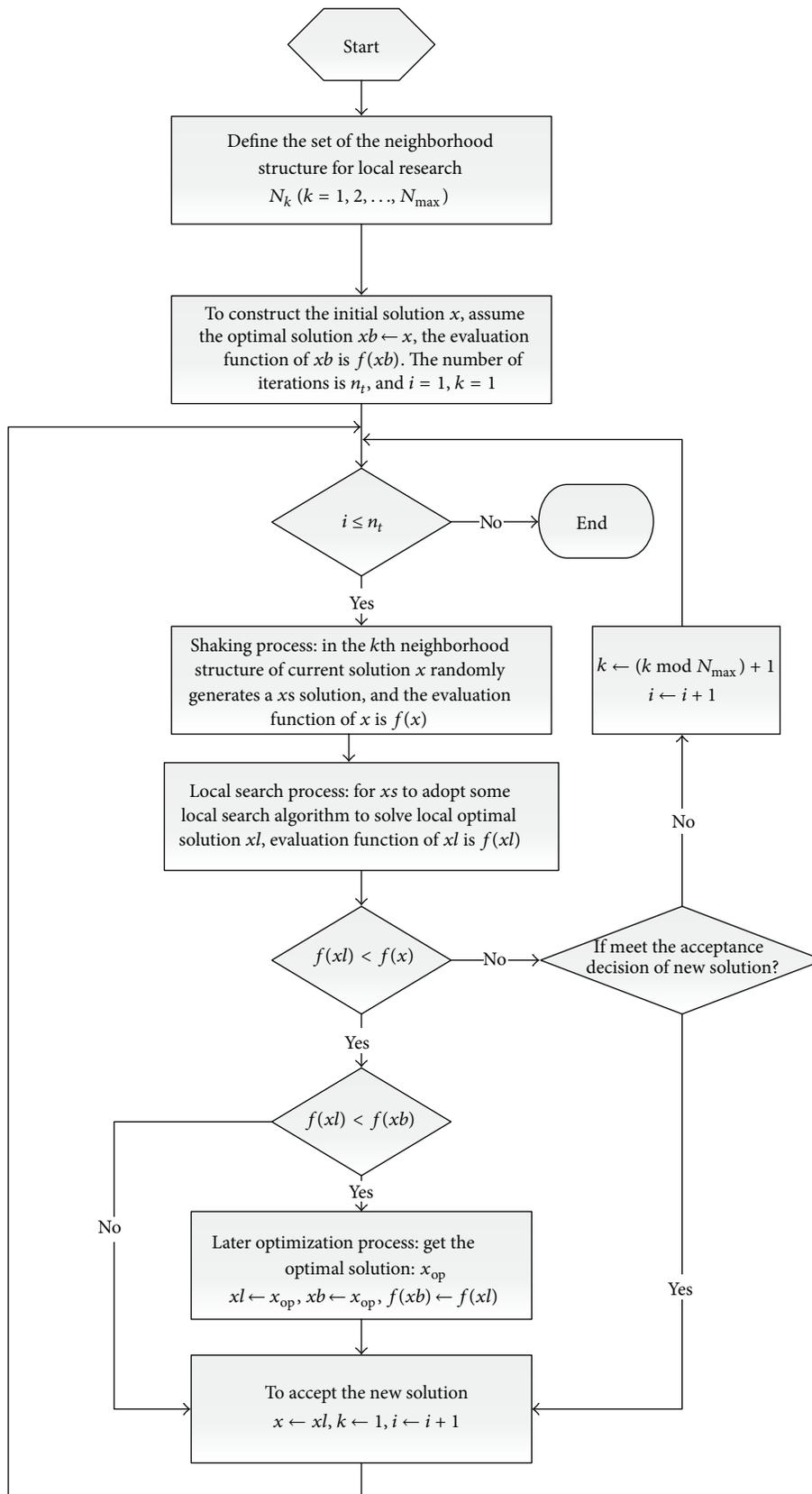
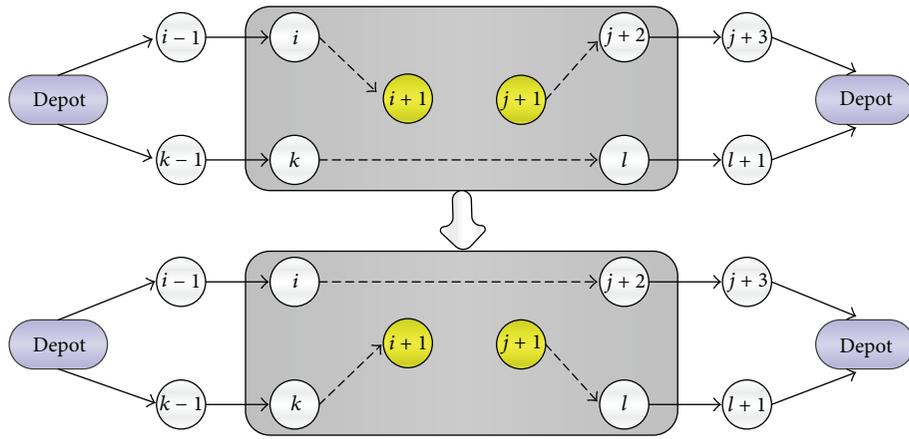
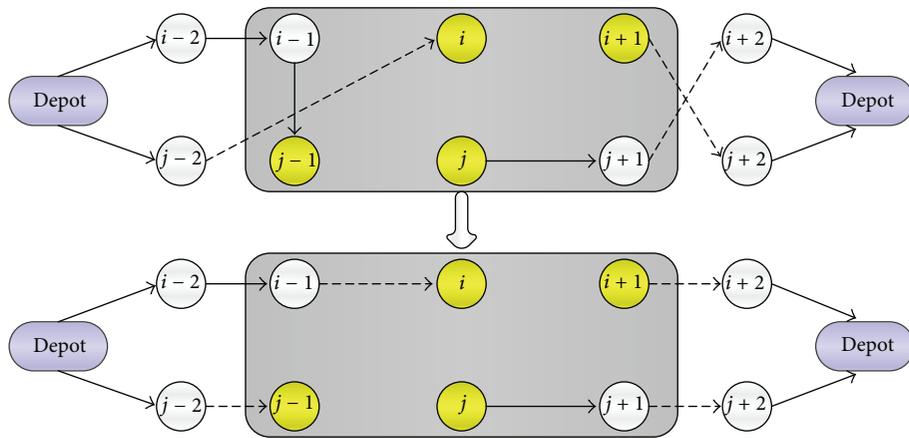


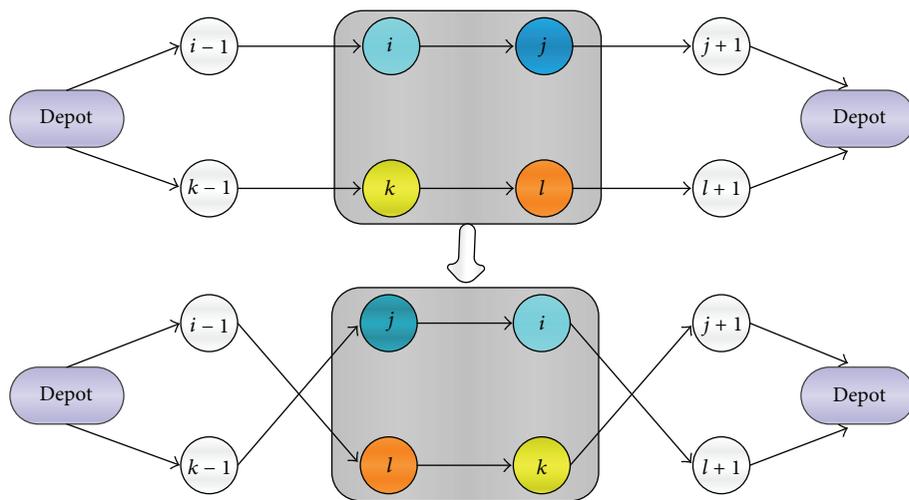
FIGURE 2: The flow of IVNS.



(a) Insert



(b) Cross



⓪ Customer node
 → Route

(c) iCross

FIGURE 3: Insert and exchange operator.

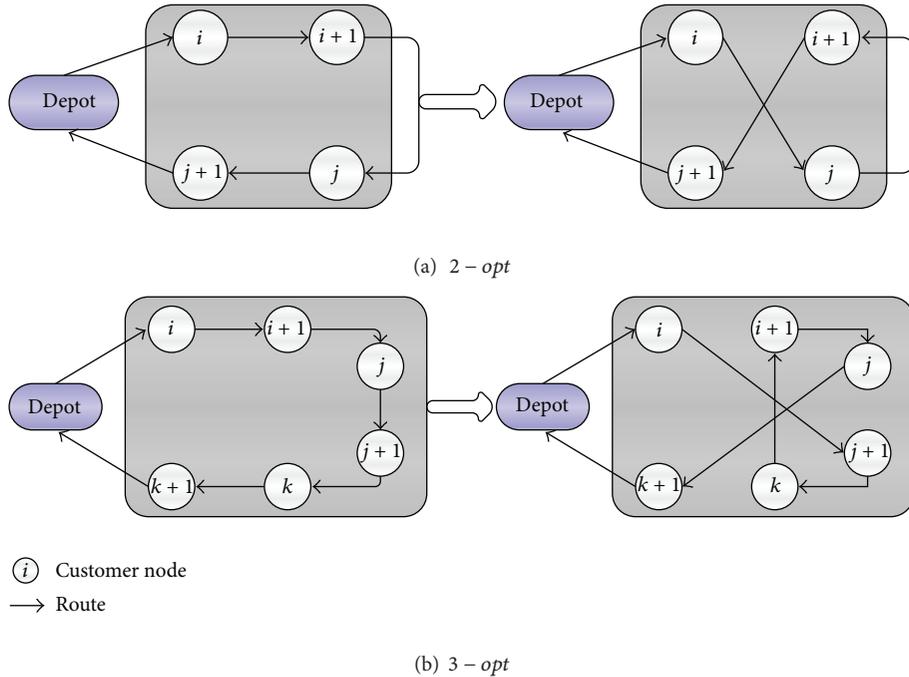


FIGURE 4: 2-opt and 3-opt strategy.

algorithms: local search operator and the search strategy. Based on the previous studies, this paper selects 2-opt and 3-opt as a local search operator in order to obtain the good quality local optimal solution in a short period; they are shown in Figure 4. According to the probability, one of the two operators is selected in each local search process. The parameter p_{2-opt} represents the probability of selection for 2-opt; similarly, the probability of selection for 3-opt can be expressed as $1 - p_{2-opt}$. This mixed operator can develop optimization ability for 2-opt and 3-opt and expand the solution space of the algorithm.

There are mainly two search strategies: first-improvement and best-improvement in local search algorithm. The former refers to access the neighborhood solution of the current x solution successively in the solution process, if the current access neighborhood solution x_n is better than x , to make $x = x_n$ and update neighborhood solution. We repeat these steps until all the neighborhood solutions of x are accessed. Finally, x will be obtained as a local optimal solution. The latter refers to traverse all of the neighborhood solution of current x solution in the solution process, to select the optimum neighborhood solution x_n as a local optimal solution. In this paper, we adopt the best-improvement strategy; it enables the algorithm to achieve a better balance in the solution quality and run time.

4.4. Later Optimization Process. In order to accelerate the convergence speed and improve the solution quality, the later optimization process is proposed in the IVNS algorithm. After the local search is completed, if the local optimal solution xl is better than the global optimal solution xb , that is, $f(xl) < f(xb)$, the later optimization process will be

continued to be implemented in order to seek a better global optimal solution [22]. The algorithm of later optimization process which was proposed by Gendreau et al. is suitable for solving the traveling salesman problem and the vehicle routing problem with time windows. The algorithm processes can be simply described as follows.

Step 1. There are some assumptions. The path that will be optimized is r , its length is n , and its value of the evaluation function is c . The final optimized path is r^* , the value of the evaluation function is c^* , and $r^* = r$, $c^* = c$, and $k = 1$.

Step 2. The *Unstring* and a *String* processes [23] are, respectively, performed for the k th customer in the r , the optimized path r' can be obtained, and its value of the evaluation function value is c' .

Step 3. If $c' < c^*$, some processes are carried out; they are $r^* = r'$, $r = r'$, $c^* = c'$, $c = c'$, and $k = 1$, jump to Step 2; otherwise, $k = k + 1$.

Step 4. If $k = n + 1$, the algorithm will be terminated; otherwise, jump to Step 2.

4.5. Acceptance Decision. The last part of the heuristic concerns the acceptance criterion. Here we have to decide whether the solution produced by VNS will be accepted as a starting solution for the next iteration. To avoid that the VNS becomes too easily trapped in local optima, due to the cost function guiding towards feasible solutions and most likely complicating the escape of basins surrounded by infeasible solutions, we also allow to accept worse solutions under certain conditions. This is accomplished by utilizing a

TABLE 1: The position and demand of static customers.

No.	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
X	29	47	45	32	6	34	44	36	38	5	27	32	17	40	37	26	34	18	43	19	25	26	29	29	14	50	47	8	42	46
Y	44	9	13	16	1	6	6	46	39	28	7	3	14	47	47	12	19	40	26	3	49	49	3	37	37	10	26	10	27	34
Demand	0.5	0.2	0.4	2	1.7	0.3	1.7	0.7	0.2	1.2	1.7	0.5	0.3	1.5	0.4	0.8	1.4	1.4	1	1.7	0.2	0.9	0.3	0.2	0.5	0.3	0.9	0.5	1.4	2

Metropolis criterion like in simulated annealing Kirkpatrick [24] for inferior solutions x^* and accepting them with a probability of (11), depending on the cost difference to the actual solution x of the VNS process and the temperature T . We update T every n_T iterations by an amount of $(T \times n_T)/\delta_{\max}$, where q_0 denotes a random number on the interval $[0, 1]$, where δ_{\max} denotes the maximal VNS iterations, and an initial temperature value is $T_0 = 10$:

$$SA(x^*, x) = \begin{cases} x^*, & \text{if } q_0 > \exp\left(\frac{f(x^*) - f(x)}{T}\right), \\ x, & \text{if } q_0 \leq \exp\left(\frac{f(x^*) - f(x)}{T}\right). \end{cases} \quad (11)$$

5. Numerical Experiments

In order to assess the performance of the improved variable neighborhood search algorithm to solve DVRP, three test problems with respect to different sizes (small, medium, and large) have been done. We analyze the solution quality and efficiency of our proposed algorithm. IVNS algorithm is implemented by the C# language, and the main configuration of the computer is an Intel Core i3 1.8 GHz, 2 GB RAM running Windows XP.

5.1. Case 1

5.1.1. *Experimental Data and Setting.* In order to assess the performance of the improved variable neighborhood search algorithm to solve DVRP, the data sets from the literature [25] are used. Firstly, the VNS algorithm is applied to solve the DVRP, and then the results are analyzed and compared with other existing algorithms.

The dynamic distribution network is randomly generated by the computer. The distribution area is a square of $50 \text{ km} \times 50 \text{ km}$; 30 static demand customers and 10 the dynamic demand customers are randomly generated. Each customer's demand is a random number of $[0, 2]$, the vehicle's capacity of distribution centers is 8 t, and the maximum driving distance of vehicle once is 100 km. These information including the coordinates and demand of 30 static customers and 10 dynamic customers are randomly generated by the computer, the location of the distribution center is $O(25 \text{ km}, 25 \text{ km})$, and the number of vehicles is 3. The objective is to arrange the delivery route of the vehicle reasonably, so that the distribution mileage is the shortest. For simplicity, the distance between customer and distribution center uses the straight-line distance. The coordinates and demands of static customers and dynamic customers are shown in

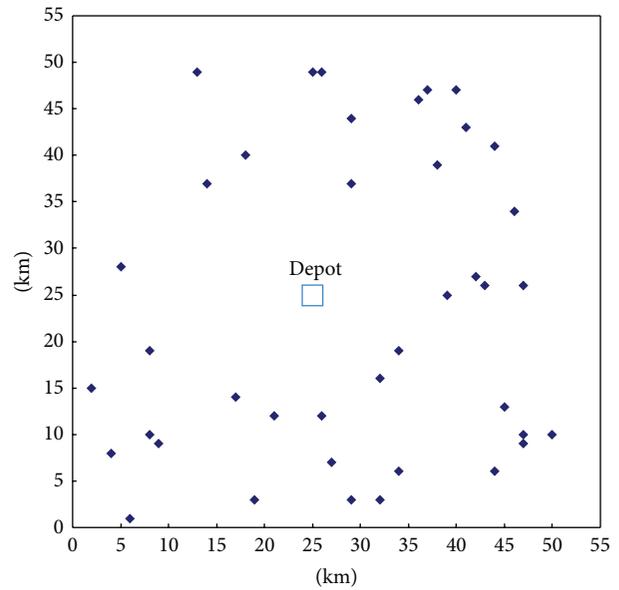


FIGURE 5: The position relationship of customer and depot.

TABLE 2: The position and demand of dynamic customers.

No.	A	B	C	D	E	F	G	H	I	J
X	8	47	2	9	21	41	39	4	44	13
Y	19	10	15	9	12	43	25	8	41	49
Period	1	2	2	1	1	1	1	2	1	2
Demand	1	1.5	1.6	3	1.4	0.3	0.2	2.2	2.8	1

Tables 1 and 2, respectively, the specific position relationship for the customer and the customer, and the customer and distribution center are shown in Figure 5.

The initial values of the various parameters for IVNS algorithm are set as follows.

- (1) The parameter settings for simulated annealing accepted criteria are initial temperature $T_0 = 10$, every $n_T = n/10$ generation to update temperature $T_{n+1} = 0.9 \times T_n$, $n_t = 1000$ to end the algorithm.
- (2) The parameters value of the Shaking operation are as follows: $p_{\text{insert}} = 0.2$, $p_{\text{cross}} = 0.15$, and $p_{\text{icross}} = 0.1$.
- (3) The $p_{2\text{-opt}}$ value is 0.5 in local search.

5.1.2. *Numerical Results.* First, distribution of fixed-demand customer is optimized, solved, and generated initial distribution route, as shown in Table 3.

The customers' demand information is updated at period 1; at this moment, the service requests are put forward by

TABLE 7: The comparison results of Lackner and IVNS.

No.	Dod (%)	Average vehicle number			Average driving distance/km			Average calculating time/s		
		IVNS	Lackner	ARE (%)	IVNS	Lackner	ARE (%)	IVNS	Lackner	ARE(%)
R1	90	14.25	15.15	-5.94	1343.82	1278.33	5.12	16.87	17.85	-5.49
	70	14.23	15.02	-5.26	1331.35	1336.1	-0.36	19.06	19.87	-4.08
	50	15.23	14.19	7.33	1293.81	1329.98	-2.72	25.69	27.46	-6.45
	30	13.47	14.22	-5.27	1286.63	1337.86	-3.83	48.34	46.97	2.92
	10	14.8	13.9	6.47	1259.18	1278.06	-1.48	70.01	68.01	2.94
C1	90	11.25	11.78	-4.50	1235.47	1479.6	-16.50	6.31	6.47	-2.47
	70	11.26	11.87	-5.14	1031.78	1261.3	-18.20	11.08	10.81	2.50
	50	12.21	11.97	2.01	1072.32	1236.06	-13.25	17.77	18.32	-3.00
	30	10.57	10.54	0.28	970.8	1066.89	-9.01	28.06	27.08	3.62
	10	10.23	10.78	-5.10	895.68	996.35	-10.10	42.46	44.25	-4.05
RC1	90	15	15	0.00	1506.43	1475.21	2.12	17.66	17.04	3.64
	70	13.58	14.65	-7.30	1513.23	1488.44	1.67	24.12	28.5	-15.37
	50	13.19	12.57	4.93	1519.12	1448.07	4.91	49.84	47.98	3.88
	30	12.98	13.43	-3.35	1489.96	1439.71	3.49	44.12	46.21	-4.52
	10	12.88	13.15	-2.05	1431.79	1426.89	0.34	86.78	88.21	-1.62
R2	90	3.23	3.56	-9.27	1045.89	1193.33	-12.36	14.1	13.9	1.44
	70	3.79	3.61	4.99	1034.22	1116.93	-7.41	19.32	21.75	-11.17
	50	3.33	3.65	-8.77	1012.11	1138.78	-11.12	32.11	32.08	0.09
	30	4.22	4.87	-13.35	987.42	1085.42	-9.03	55.22	58.29	-5.27
	10	6.22	6.1	1.97	951	1052.85	-9.67	71.01	70.11	1.28
C2	90	3.14	3.02	3.97	695.5	792.46	-12.24	6.21	5.52	12.50
	70	3.67	3.9	-5.90	671.86	743.78	-9.67	9.14	9.91	-7.77
	50	3.88	3.7	4.86	610.98	689.25	-11.36	18	18	0.00
	30	3.19	3.5	-8.86	655.23	632.33	3.62	27.18	28.35	-4.13
	10	4	4	0.00	582.32	629.08	-7.43	62.82	63.67	-1.34
RC2	90	5	5	0.00	1351.21	1476.76	-8.50	9.96	10.89	-8.54
	70	3.98	4.1	-2.93	1256.32	1346.76	-6.72	19.03	18.98	0.26
	50	4.32	4.45	-2.92	1189.13	1269.29	-6.32	25.78	27.96	-7.80
	30	5.21	5.65	-7.79	1151.35	1244.85	-7.51	38.67	39.11	-1.13
	10	6.8	7.1	-4.23	1183.11	1220.9	-3.10	57.1	57.29	-0.33
Average		8.637	8.81	-2.01	1118.63	1183.72	-5.50	32.46	33.03	-1.72

the four algorithms of search success rate from the smallest to largest is tabu search algorithm, two-stage algorithm, genetic algorithm, and IVNS; the order for the worst value is as follows: the average value from small to big is IVNS, two-stage algorithm, genetic algorithm, and tabu search algorithm; it reflects the IVNS has the better global search capability. The number of iterations for four algorithms from the smallest to the largest is IVNS, tabu search algorithm, two-stage algorithm, and genetic algorithm, this also shows that convergence rate of IVNS is faster than the other algorithms, and it can be more appropriate to solve dynamic vehicle routing problem to some extent.

5.2. Case 2. The experimental test uses the benchmark data which was 100-node VRPTW calculation example and compiled by Solomon in 1987. Every sample contains 100 nodes and distributes into 100×100 Euclidean plane. The sample is divided into six categories: R1, R2, C1, C2, RC1, and RC2. DVRPTW adopts the Lackner dynamic test data

set which is designed in 2004 based on the Solomon example. For each question in the Solomon example, there are five data sets corresponding to it. They are 90%, 70%, 50%, 30%, and 10% five dynamic degree.

The dynamic degree is described as follows:

$$\text{Dod} = \frac{N_d}{N_d + N_s} \times 100\%. \quad (12)$$

N_d is the number of dynamic customer demand, and N_s is the number of static customer demand.

Table 7 gives the comparison results of the Solomon problem to IVNS and Lackner under different dynamic degrees. From the number of vehicles, the average driving distance, and the average calculation time, we compared the calculation results and the relative error. For five dynamic degrees, the average values of the number of vehicles, respectively, are 8.637 and 8.81 for IVNS and Lackner, the

TABLE 8: The comparison results of Gehring and IVNS.

Instance set	Objective		The number of vehicles		The calculation time/s		Total demand	New demand	Reject demand
	Gehring	IVNS	Gehring	IVNS	Gehring	IVNS			
C1_10.1	43494.2	43265.9	100	95	207	201	720	122	0
C2_10.1	21778.4	21763	31	30	216	212	640	114	2
R1_10.1	56389.8	56383.1	103	100	189.6	190.3	701	90	0
R2_10.1	37014.7	36982.2	19	20	237	228	550	49	2
RC1_10.1	48056.1	48005.7	97	98	316.8	313.5	853	80	3
RC2_10.1	33534.3	33156.1	23	21	507.6	499.8	690	189	0
Average	40044.58	39926.00	62.17	60.67	279.00	274.10	692.33	107.33	1.17

relative error is -2.01 ; the average values of driving distance respectively are 1118.63 and 1183.72 for IVNS and Lackner, and the relative error is -5.50 ; the average values of calculation time, respectively, are 32.46 and 33.03 for IVNS and Lackner, and the relative error is -1.72 . According to Table 7, the following conclusions can be drawn.

- (1) In terms of the total driving distance, relative to the best results which are obtained by Lackner's various algorithms, except for the RC1, R1, and C2, the proposed algorithm can obtain better calculation results for other type's dynamic degrees.
- (2) The present algorithm, in which the average computation time is less than 90 seconds, fully meets the requirements of real-time scheduling. The average calculation time of IVNS is lower than Lackner; it demonstrates that the algorithm has better search capabilities.

5.3. *Case 3.* In order to verify the proposed variable neighborhood search algorithm to solve the large-scale dynamic vehicle routing problem effectively, this paper extends to large-scale test instances with 1000 customer which are presented by Gehring and Homberger. Table 8 gives the comparison results of the first path adjustment for the part instance based on Gehring and our proposed algorithm.

As shown in Table 8, the results of the objective value, the number of vehicles, and the calculation time are compared based on Gehring and IVNS. 107.33 average new demands are occurred, and 692.33 total demands are handled in the path adjustment instance; only 1.17 customer demands failed to be satisfied, while the service level is close to 100%. In dealing with a large-scale problem, the algorithm running time is an important evaluation index; the average running time of our proposed algorithm is 274.10 s; it is smaller than Gehring's algorithm. The number of vehicle is closely related to the cost, and it is decreased from 62.17 to 60.67. The objective value is reduced from 40044.58 to 39926.00; it demonstrates that the results have been improved. According to the analysis, to some extent, the proposed algorithm can solve the large-scale dynamic VRP and improve the results.

6. Conclusions

In this research, we proposed a formulation for a dynamic vehicle routing problem. We also presented an improved variable neighborhood search (IVNS) metaheuristic for DVRP. In the initial solution, the routes are constructed by solving a vehicle routing problem for each day using the Clarke and Wright savings algorithm Clarke and Wright [20]. In the shaking execution, there are two neighborhood structures to achieve the shaking: insert and exchange. This paper selects $2-opt$ and $3-opt$ as a local search operator in order to obtain the good quality local optimal solution in a short period. In order to accelerate the convergence speed and improve the solution quality, the later optimization process is proposed in the IVNS algorithm. To test the performance of the proposed improved variable neighborhood search algorithm, we test three problems with respect to different sizes (small, medium, and large) and compare the results with other algorithms. The results show that the proposed model and the algorithm are effective and can solve the DVRP within a very short time and improve the quality of solution to some extent.

Acknowledgments

Funding for this research is supported by the National Science Foundation of China under Grants nos. 90924020, 70971005, and 71271013, the Societal Science Foundation of China under Grant no. 11AZD096, China Postdoctoral Science Foundation funded project no. 2012M520008, and Quality Inspection Projects no. 200910088 and no. 201010268.

References

- [1] P. Hansen and N. Mladenović, "Variable neighborhood search for the p-median," *Location Science*, vol. 5, no. 4, pp. 207–226, 1997.
- [2] P. Hansen, N. Mladenović, and D. Perez-Britos, "Variable neighborhood decomposition search," *Journal of Heuristics*, vol. 7, no. 4, pp. 335–350, 2001.
- [3] M. Gendreau and J.-Y. Potvin, "Dynamic vehicle routing and dispatching," in *Fleet Management and Logistic*, T. G. Crainic and G. Laporte, Eds., pp. 115–226, Kluwer Academic, 1998.
- [4] B. Fleischmann, S. Gnutzmann, and E. Sandvoß, "Dynamic vehicle routing based on online traffic information," *Transportation Science*, vol. 38, no. 4, pp. 420–433, 2004.

- [5] J. Yang, P. Jaillet, and H. Mahmassani, "Real-time multivehicle truckload pickup and delivery problems," *Transportation Science*, vol. 38, no. 2, pp. 135–148, 2004.
- [6] R. Montemanni, L. M. Gambardella, A. E. Rizzoli, and A. V. Donati, "Ant colony system for a dynamic vehicle routing problem," *Journal of Combinatorial Optimization*, vol. 10, no. 4, pp. 327–343, 2005.
- [7] M. Gendreau, F. Guertin, J. Y. Potvin, and R. Séguin, "Neighborhood search heuristics for a dynamic vehicle dispatching problem with pick-ups and deliveries," *Transportation Research C*, vol. 14, no. 3, pp. 157–174, 2006.
- [8] A. Goel and V. Gruhn, "Solving a dynamic real-life vehicle routing problem," *Operations Research Proceedings*, vol. 2005, no. 8, pp. 367–372, 2006.
- [9] J. Schönberger and H. Kopfer, "Online decision making and automatic decision model adaptation," *Computers and Operations Research*, vol. 36, no. 6, pp. 1740–1750, 2009.
- [10] C. Novoa and R. Storer, "An approximate dynamic programming approach for the vehicle routing problem with stochastic demands," *European Journal of Operational Research*, vol. 196, no. 2, pp. 509–515, 2009.
- [11] R. M. Branchini, V. A. Armentano, and A. Løkketangen, "Adaptive granular local search heuristic for a dynamic vehicle routing problem," *Computers and Operations Research*, vol. 36, no. 11, pp. 2955–2968, 2009.
- [12] J. Müller, "Approximative solutions to the bicriterion vehicle routing problem with time windows," *European Journal of Operational Research*, vol. 202, no. 1, pp. 223–231, 2010.
- [13] O. Bräysy, "A reactive variable neighborhood search for the vehicle-routing problem with time windows," *INFORMS Journal on Computing*, vol. 15, no. 4, pp. 347–368, 2003.
- [14] M. Polacek, R. F. Hartl, K. Doerner, and M. Reimann, "A variable neighborhood search for the multi depot vehicle routing problem with time windows," *Journal of Heuristics*, vol. 10, no. 6, pp. 613–627, 2004.
- [15] J. Kytöjoki, T. Nuortio, O. Bräysy, and M. Gendreau, "An efficient variable neighborhood search heuristic for very large scale vehicle routing problems," *Computers and Operations Research*, vol. 34, no. 9, pp. 2743–2757, 2007.
- [16] A. Goel and V. Gruhn, "A general vehicle routing problem," *European Journal of Operational Research*, vol. 191, no. 3, pp. 650–660, 2008.
- [17] V. C. Hemmelmayr, K. F. Doerner, and R. F. Hartl, "A variable neighborhood search heuristic for periodic routing problems," *European Journal of Operational Research*, vol. 195, no. 3, pp. 791–802, 2009.
- [18] K. Fleszar, I. H. Osman, and K. S. Hindi, "A variable neighbourhood search algorithm for the open vehicle routing problem," *European Journal of Operational Research*, vol. 195, no. 3, pp. 803–809, 2009.
- [19] A. Larsen, "The dynamic vehicle routing problem," IMM, DTU, 2001.
- [20] G. Clarke and J. W. Wright, "Scheduling of vehicles from a central depot to a number of delivery points," *Operations Research*, vol. 12, no. 4, pp. 568–581, 1964.
- [21] É. Taillard, P. Badeau, M. Gendreau, F. Guertin, and J. Y. Potvin, "A tabu search heuristic for the vehicle routing problem with soft time windows," *Transportation Science*, vol. 31, no. 2, pp. 170–186, 1997.
- [22] Z. Wang, J. Zhang, and X. Wang, "A modified variable neighborhood search algorithm for multi depot vehicle routing problem with time windows," *Chinese Journal of Management Science*, vol. 19, no. 2, pp. 99–108, 2011.
- [23] M. Gendreau, A. Hertz, G. Laporte, and M. Stan, "A generalized insertion heuristic for the traveling salesman problem with time windows," *Operations Research*, vol. 46, no. 3, pp. 330–335, 1998.
- [24] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi, "Optimization by simulated annealing," *Science*, vol. 220, no. 4598, pp. 671–680, 1983.
- [25] W. Xu, G. Xianlong, and D. Ying, "Research on dynamic vehicle routing problem based on two-phase algorithm," *Control and Decision*, vol. 27, no. 2, pp. 175–181, 2012.



Hindawi

Submit your manuscripts at
<http://www.hindawi.com>

