

Research Article

A Hybrid GRASP+VND Heuristic for the Two-Echelon Vehicle Routing Problem Arising in City Logistics

Zheng-yang Zeng, Wei-sheng Xu, Zhi-yu Xu, and Wei-hui Shao

School of Electronics and Information Engineering, Tongji University, Shanghai 201804, China

Correspondence should be addressed to Wei-sheng Xu; xuweisheng@tongji.edu.cn

Received 29 November 2013; Accepted 30 March 2014; Published 23 April 2014

Academic Editor: Baozhen Yao

Copyright © 2014 Zheng-yang Zeng 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.

The two-echelon vehicle routing problem (2E-VRP) is a variant of the classical vehicle routing problem (VRP) arising in two-level transportation systems such as those encountered in the context of city logistics. In the 2E-VRP, freight from a depot is compulsorily delivered through intermediate depots, named satellites. The first echelons are routes that distribute freight from depot to satellites, and the second are those from satellites to customers. This problem is solved by a hybrid heuristic which is composed of a greedy randomized adaptive search procedure (GRASP) with a route-first cluster-second procedure embedded and a variable neighborhood descent (VND), called GRASP+VND hereafter. Firstly, an extended split algorithm in the GRASP continuously splits randomly generated permutations of all customers and assigns customers to satellites reasonably until a feasible assignment appears, and a complete 2E-VRP feasible solution is obtained by solving the first echelon problem subsequently and, secondly, a VND phase attempts to improve this solution until no more improvements can be found. The process above is iterated until the maximum number of iterations is reached. Computational tests conducted on three sets of benchmark instances from the literature show that our algorithm is both effective and efficient and outperforms the best existing heuristics for the 2E-VRP.

1. Introduction

The transportation of freight constitutes an extremely important activity taking place in urban areas, but it is also very disturbing. The increase in the number of freight transportation trucks using urban roads makes a more and more significant contribution to traffic congestion and many associated negative environmental impacts, such as air pollution and noise. For the purpose of preventing the urban environment from getting worse, many municipalities place restrictions on these big trucks to keep them out of their city centers by creating peripheral intermediate facilities, called satellites. External carriers need to supply these satellites from central depots and then smaller and environmentally friendly vehicles would distribute the freight downtown from these satellites [1–3]. Therefore, two distribution echelons are involved in city logistics. With the customer demands, satellite capacities, and vehicle capacities from the two levels known in advance, the two-echelon vehicle routing problem (2E-VRP) [4, 5] consists in building a set of the least-cost trips (the fewest number

of vehicles used and least vehicle traveling cost) for the two echelons.

The 2E-VRP is a new two-echelon variant of the well-known vehicle routing problem (VRP). Several models have been proposed and different kinds of algorithms have been designed, including both exact algorithms [4–7] and heuristic algorithms [8–11].

Perboli et al. [4] introduced a family of two-echelon vehicle routing problems and proposed a three-index flow-based formulation for the 2E-VRP. The authors also introduced some valid inequalities and two math-heuristics based on the 2E-VRP model, which were used within a branch-and-cut framework. They were able to solve to optimality instances containing up to 21 customers.

Perboli and Tadei [5] proposed several new classes of valid inequalities based on the traveling salesman problem (TSP) and the VRP and strengthened the previous 2E-VRP formulation with new cuts (including capacity cuts), which allowed their algorithm to solve seven new instances to

optimality and reduce the optimality gap on several other instances.

Jepsen et al. [6] presented an edge flow based model for the 2E-VRP and employed a specialized branching scheme to branch on infeasible integer solutions in their branch-and-cut algorithm to obtain feasible solutions. Their algorithm was able to solve 47 instances to optimality, surpassing previous exact algorithms. They found that the coupling between the two echelons in the 2E-VRP would pose a challenge to incorporate.

Baldacci et al. [7] proposed a new mathematical formulation of the 2E-VRP (used to derive valid lower bounds) and a new exact method. They decomposed the 2E-VRP into a limited set of multidepot vehicle routing problems (MDVRPs) with side constraints. Computational results on extensive benchmark instances showed that their exact algorithm outperformed the state-of-the-art exact methods in terms of size, number of problems solved to optimality, and computing time.

Since exact algorithms are usually computationally expensive for large-scale combinatorial optimization problems, approximate solutions with sufficient accuracy that can be obtained fast are often desired in practice.

Crainic et al. [8] developed a family of multistart heuristics, based on separating the depot-to-satellite transfer and the satellite-to-customer delivery by iteratively solving the two resulting routing subproblems, while adjusting the satellite workloads that linked them. Besides, an intensification phase aiming at improving feasible solutions by local search was followed by a diversification phase to avoid local optimum in their algorithms. Their ideas are very useful to handle the 2E-VRP; however, they obtained relatively poor results even in small-scale instances.

Meihua et al. [9] proposed a hybrid ant colony optimization algorithm which combined three heuristics for the 2E-VRP. They firstly divided the problem into several VRPs by a separation strategy and then applied improved ant colony optimization with multiple neighborhood descent to build better feasible solutions. Computational tests on 22 benchmark instances from the literature showed that their algorithm was better than previous published algorithms.

Hemmelmayr et al. [10] developed an adaptive large neighborhood search (ALNS) heuristic for the 2E-VRP, based on the destroy-and-repair principle in which two different sets of operators (destroy operators and repair operators) are alternated. They used existing operators and new operators designed specifically for the 2E-VRP. Their algorithm was shown to provide better solutions and outperform existing heuristic methods for the 2E-VRP but is much complicated to implement in practice because of adopting too many kinds of operators (8 kinds of destroy operators, 5 kinds of repair operators, and 5 kinds of local search operators) and many parameters.

Crainic et al. [11] proposed a heuristic algorithm based on greedy randomized adaptive search procedure (GRASP) combined with path relinking to address the 2E-VRP. The problem was treated by separating the depot-to-satellite transfer and the satellite-to-customer delivery and iteratively

solving the two resulting routing subproblems, while adjusting the satellite workloads that link them; this idea is the same as Crainic et al. [8]. The path relinking procedure with feasibility search was applied to link current solution to elite one. The computational results on instances with up to 50 customers and 5 satellites showed that the proposed heuristic can improve literature results, in both efficiency and accuracy, but their solution quality cannot outperform the ALNS of Hemmelmayr et al. [10].

According to the published computational results of these heuristic methods mentioned above, we can retrieve that the best existing heuristic for the 2E-VRP is the ALNS algorithm of Hemmelmayr et al. [10].

Greedy randomized adaptive search procedure (GRASP) is one of the most well-known multistart heuristics for combinatorial optimization problems. It was introduced by Feo and Resende [12]. Each GRASP iteration consists basically of constructing a feasible solution and then applying a local search procedure to improve it until a local optimum is found, and the best overall solution is kept as the final result [12, 13].

Variable neighborhood descent (VND) is a deterministic version of variable neighborhood search (VNS), originally proposed by Mladenović and Hansen [14]. VNS is a meta-heuristic for solving combinatorial optimization problems, whose basic idea is a systematic change of neighborhoods, both within a descent phase to find a local optimum and in a perturbation phase to get out of the corresponding valley [14, 15]. VNS explores an ordered list of neighborhoods. It starts with a given neighborhood and switches to the next one in the list when it finds a local minimum. The search is reinitialized from the first neighborhood whenever a new better solution is found or when all neighborhoods have been checked. VND also changes the neighborhood operators once the search is stuck in a local optimum, but it differs from VNS that no random perturbation is applied.

The hybrid of GRASP and VND has successfully solved several kinds of routing problems, such as pickup-and-delivery traveling salesman problem [16], truck and trailer routing problem [17], traveling repairman problem [18], three-dimensional bin packing problem [19], and school bus routing problem [20]. Because the combinations of GRASP and VND are not only effective and efficient in solving combinatorial optimization problems but also very simple to implement, we design a hybrid GRASP+VND heuristic for the 2E-VRP based on the characteristics of this problem to meet the practical requirements arising in city logistics.

The remainder of this paper is organized as follows. Section 2 first describes the 2E-VRP and its mathematical model. Section 3 gives the details of our hybrid heuristic for the 2E-VRP. Section 4 presents computational results on three sets of instances. Section 5 contains the discussion and Section 6 gives the conclusion.

2. Problem Description and Mathematical Formulation

The definition of 2E-VRP provided here follows those in [4, 7, 10]. 2E-VRP can be defined on a weighted, complete, and undirected graph $G = (V, A)$ with node set V and arc set

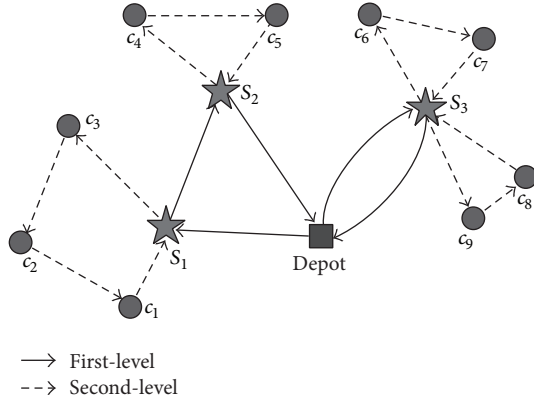


FIGURE 1: A solution to the 2E-VRP.

A. Node set V is composed of three kinds of nodes: a depot V_0 , a subset V_S of n_s satellites, and a subset V_C of n_c customers. In the arc set A , each arc (i, j) represents the shortest path in the actual road network, with a known traveling cost c_{ij} . Costs in both levels are assumed to satisfy the triangle inequalities. Each customer $i \in V_C$ has an associated demand d_i , known in advance, and cannot be split. The demand should not be delivered by direct shipping from the depot but must be consolidated in a satellite. The deliveries to the satellites on the first echelon can be split. Each vehicle has a capacity constraint that has to be respected. This capacity is the same for all vehicles belonging to the same level but may differ for each level. The capacities of the first- and the second-level vehicles are denoted by Q_1 and Q_2 , respectively. The total number of vehicles available is given by m^1 for the first level and m^2 for the second. Each satellite $k \in V_S$ has a capacity B_k that limits the total amount of customers' demands that can be delivered to it by first-level routes. Moreover, there are m_k first-level vehicles available at each satellite $k \in V_S$. No additional limitation on the route size, neither in length nor in number of visited customers, is introduced.

Figure 1 illustrates a 2E-VRP solution with a depot, three satellites, and nine customers. The satellites are represented by five-pointed stars and customers by circles and the depot by a square. The routes that serve the satellites from the depot are called the first-level routes. The second-level routes are those that start from a satellite, visit the customers, and return to the same satellite. Vehicle routes of the first and the second echelon are represented by continuous thick lines and dashed thin lines, respectively.

2.1. Mathematical Formulation. Based on the above description, the 2E-VRP can be formulated as follows [7]. Let \mathcal{M} be the index set of all first-level routes, and let $\mathcal{M}_k \subseteq \mathcal{M}$ be the subset of first-level routes serving satellite $k \in V_S$. Let R_r and $A(r)$ be the subset of satellites visited and subset of arcs traversed by a first-level route $r \in \mathcal{M}$, respectively. Let \mathcal{R}_k be the index of the second-level routes passing through satellite $k \in V_S$, and let $\mathcal{R}_{ik} \subseteq \mathcal{R}_k$ be the subset of routes passing through satellite $k \in V_S$ and customer $i \in V_C$. The subset of customers visited and the subset of arcs traversed by

a second-level route $l \in \mathcal{R}_k$ are represented by R_{kl} and $A(l)$, respectively. Let y_r be a binary variable equal to 1 if and only if route $r \in \mathcal{M}$ is in 2E-VRP solution, x_{kl} a binary variable equal to 1 if and only if route $l \in \mathcal{R}_k$ of satellite $k \in V_S$ is in 2E-VRP solution, and d_{kr} a nonnegative integer variable representing the quantity delivered by first-level route $r \in \mathcal{M}$ to satellite $k \in R_r$. And the mathematical model of the 2E-VRP can be stated as follows [7]:

$$\min \sum_{k \in V_S} \sum_{l \in \mathcal{R}_k} \sum_{(i,j) \in A(l)} c_{ij} x_{kl} + \sum_{r \in \mathcal{M}} \sum_{(i,j) \in A(r)} c_{ij} y_r \quad (1)$$

$$\text{subject to: } \sum_{k \in V_S} \sum_{l \in \mathcal{R}_k} x_{kl} = 1, \quad i \in V_C, \quad (2)$$

$$\sum_{l \in \mathcal{R}_k} x_{kl} \leq m_k, \quad k \in V_S, \quad (3)$$

$$\sum_{k \in V_S} \sum_{l \in \mathcal{R}_k} x_{kl} \leq m^2, \quad (4)$$

$$\sum_{l \in \mathcal{R}_k} \sum_{i \in R_{kl}} d_i x_{kl} \leq B_k, \quad k \in V_S, \quad (5)$$

$$\sum_{r \in \mathcal{M}} y_r \leq m^1, \quad (6)$$

$$\sum_{r \in \mathcal{M}_k} d_{kr} = \sum_{l \in \mathcal{R}_k} \sum_{i \in R_{kl}} d_i x_{kl}, \quad k \in V_S, \quad (7)$$

$$\sum_{r \in R_r} d_{kr} \leq Q_1 y_r, \quad r \in \mathcal{M}, \quad (8)$$

$$x_{kl} \in \{0, 1\}, \quad k \in V_S, \quad l \in \mathcal{R}_k, \quad (9)$$

$$y_r \in \{0, 1\}, \quad r \in \mathcal{M}, \quad (10)$$

$$d_{kr} \in \mathbb{Z}_+, \quad k \in R_r, \quad r \in \mathcal{M}. \quad (11)$$

The objective function (1) aims to minimize the total cost of two echelons. Constraints (2) ensure that each customer $i \in V_C$ must be visited by exactly one second-level route. Constraints (3), (4), and (6) impose the upper bounds on the number of first- and second-level routes in the solution. Constraints (5) specify the capacities of each satellite. The balance between the quantity delivered by first-level routes to every satellite and the customers demands supplied from this satellite is ensured by constraints (7). Constraints (8) impose that the vehicle capacity of the first-level vehicles is not exceeded.

Each second-level route must begin and end at the same satellite, and each customer must be served by exactly one second-level vehicle. The demand of each satellite is the total demand of its assigned customers, so each satellite must receive enough freight from the depot to satisfy the customers of its second-level routes. Besides, any change to the customer-to-satellite assignment affects the first-level routing and therefore has an impact on the first-level transportation costs. The objective function to be minimized is the global transportation costs in both levels. 2E-VRP can be easily seen to be a reduction to the vehicle routing problem (VRP), which

is a special case of 2E-VRP arising when just one satellite is considered, so it is also NP-hard [10].

3. The Hybrid GRASP+VND Heuristic

The proposed algorithm is a memoryless multistart heuristic method in which each iteration consists of two phases: a *greedy randomized adaptive search procedure* (GRASP) construction phase and a *variable neighborhood descent* (VND) improvement phase. Since the solutions generated by the GRASP construction phase are not guaranteed to be locally optimal, it may be very beneficial to apply a local search to further improve each constructed solution [12, 17]. We use the term GRASP+VND to symbolize this hybrid algorithm. GRASP+VND independently creates and improves a number of initial solutions and in the end returns the best solution obtained during the entire search.

3.1. Search Space. The search is restricted to feasible solutions. We do not allow any violations of the constraints on the vehicle capacity, number of vehicles available, and the capacity constraints of the satellites. The reason that we always maintain the feasibility of solutions is that it is quite difficult and time consuming to perform feasibility repairing (due to the constraints of the number and capacity of the second-level vehicles) after the solutions are infeasible in the search.

The initial solutions of GRASP+VND are generated from solutions encoded as random permutations (TSP tours covering the entire n_c customers). Any random permutation T can be converted into a 2E-VRP solution S (with respect to the orders of customers in T) using an extended version of the splitting procedure *split* of Nguyen et al. [21]. Customers are assigned to satellites by this extended *split* according to their orders in T in a relatively reasonable manner and then the demand of each satellite is obtained by the summation of the demands of all its assigned customers.

With the obtained demands of all the satellites, the first-level routes are constructed as follows [10, 22]: we first create as many full-load direct trips from the depot to each satellite as possible until the remaining demand of each satellite is less than the capacity Q_1 of the first-level vehicle and then solve a TSP tour using the saving algorithm of Clarke and Wright [23], as it was handled by Hemmelmayr et al. [10].

3.2. Initial Solution Construction Phase. The goal of GRASP is to create initial feasible solutions for the second phase to further improve them. For the hybrid GRASP+VND algorithm to work well, it is essential that solutions of relatively high quality are constructed during the solution construction phase.

The idea of splitting a good TSP tour to a VRP solution was first introduced by Beasley [24], as a route-first cluster-second heuristic for the VRP. A random permutation T of all customers is first cut into second-level routes by an extended version of the *split* algorithm of Nguyen et al. [21]. The first stage of our extended algorithm consists of minimizing the number of second-level routes, that is, the number of second-level vehicles used. This extension [25] of *split* defines

```
(1) Initialize random generator;
(2) repeat
(3)    $T \leftarrow \text{RandomPermutation}(V_C)$ ;
(4)    $\text{Trips} \leftarrow \text{Split}(T)$ ;
(5) until  $\text{Trips}$  are feasible
(6)  $S \leftarrow \text{CompleteSolution}(\text{Trips})$ ;
(7) return the feasible solution  $S$  generated;
```

ALGORITHM 1: GRASP for the construction of an initial solution.

a second label N_j for the number of arcs on the shortest path (i.e., the number of second-level vehicles used up to j), sets $N_0 = 0$ at the beginning, and tests N_j before L_j (the cost of the shortest path up to j) to minimize first the number of second-level vehicles.

The adoption of this extension is because the number of routes is often considered as the primary objective of VRP and we also need to obtain feasible solutions concerning the upper bound on the number of the second-level vehicles.

Algorithm 1 gives the framework of the initial solution construction procedure in our GRASP+VND.

Algorithm 2 implements the extended *split* for a random permutation T of all the n_c customers. SA_j records the assigned satellite of the j th customer in T and P_j records the predecessor of this customer on the shortest path. $\mathcal{W} \cdot c(k, V_0)$ represents the additional penalty for each second-level route according to the distance between its assigned satellite and the depot. \mathcal{W} is a nonnegative value, it can be adjusted to change the impact of the first-level routes on the complete initial solution, and, as it increases, the first-level routes are given more importance. It is set to 1 as a default value to obtain a compromised consideration on routes of both echelons.

3.3. Solution Improvement Phase. After an initial feasible solution is generated, an attempt is made to improve it exhaustively. We apply a VND approach to perform the solution improvement procedure until no more improvements can be found. VND is a simplified variant of the VNS heuristic, in which the shaking phase is omitted. Therefore, VND is usually completely deterministic contrary to VNS.

The neighborhoods of our VND can be classified into two types: intersecond-level-route and intrasecond-level-route. Traditional VRP local search operators contain interrout and intraroute operators. Intraroute operators search and improve a single route at a time, while interrout operators deal with several routes simultaneously. When handling the 2E-VRP, we slightly modify the name of this classification for the second-level routes. The existing three interrout neighborhood structures *Shift*, *Swap*, and *Cross* and three intraroute neighborhood structures *2-Opt*, *Or-Opt*, and *Exchange* of Subramanian et al. [26–28] are employed, and we also propose two new neighborhoods *Satellites-Change* and *Satellites-Swap* specially for the 2E-VRP.

Following the representation of Subramanian et al. [26–28], we consider *Shift* ($\lambda, 0$), $\lambda \in \{1, 2, 3\}$, and *Swap* (λ_1, λ_2), $\lambda_1, \lambda_2 \in \{1, 2, 3\}$. We restrict that *Shift* moves at most 3 adjacent customers from a second-level route to another; *Swap*


```

(1)  $L_0 \leftarrow 0; N_0 \leftarrow 0;$ 
(2) for  $i \leftarrow 1$  to  $n_c$  do
(3)    $L_i \leftarrow +\infty; N_i \leftarrow i; SA_i \leftarrow \emptyset;$ 
(4) end for
(5) for  $i \leftarrow 1$  to  $n_c$  do
(6)    $load \leftarrow 0; cost1 \leftarrow 0; j \leftarrow i;$ 
(7)   repeat
(8)      $load \leftarrow load + d(T_i);$ 
(9)     if  $j = i$  then
(10)        $cost1 \leftarrow 0;$ 
(11)     else
(12)        $cost1 \leftarrow cost1 + c(T_{j-1}, T_j);$ 
(13)     end if
(14)      $cost \leftarrow cost1 + \min\{c(k, T_i) + c(T_j, k) + \mathcal{W} \cdot c(k, V_0) \mid k \in V_S\};$  Record the best satellite  $S_i$ ;
(15)     if  $load \leq W_2$  then
(16)       if  $N_j > N_{i-1} + 1$  then
(17)          $N_j \leftarrow N_{i-1} + 1; L_j \leftarrow L_{i-1} + cost;$ 
(18)          $P_j \leftarrow i - 1; SA_j \leftarrow S_i;$ 
(19)       end if
(20)       if  $(N_j = N_{i-1} + 1)$  and  $(L_{i-1} + cost < L_j)$  then
(21)          $L_j \leftarrow L_{i-1} + cost; P_j \leftarrow i - 1; SA_j \leftarrow S_i;$ 
(22)       end if
(23)        $j \leftarrow j + 1;$ 
(24)     end if
(25)   until  $(j > n_c)$  or  $(load > W_2)$ 
(26) end for
(27) Using vectors  $P_j$  and  $SA_j$ , return the resulting set Trips of second-level routes

```

ALGORITHM 2: Procedure *split* for a random permutation T of all n_c customers.

swaps at most 3 adjacent customers from a second-level route with 3 adjacent customers from another. As a result, nine distinct neighborhood structures can be identified, that is, *Shift* (1, 0), *Shift* (2, 0), *Shift* (3, 0), *Swap* (1, 1), *Swap* (1, 2), *Swap* (1, 3), *Swap* (2, 2), *Swap* (2, 3), and *Swap* (3, 3). The neighborhoods of *Swap* (λ_1, λ_2) and *Shift* ($\lambda, 0$) are shown in Figures 2 and 3, respectively. In *Shift* ($\lambda, 0$), λ consecutive customers in a second-level route are moved from a route r_1 to another route r_2 . In *Swap* (λ_1, λ_2), λ_1 consecutive customers from a second-level route r_1 are interchanged with λ_2 consecutive customers from another second-level route r_2 . The *Cross* operator is also called 2-*Opt** in the literature, firstly proposed by Potvin and Rousseau [29], consisting of replacing arc $(i, i + 1)$ from a route r_1 and arc $(j, j + 1)$ from a route r_2 by arcs $(i, j + 1)$ and $(j, i + 1)$ (see Figure 4) or by arcs (i, j) and $(i + 1, j + 1)$ (see Figure 5).

The *Satellites-Change* neighborhood changes the assigned satellite of a second-level route at a time, while *Satellites-Swap* neighborhood swaps the assigned satellites of two distinct second-level routes if they belong to different satellites. Both of them may modify the demands of the satellites and hence may change the total cost of the 2E-VRP. The *Satellites-Change* can be seen as a special case of *Satellites-Swap* when a second-level route swaps its satellite with an empty second-level route. We define both of them as intersecond-level-route neighborhoods. Neighborhoods of *Satellites-Swap* and *Satellites-Change* are shown in Figures 6 and 7, respectively.

Once a new initial feasible solution is generated, an intersecond-level-route move is performed to modify the second-level routes first and then an intrasecond-level-route improvement procedure that uses 2-*Opt*, *Or-Opt*, and *Exchange* neighborhoods sequentially is triggered to improve the modified routes if they are feasible. Details of the three neighborhoods can be found in the survey of Bräysy and Gendreau [30]. 2-*Opt* moves delete two nonadjacent arcs and add two new arcs to generate a new route (see Figure 8); *Or-Opt* moves at most two adjacent customers back and forth in the current second-level route (see Figure 9), while *Exchange* exchanges the positions of two customers or the positions of a customer and two adjacent customers in the same second-level route (see Figure 10). *Or-Opt* used in this paper can be seen as the intraroute version of *Shift* (1, 0) and *Shift* (2, 0), while *Exchange* can be seen as the intraroute version of *Swap* (1, 1) and *Swap* (1, 2).

All the neighborhoods are searched until no more improvements can be obtained, in a first-accept manner, and infeasible move against the capacity of each echelon vehicles is never allowed. When no more moves that improve the current solution can be found in a neighborhood, the search continues with the next neighborhood. VND ends when the current solution is a local optimum with respect to all the applied neighborhoods. The framework of VND [31, 32] is briefly shown in Algorithm 3. Each N_h ($h < h_{\max}$) is actually a combination of an intersecond-level-route neighborhood

```

(1) Define the neighborhood  $N_h = \text{Shift}, \text{Swap}, \text{Cross}, \dots$ 
(2)  $S^* \leftarrow S$ ;
(3) repeat
(4)    $\text{improve\_flag} \leftarrow \text{false}$ ;
(5)    $h \leftarrow 1$ ;
(6)   while  $h \leq h_{\max}$  do
(7)      $S \leftarrow \text{LocalSearch}(S, N_h)$ ;
(8)     if  $\text{Cost}(S) < \text{Cost}(S^*)$  then
(9)        $\text{improve\_flag} \leftarrow \text{true}$ ;
(10)       $S^* \leftarrow S$ ;
(11)       $h \leftarrow h + 1$ ;
(12)    end if
(13)  end while
(14) until  $\text{improve\_flag} = \text{false}$ 
(15) return the local optimum  $S^*$  found;

```

ALGORITHM 3: VND for the improvement of a solution S .

```

(1)  $S^* \leftarrow \emptyset$ ,  $\text{Cost}(S^*) \leftarrow +\infty$ 
(2) for  $i \leftarrow 1$  to  $\max i$  do
(3)    $S \leftarrow \text{GRASP}()$ ;
(4)    $S \leftarrow \text{VND}(S)$ ;
(5)   if  $\text{Cost}(S) < \text{Cost}(S^*)$  then
(6)      $S^* \leftarrow S$ ;
(7)   end if
(8) end for
(9) return the best solution  $S^*$  found;

```

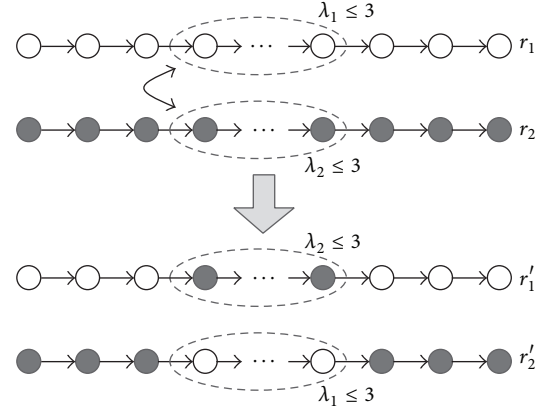
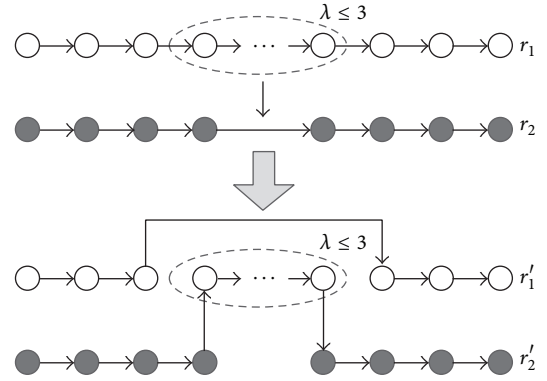
ALGORITHM 4: GRASP+VND for 2E-VRP outline.

and the three intrasecond-level-route neighborhoods. But the neighborhood $N_{h_{\max}}$ placed at the end is an exception and it is only composed of the above three intrasecond-level-route neighborhoods, which are searched circularly until none of them can improve every second-level route of current solution. Every time the solution is modified due to a feasible intersecond-level-route move, the three intrasecond-level-route operators are performed sequentially to improve the newly generated second-level routes.

3.4. Framework of the Resulting GRASP+VND. Our hybrid GRASP+VND heuristic for the 2E-VRP combines a GRASP construction phase with a VND improvement phase. Both of them are iterated $\max i$ times, and the best solution found of all iterations is kept as the final result. An outline of the proposed hybrid algorithm is shown in Algorithm 4. The term S^* stands for the global best solution found.

4. Numeric Verification

We conducted computational study on three sets of benchmark instances in order to assess the proposed hybrid GRASP+VND algorithm with respect to solution quality and computing times. Our hybrid algorithm was coded in C++, compiled by Microsoft Visual C++ 6.0, and run on a

FIGURE 2: $\text{Swap}(\lambda_1, \lambda_2)$ neighborhoods.FIGURE 3: $\text{Shift}(\lambda, 0)$ neighborhoods.

single core of an Intel Pentium Dual-Core E5500 processor (2.8 GHz) and 2 GB of memory.

4.1. Instance Description. The proposed hybrid algorithm was tested on the 2E-VRP benchmark instances Set 2, Set 3, and Set 4. Instances Set 2 and Set 3 were proposed by Perboli et al. [4] and they are based on the following VRP instances of Christofides and Eilon: E-n22-k4, E-n33-k4, and E-n51-k5. The cost matrix of each instance is given by the corresponding VRP instance. The capacity of the first-level vehicles is 2.5 times the capacity of the second. The capacity and the available number of the second-level vehicles are equal to the corresponding VRP instance. The satellites are located at several randomly chosen customers. Instances in this set range between 21 and 50 customers and consider 2 or 4 satellites. Set 4 was proposed by Crainic et al. [33] and contains 54 instances. Each instance has 50 customers and the number of available satellites is 2, 3, or 5. They were generated using three different customer distributions (*Random*, *Centroids*, and *Quadrants*) and three satellite location patterns (*Random*, *Sliced*, and *Forbidden Zone*). A summary of the characteristics of these instances can be found in Hemmelmayr et al. [10].

4.2. Parameter Setting. The nonnegative weight \mathcal{W} in Algorithm 2 is set to 1 as default, which has been determined

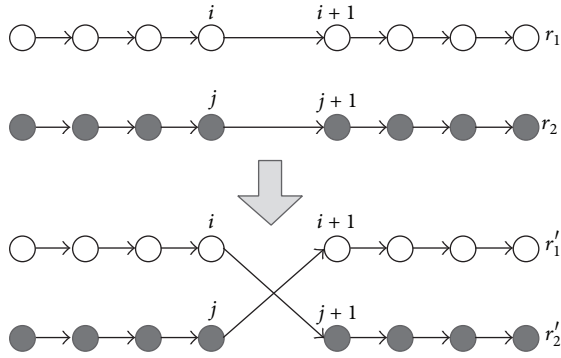


FIGURE 4: Cross neighborhoods type-1.

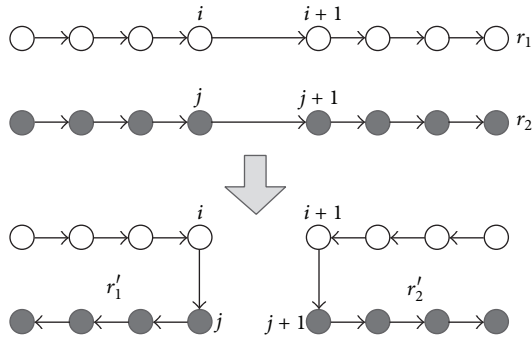


FIGURE 5: Cross neighborhoods type-2.

by experience. The maximum number of iterations *maxi* (i.e., the number of initial feasible solutions generated and improved by VND) is set to 500, and each instance is conducted on five independent runs, as it was done in [10]. We record the best and average solutions, as well as the average time needed to find the best solution of five runs for each instance.

4.3. Computational Results. In this section, we compare the computational results of our hybrid GRASP+VND heuristic with the best existing heuristic method for the 2E-VRP, that is, the ALNS algorithm of Hemmelmayr et al. [10].

The results of instances Set 2, Set 3, and Set 4 obtained by the two compared heuristics are shown in Tables 1, 2, and 3, respectively. The column instance name gives the names of these instances. The column BKS gives the best-known results of these 2E-VRP instances (published in Baldacci et al. [7]). The columns *ALNS* and *GRASP+VND* report the computational results of the two algorithms, respectively. The ALNS algorithm [10] and our GRASP+VND algorithm both report the best and average solutions (containing their percentage deviation from the best-known solutions), as well as the average time (in seconds) needed to find the best solutions of five runs.

The two heuristics were both implemented in C++ but used different workstations. The ALNS heuristic was tested on a single core of an AMD Opteron 275 processor (2.2 GHz). In order to make a fair comparison, we choose the running times of ALNS as a benchmark and scale the running times

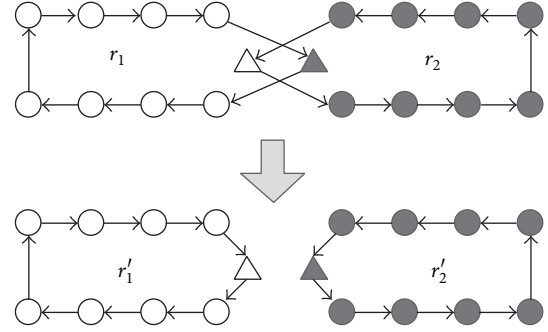


FIGURE 6: Satellites-Swap neighborhoods.

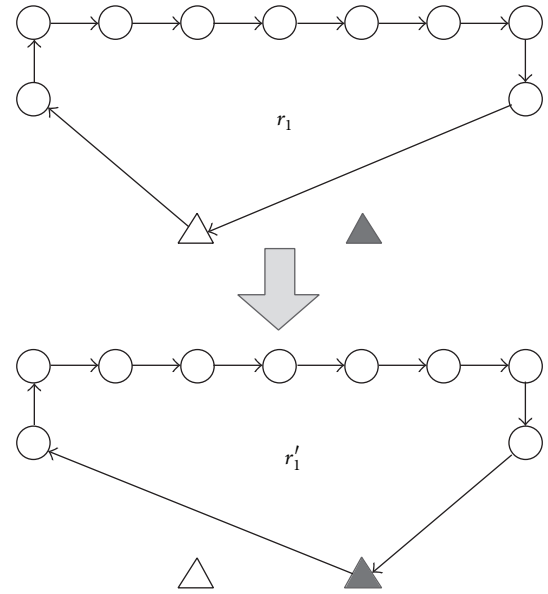


FIGURE 7: Satellites-Change neighborhoods.

of our GRASP+VND according to the CPU performances reported at http://www.cpubenchmark.net/cpu_list.php. Our machine is 1.34 times faster than the AMD Opteron 275 processor (2.2 GHz) used by Hemmelmayr et al. [10]; therefore, our running times are multiplied by 1.34. The times shown in Tables 1, 2, and 3 are already the scaled value of actual running times. Values in bold fonts correspond to those that our GRASP+VND outperforms the ALNS, while those italic mean that our algorithm is worse than the ALNS.

5. Discussion

As seen from Tables 1, 2, and 3, our GRASP+VND and the ALNS can both find the best-known solutions for all the 39 instances in Set 2 and Set 3, and the ALNS can find the best-known 44 solutions for instances in Set 4, while the GRASP+VND can find 52; besides, there are ten better results in the best values found by our GRASP+VND. In terms of the average solutions, the two heuristics obtained the same results in Set 2 and Set 3, but we can see from Table 3 that our GRASP+VND is much better than the ALNS. In regard to

TABLE 1: Computational results comparison for Set 2 for the 2E-VRP.

Number	Instance name	n_s	BKS [7]	ALNS					GRASP+VND				
				Best	% dev.	Average	% dev.	Time (s)	Best	% dev.	Average	% dev.	Time (s)
1	E-n22-k4-s6-17	2	417.07	417.07	0.00	417.07	0.00	0	417.07	0.00	417.07	0.00	0
2	E-n22-k4-s8-14	2	384.96	384.96	0.00	384.96	0.00	0	384.96	0.00	384.96	0.00	0
3	E-n22-k4-s9-19	2	470.60	470.60	0.00	470.60	0.00	0	470.60	0.00	470.60	0.00	0
4	E-n22-k4-s10-14	2	371.50	371.50	0.00	371.50	0.00	0	371.50	0.00	371.50	0.00	0
5	E-n22-k4-s11-12	2	427.22	427.22	0.00	427.22	0.00	0	427.22	0.00	427.22	0.00	0
6	E-n22-k4-s12-16	2	392.78	392.78	0.00	392.78	0.00	0	392.78	0.00	392.78	0.00	0
7	E-n33-k4-s1-9	2	730.16	730.16	0.00	730.16	0.00	0	730.16	0.00	730.16	0.00	0
8	E-n33-k4-s2-13	2	714.63	714.63	0.00	714.63	0.00	0	714.63	0.00	714.63	0.00	0
9	E-n33-k4-s3-17	2	707.48	707.48	0.00	707.48	0.00	0	707.48	0.00	707.48	0.00	0
10	E-n33-k4-s4-5	2	778.74	778.74	0.00	778.74	0.00	3	778.74	0.00	778.74	0.00	1
11	E-n33-k4-s7-25	2	756.85	756.85	0.00	756.85	0.00	0	756.85	0.00	756.85	0.00	0
12	E-n33-k4-s14-22	2	779.05	779.05	0.00	779.05	0.00	0	779.05	0.00	779.05	0.00	0
13	E-n51-k5-s2-17	2	597.49	597.49	0.00	597.49	0.00	7	597.49	0.00	597.49	0.00	1
14	E-n51-k5-s4-46	2	530.76	530.76	0.00	530.76	0.00	0	530.76	0.00	530.76	0.00	0
15	E-n51-k5-s6-12	2	554.81	554.81	0.00	554.81	0.00	2	554.81	0.00	554.81	0.00	2
16	E-n51-k5-s11-19	2	581.64	581.64	0.00	581.64	0.00	6	581.64	0.00	581.64	0.00	5
17	E-n51-k5-s27-47	2	538.22	538.22	0.00	538.22	0.00	1	538.22	0.00	538.22	0.00	1
18	E-n51-k5-s32-37	2	552.28	552.28	0.00	552.28	0.00	1	552.28	0.00	552.28	0.00	1
19	E-n51-k5-s2-4-17-46	4	530.76	530.76	0.00	530.76	0.00	1	530.76	0.00	530.76	0.00	1
20	E-n51-k5-s6-12-32-37	4	531.92	531.92	0.00	531.92	0.00	0	531.92	0.00	531.92	0.00	1
21	E-n51-k5-s11-19-27-47	4	527.63	527.63	0.00	527.63	0.00	1	527.63	0.00	527.63	0.00	1
Average			565.55	565.55	0.00	565.55	0.00	1	565.55	0.00	565.55	0.00	0.67

TABLE 2: Computational results comparison for Set 3 for the 2E-VRP.

Number	Instance name	n_s	BKS [7]	ALNS					GRASP+VND				
				Best	% dev.	Average	% dev.	Time (s)	Best	% dev.	Average	% dev.	Time (s)
1	E-n22-k4-s13-14	2	526.15	526.15	0.00	526.15	0.00	0	526.15	0.00	526.15	0.00	0
2	E-n22-k4-s13-16	2	521.09	521.09	0.00	521.09	0.00	0	521.09	0.00	521.09	0.00	0
3	E-n22-k4-s13-17	2	496.38	496.38	0.00	496.38	0.00	0	496.38	0.00	496.38	0.00	0
4	E-n22-k4-s14-19	2	498.80	498.80	0.00	498.80	0.00	0	498.80	0.00	498.80	0.00	0
5	E-n22-k4-s17-19	2	512.81	512.81	0.00	512.81	0.00	0	512.81	0.00	512.81	0.00	0
6	E-n22-k4-s19-21	2	520.42	520.42	0.00	520.42	0.00	0	520.42	0.00	520.42	0.00	0
7	E-n33-k4-s16-22	2	672.17	672.17	0.00	672.17	0.00	3	672.17	0.00	672.17	0.00	0
8	E-n33-k4-s16-24	2	666.02	666.02	0.00	666.02	0.00	0	666.02	0.00	666.02	0.00	0
9	E-n33-k4-s19-26	2	680.37	680.37	0.00	680.37	0.00	0	680.37	0.00	680.37	0.00	0
10	E-n33-k4-s22-26	2	680.37	680.37	0.00	680.37	0.00	0	680.37	0.00	680.37	0.00	0
11	E-n33-k4-s24-28	2	670.43	670.43	0.00	670.43	0.00	0	670.43	0.00	670.43	0.00	0
12	E-n33-k4-s25-28	2	650.58	650.58	0.00	650.58	0.00	0	650.58	0.00	650.58	0.00	0
13	E-n51-k5-s12-18	2	690.59	690.59	0.00	690.59	0.00	4	690.59	0.00	690.59	0.00	1
14	E-n51-k5-s12-41	2	683.05	683.05	0.00	683.05	0.00	38	683.05	0.00	683.05	0.00	1
15	E-n51-k5-s12-43	2	710.41	710.41	0.00	710.41	0.00	1	710.41	0.00	710.41	0.00	1
16	E-n51-k5-s39-41	2	728.54	728.54	0.00	728.54	0.00	18	728.54	0.00	728.54	0.00	6
17	E-n51-k5-s40-41	2	723.75	723.75	0.00	723.75	0.00	17	723.75	0.00	723.75	0.00	4
18	E-n51-k5-s40-43	2	752.15	752.15	0.00	752.15	0.00	15	752.15	0.00	752.15	0.00	12
Average			632.45	632.45	0.00	632.45	0.00	5	632.45	0.00	632.45	0.00	1.44

TABLE 3: Computational results comparison for Set 4 for the 2E-VRP.

Number	Instance name	n_s	BKS [7]	ALNS					GRASP+VND				
				Best	% dev.	Average	% dev.	Time (s)	Best	% dev.	Average	% dev.	Time (s)
1	Instance50-s2-01	2	1569.42	1569.42	0.00	1569.42	0.00	6	1569.42	0.00	1569.42	0.00	2
2	Instance50-s2-02	2	1438.33	1438.33	0.00	1441.02	0.19	43	1438.33	0.00	1438.33	0.00	53
3	Instance50-s2-03	2	1570.43	1570.43	0.00	1570.43	0.00	3	1570.43	0.00	1570.43	0.00	2
4	Instance50-s2-04	2	1424.04	1424.04	0.00	1424.04	0.00	11	1424.04	0.00	1429.04	0.35	98
5	Instance50-s2-05	2	2193.52	2194.11	0.03	2194.11	0.03	63	2193.52	0.00	2193.95	0.02	46
6	Instance50-s2-06	2	1279.87	1279.87	0.00	1279.87	0.00	2	1279.87	0.00	1279.87	0.00	1
7	Instance50-s2-07	2	1408.57	1458.63	3.55	1458.63	3.55	6	1408.57	0.00	1408.57	0.00	22
8	Instance50-s2-08	2	1360.32	1360.32	0.00	1360.32	0.00	5	1360.32	0.00	1360.32	0.00	5
9	Instance50-s2-09	2	1403.53	1450.27	3.33	1450.27	3.33	46	1403.53	0.00	1403.53	0.00	6
10	Instance50-s2-10	2	1360.56	1360.56	0.00	1360.56	0.00	1	1360.56	0.00	1360.56	0.00	2
11	Instance50-s2-11	2	2047.46	2059.88	0.61	2059.88	0.61	101	2059.41	0.58	2059.41	0.58	5
12	Instance50-s2-12	2	1209.42	1209.42	0.00	1209.42	0.00	44	1209.42	0.00	1209.42	0.00	8
13	Instance50-s2-13	2	1450.93	1481.83	2.13	1481.83	2.13	25	1450.93	0.00	1450.93	0.00	3
14	Instance50-s2-14	2	1393.61	1393.61	0.00	1393.61	0.00	6	1393.61	0.00	1393.61	0.00	2
15	Instance50-s2-15	2	1466.83	1489.94	1.58	1489.94	1.58	9	1466.83	0.00	1466.83	0.00	1
16	Instance50-s2-16	2	1387.83	1387.83	0.00	1387.83	0.00	6	1387.83	0.00	1387.83	0.00	8
17	Instance50-s2-17	2	2088.49	2088.49	0.00	2088.49	0.00	165	2088.49	0.00	2088.49	0.00	36
18	Instance50-s2-18	2	1227.61	1227.61	0.00	1227.61	0.00	3	1227.61	0.00	1227.61	0.00	2
19	Instance50-s3-19	3	1546.28	1546.28	0.00	1546.28	0.00	25	1546.28	0.00	1546.28	0.00	34
20	Instance50-s3-20	3	1272.97	1272.97	0.00	1272.97	0.00	12	1272.97	0.00	1272.97	0.00	76
21	Instance50-s3-21	3	1577.82	1577.82	0.00	1577.82	0.00	61	1577.82	0.00	1577.82	0.00	25
22	Instance50-s3-22	3	1281.83	1281.83	0.00	1281.83	0.00	2	1281.83	0.00	1281.83	0.00	37
23	Instance50-s3-23	3	1652.98	1652.98	0.00	1652.98	0.00	5	1652.98	0.00	1652.98	0.00	4
24	Instance50-s3-24	3	1282.68	1282.68	0.00	1282.68	0.00	2	1282.68	0.00	1282.68	0.00	2
25	Instance50-s3-25	3	1408.57	1440.68	2.28	1440.84	2.29	53	1408.57	0.00	1408.57	0.00	23
26	Instance50-s3-26	3	1167.46	1167.46	0.00	1167.46	0.00	0	1167.46	0.00	1167.46	0.00	9
27	Instance50-s3-27	3	1444.51	1444.51	0.00	1447.79	0.23	12	1444.51	0.00	1454.63	0.70	52
28	Instance50-s3-28	3	1210.44	1210.44	0.00	1210.44	0.00	7	1210.44	0.00	1210.44	0.00	3
29	Instance50-s3-29	3	1552.66	1559.82	0.46	1561.81	0.59	102	1552.66	0.00	1555.56	0.19	40
30	Instance50-s3-30	3	1211.59	1211.59	0.00	1211.59	0.00	5	1211.59	0.00	1211.59	0.00	2
31	Instance50-s3-31	3	1440.86	1440.86	0.00	1440.86	0.00	37	1440.86	0.00	1441.07	0.01	39
32	Instance50-s3-32	3	1199.00	1199.00	0.00	1199.00	0.00	11	1199.00	0.00	1199.00	0.00	13
33	Instance50-s3-33	3	1478.86	1478.86	0.00	1478.86	0.00	16	1478.86	0.00	1478.86	0.00	19
34	Instance50-s3-34	3	1233.92	1233.92	0.00	1233.92	0.00	4	1233.92	0.00	1233.92	0.00	15
35	Instance50-s3-35	3	1570.72	1570.72	0.00	1570.80	0.01	116	1570.72	0.00	1570.72	0.00	6
36	Instance50-s3-36	3	1228.89	1228.89	0.00	1228.89	0.00	6	1228.89	0.00	1228.89	0.00	3
37	Instance50-s5-37	5	1528.73	1528.73	0.00	1528.81	0.01	55	1528.73	0.00	1528.98	0.02	58
38	Instance50-s5-38	5	1163.07	1163.07	0.00	1163.07	0.00	15	1163.07	0.00	1163.07	0.00	2
39	Instance50-s5-39	5	1520.92	1520.92	0.00	1520.92	0.00	33	1520.92	0.00	1520.92	0.00	21
40	Instance50-s5-40	5	1163.04	1163.04	0.00	1165.24	0.19	20	1163.04	0.00	1163.04	0.00	9
41	Instance50-s5-41	5	1652.98	1652.98	0.00	1652.98	0.00	12	1652.98	0.00	1652.98	0.00	12
42	Instance50-s5-42	5	1190.17	1190.17	0.00	1190.17	0.00	31	1190.17	0.00	1190.17	0.00	95
43	Instance50-s5-43	5	1406.11	1406.11	0.00	1408.95	0.20	60	1406.11	0.00	1406.11	0.00	32
44	Instance50-s5-44	5	1035.03	1035.03	0.00	1035.32	0.03	30	1035.03	0.00	1035.03	0.00	9
45	Instance50-s5-45	5	1401.87	1403.10	0.09	1406.43	0.33	104	1402.03	0.01	1402.03	0.01	36
46	Instance50-s5-46	5	1058.11	1058.11	0.00	1058.97	0.08	17	1058.11	0.00	1058.11	0.00	9
47	Instance50-s5-47	5	1552.66	1559.82	0.46	1564.41	0.76	103	1552.66	0.00	1557.04	0.28	33
48	Instance50-s5-48	5	1074.50	1074.50	0.00	1074.50	0.00	2	1074.50	0.00	1074.50	0.00	1
49	Instance50-s5-49	5	1434.88	1434.88	0.00	1435.28	0.03	81	1434.88	0.00	1434.88	0.00	51
50	Instance50-s5-50	5	1065.25	1065.25	0.00	1065.25	0.00	16	1065.25	0.00	1065.25	0.00	3
51	Instance50-s5-51	5	1387.51	1387.51	0.00	1387.72	0.02	46	1387.51	0.00	1387.51	0.00	4
52	Instance50-s5-52	5	1103.42	1103.42	0.00	1103.76	0.03	47	1103.42	0.00	1103.42	0.00	30
53	Instance50-s5-53	5	1545.73	1545.73	0.00	1545.73	0.00	37	1545.73	0.00	1545.73	0.00	6
54	Instance50-s5-54	5	1113.62	1113.62	0.00	1113.62	0.00	2	1113.62	0.00	1113.62	0.00	16
Average			1397.04	1400.96	0.28	1401.39	0.31	32	1397.27	0.02	1397.70	0.05	21

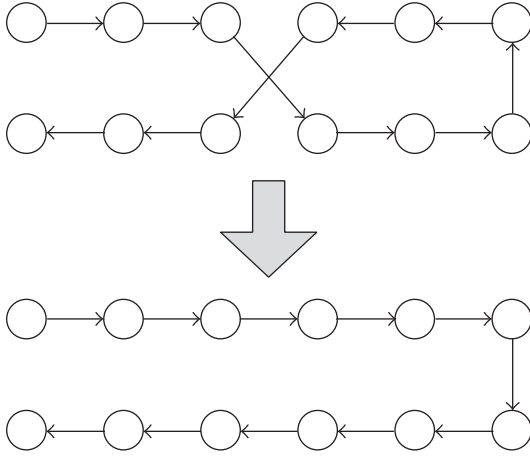


FIGURE 8: 2-Opt neighborhoods.

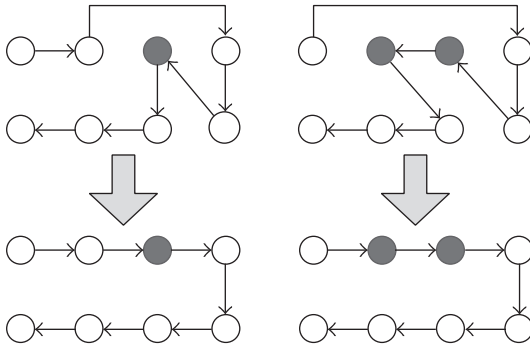


FIGURE 9: Two kinds of Or-Opt neighborhoods.

the deviations of the solutions from the best-known results, we can find that our GRASP+VND is as good as the ALNS in Set 2 and Set 3 but outperforms the ALNS in Set 4. With respect to the running times, our GRASP+VND is as fast as the ALNS in Set 2, a little faster than the ALNS in Set 3 and much faster in Set 4; besides, the running times of our GRASP+VND are limited and reasonable. Our GRASP+VND adopted 8 kinds of local search operators, while the ALNS used 8 kinds of destroy operators, 5 kinds of repair operators, and 5 kinds of local search operators; besides, our GRASP+VND used only 2 parameters, but the ALNS employed 9 parameters, so the GRASP+VND is much easier to implement and tune. From the comparisons above, we can retrieve that our GRASP+VND is both effective and efficient and outperforms the best existing heuristics for the 2E-VRP.

6. Conclusion

This paper has presented a very simple hybrid GRASP+VND heuristic to address the two-echelon vehicle routing problem (2E-VRP), a newly defined multiechelon variant of the classical vehicle routing problem (VRP). The heuristic is composed of a GRASP construction phase (embedding an extended split

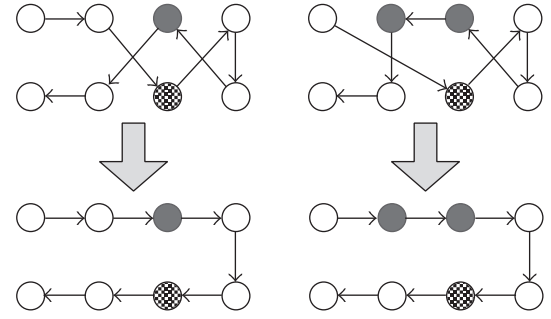


FIGURE 10: Two kinds of Exchange neighborhoods.

algorithm) to generate feasible and relatively good solutions and a VND phase to improve them.

Computational experiments on three sets of benchmark instances from the literature showed the effectiveness and efficiency of the proposed algorithm, and it outperformed the best existing heuristic methods for the 2E-VRP in both solutions quality and computing times. Moreover, the implementation of the hybrid heuristic is much easier than other heuristics. As a result, the proposed hybrid heuristic is more suitable for handling practical 2E-VRP and its relative variants arising in city logistics.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgments

This work is supported by the National Natural Science Foundation of China under Grant no. 71090404. The authors are extremely grateful to the anonymous referees for their helpful comments.

References

- [1] E. Taniguchi and R. G. Thompson, "Modeling city logistics," *Transportation Research Record*, vol. 1790, no. 1, pp. 45–51, 2002.
- [2] T. G. Crainic, N. Ricciardi, and G. Storchi, "Advanced freight transportation systems for congested urban areas," *Transportation Research C: Emerging Technologies*, vol. 12, no. 2, pp. 119–137, 2004.
- [3] T. G. Crainic, N. Ricciardi, and G. Storchi, "Models for evaluating and planning city logistics systems," *Transportation Science*, vol. 43, no. 4, pp. 432–454, 2009.
- [4] G. Perboli, R. Tadei, and D. Vigo, "The two-echelon capacitated vehicle routing problem: models and math-based heuristics," *Transportation Science*, vol. 45, no. 3, pp. 364–380, 2011.
- [5] G. Perboli and R. Tadei, "New families of valid inequalities for the two-echelon vehicle routing problem," *Electronic Notes in Discrete Mathematics*, vol. 36, pp. 639–646, 2010, Proceedings of the International Symposium on Combinatorial Optimization (ISCO '10).
- [6] M. Jepsen, S. Spoorendonk, and S. Ropke, "A branch-and-cut algorithm for the symmetric two-echelon capacitated vehicle

- routing problem,” *Transportation Science*, vol. 47, no. 1, pp. 23–37, 2013.
- [7] R. Baldacci, A. Mingozzi, R. Roberti, and R. W. Calvo, “An exact algorithm for the two-echelon capacitated vehicle routing problem,” *Operations Research*, vol. 61, no. 2, pp. 298–314, 2013.
 - [8] T. G. Crainic, S. Mancini, G. Perboli, and R. Tadei, “Multi-start heuristics for the two-echelon vehicle routing problem,” in *Evolutionary Computation in Combinatorial Optimization*, P. Merz and J. K. Hao, Eds., vol. 6622 of *Lecture Notes in Computer Science*, pp. 179–190, Springer, Berlin, Germany, 2011.
 - [9] W. Meihua, T. Xuhong, C. Shan, and W. Shumin, “Hybrid ant colony optimization algorithm for two echelon vehicle routing problem,” *Procedia Engineering*, vol. 15, pp. 3361–3365, 2011.
 - [10] V. C. Hemmelmayr, J. Cordeau, and T. G. Crainic, “An adaptive large neighborhood search heuristic for two-echelon vehicle routing problems arising in city logistics,” *Computers and Operations Research*, vol. 39, no. 12, pp. 3215–3228, 2012.
 - [11] T. G. Crainic, S. Mancini, G. Perboli, and R. Tadei, “Grasp with path relinking for the two-echelon vehicle routing problem,” in *Advances in Metaheuristics*, L. D. Gaspero, A. Schaerf, and T. Stutzle, Eds., vol. 53 of *Operations Research/Computer Science Interfaces Series*, pp. 113–125, Springer, New York, NY, USA, 2013.
 - [12] T. A. Feo and M. G. C. Resende, “Greedy randomized adaptive search procedures,” *Journal of Global Optimization*, vol. 6, no. 2, pp. 109–133, 1995.
 - [13] M. G. Resende and C. C. Ribeiro, “Greedy randomized adaptive search procedures: advances, hybridizations, and applications,” in *Handbook of Metaheuristics*, M. Gendreau and J. Y. Potvin, Eds., vol. 146 of *International Series in Operations Research & Management Science*, pp. 283–319, Springer US, New York, NY, USA, 2nd edition, 2010.
 - [14] N. Mladenović and P. Hansen, “Variable neighborhood search,” *Computers and Operations Research*, vol. 24, no. 11, pp. 1097–1100, 1997.
 - [15] P. Hansen, N. Mladenovic, J. Brimberg, and J. Perez, “Variable neighborhood search,” in *Handbook of Metaheuristics*, M. Gendreau and J. Y. Potvin, Eds., pp. 61–86, Springer US, New York, NY, USA, 2nd edition, 2010.
 - [16] H. Hernández-Pérez, I. Rodríguez-Martín, and J. J. Salazar-González, “A hybrid GRASP/VND heuristic for the one-commodity pickup-and-delivery traveling salesman problem,” *Computers and Operations Research*, vol. 36, no. 5, pp. 1639–1645, 2009.
 - [17] J. G. Villegas, C. Prins, C. Prodhon, A. L. Medaglia, and N. Velasco, “GRASP/VND and multi-start evolutionary local search for the single truck and trailer routing problem with satellite depots,” *Engineering Applications of Artificial Intelligence*, vol. 23, no. 5, pp. 780–794, 2010.
 - [18] A. Salehipour, K. Sörensen, P. Goos, and O. Bräysy, “Efficient GRASP+VND and GRASP+VNS metaheuristics for the traveling repairman problem,” *4OR*, vol. 9, no. 2, pp. 189–209, 2011.
 - [19] F. Parreño, R. Alvarez-Valdes, J. F. Oliveira, and J. M. Tamarit, “A hybrid GRASP/VND algorithm for two- and three-dimensional bin packing,” *Annals of Operations Research*, vol. 179, no. 1, pp. 203–220, 2010.
 - [20] P. Schittekat, J. Kinable, K. Sorensen, M. Sevaux, F. Spieksma, and J. Springael, “A metaheuristic for the school bus routing problem with bus stop selection,” *The European Journal of Operational Research*, vol. 229, no. 2, pp. 518–528, 2013.
 - [21] V. Nguyen, C. Prins, and C. Prodhon, “Solving the two-echelon location routing problem by a GRASP reinforced by a learning process and path relinking,” *The European Journal of Operational Research*, vol. 216, no. 1, pp. 113–126, 2012.
 - [22] C. Archetti, M. G. Speranza, and A. Hertz, “A tabu search algorithm for the split delivery vehicle routing problem,” *Transportation Science*, vol. 40, no. 1, pp. 64–73, 2006.
 - [23] 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.
 - [24] J. Beasley, “Route first-Cluster second methods for vehicle routing,” *Omega*, vol. 11, no. 4, pp. 403–408, 1983.
 - [25] C. Prins, “A simple and effective evolutionary algorithm for the vehicle routing problem,” *Computers and Operations Research*, vol. 31, no. 12, pp. 1985–2002, 2004.
 - [26] A. Subramanian, L. M. A. Drummond, C. Bentes, L. S. Ochi, and R. Farias, “A parallel heuristic for the vehicle routing problem with simultaneous pickup and delivery,” *Computers and Operations Research*, vol. 37, no. 11, pp. 1899–1911, 2010.
 - [27] A. Subramanian, P. H. V. Penna, E. Uchoa, and L. S. Ochi, “A hybrid algorithm for the heterogeneous fleet vehicle routing problem,” *The European Journal of Operational Research*, vol. 221, no. 2, pp. 285–295, 2012.
 - [28] A. Subramanian, P. H. V. Penna, E. Uchoa, and L. S. Ochi, “A hybrid algorithm for a class of vehicle routing problems,” *The European Journal of Operational Research*, vol. 40, no. 10, pp. 2519–2531, 2013.
 - [29] J. Potvin and J. Rousseau, “Exchange heuristic for routeing problems with time windows,” *Journal of the Operational Research Society*, vol. 46, no. 12, pp. 1433–1446, 1995.
 - [30] O. Bräysy and M. Gendreau, “Vehicle routing problem with time windows, part I: route construction and local search algorithms,” *Transportation Science*, vol. 39, no. 1, pp. 104–118, 2005.
 - [31] V. P. Nguyen, C. Prins, and C. Prodhon, “A multi-start evolutionary local search for the two-echelon location routing problem,” in *Hybrid Metaheuristics*, M. J. Blesa, C. Blum, G. Raidl, A. Roli, and M. Sampels, Eds., vol. 6373 of *Lecture Notes in Computer Science*, pp. 88–102, Springer, Berlin, Germany, 2010.
 - [32] P. Chen, H. Huang, and X. Dong, “Iterated variable neighborhood descent algorithm for the capacitated vehicle routing problem,” *Expert Systems with Applications*, vol. 37, no. 2, pp. 1620–1627, 2010.
 - [33] T. G. Crainic, G. Perboli, S. Mancini, and R. Tadei, “Two-echelon vehicle routing problem: a satellite location analysis,” *Procedia—Social and Behavioral Sciences*, vol. 2, no. 3, pp. 5944–5955, 2010, Proceedings of the 6th International Conference on City Logistics.

