Two-echelon vehicle routing problem with time windows and simultaneous pickup and delivery

Hang Zhou · Hu Qin · Zizhen Zhang · Jiliu Li

Optimization

Abstract
This paper proposes a tabu search algorithm for the two-echelon vehicle routing problem with time windows and simultaneous pickup and delivery (2E-VRPTWSPD), which is a new variant of the two-echelon vehicle routing problem. 2E-VRPTWSPD involves three stages of routing: (1) the first-echelon vehicles start from the depot to deliver cargoes to satellites; (2) the second-echelon vehicles start from satellites to serve customers within time windows in a simultaneously pickup and delivery manner and finally return to their satellites with pickup cargoes; (3) the first-echelon vehicles start from the depot to collect cargoes on satellites. To solve this problem, we formulate it with a mathematical model. Then, we implement a variable neighborhood tabu search algorithm with a tailored solution representation to solve large-scale instances. Dummy satellites time windows are applied in our algorithm to speed up the search. Finally, we generate two benchmark instance sets to analyze the performance of our algorithm. Computational results show that our algorithm can obtain the optimal results for 10 out of 11 small-scale instances and produce promising solutions for 200-customer instances within a few minutes. Additional experiments indicate that the usage of dummy satellite time windows can save 19% computation time on average.

Keywords Two-echelon vehicle routing problem · Time windows · Pickup and delivery · Tabu search

1 Introduction
In recent years, the transportation cost in logistics has increased rapidly. The vehicle routing problem (VRP), which aims to determine the best routing plan for vehicles to serve a set of customers, is widely studied to cope with this situation. The two-echelon vehicle routing problem (2E-VRP) is a well-known variant of the classic VRP. It involves a two-echelon distribution network with a CD (i.e., central depot), a set of satellites, and a set of end customers. In 2E-VRP, vehicles are divided into two types; each has a specific capacity. Delivery tasks in the first level are usually accomplished by first-echelon vehicles with a large capacity, while those in the second level are usually accomplished by second-echelon vehicles with a small capacity. The freight is first transported from the depot to satellites by first-echelon vehicles. Then, the cargoes on the first-echelon vehicles are loaded to the second-echelon vehicles at satellites. Finally, the freight is transported to customers by second-echelon vehicles. Each customer has a demand and must be served exactly once. The objective is to minimize the sum of the total routing cost of the two vehicle types. Figure 1 illustrates an example of 2E-VRP, which includes a depot, three satellites, and nine customers.

In real-world city logistics, more constraints need to be considered. For example, time window constraints are always taken into account by some activities like take-out service or delivery service for some special food which needs fresh-keeping. Simultaneous pickup and delivery is another important VRP characteristic, which allows the pickup and delivery of cargoes for a customer at the same time.

In this study, we consider a two-echelon vehicle routing problem with time windows and simultaneous pickup and delivery (2E-VRPTWSPD), which is a new variant of 2E-VRP. This problem can easily be applied to some real-world circumstances. For example, let us consider the delivery of some medical supplies in a multimodal urban distribution. The first-echelon vehicles serve between cities, and the second-echelon vehicles are city freighters who directly visit...
customers’ houses. Customers may use some medical products in their own houses for convenience. Some parts of the medical product, such as the wrapper for liquid or the used syringe needle, need to be called back immediately. This is because the abandon wrappers with remnant medicine will pollute the environment, or even be illicitly purchased by drug traffickers and then cause harms. In sum, the reverse logistic plays an important role in the delivery system. Many reverse logistic examples in multimodal distribution are the applications of 2E-VRPTWSPD.

We consider several practical features of 2E-VRPTWSPD. First, the called-back part is lighter than the original product. This means that the pickup demand in each customer is less than the delivery demand. This is a practical assumption, although it is easy to adjust our method to adapt to the general problem that the pickup demand may be larger. Second, we find that in most literature on 2E-VRPTW, the service duration in satellites is always fixed. In practice, however, the time used to transfer cargoes depends on the quality of cargoes. In this paper, we assume that the service time in satellites is proportionally correlated with the quantity of cargoes.

Our work can be summarized as follows. We first introduce 2E-VRPTWSPD, which is a new variant of 2E-VRP. We then propose a mixed integer programming model to formulate the problem. Next, we provide a heuristic algorithm that includes a variable neighborhood tabu search phase to solve the problem. The model formulations and the heuristic algorithm are tested by the instances we generated.

For the remaining parts of the paper, Sect. 2 reviews some studies on the related work. Section 3 formally defines 2E-VRPTWSPD and introduces a mixed-integer linear programming model. Section 4 presents our solution approach. Section 5 describes the test instances and the results, and analyzes the speciality of the problem and algorithm. Section 6 gives conclusions and future directions on this subject.

2 Literature review

2E-VRPTWSPD is an extension of 2E-VRP by further considering time window constraints and simultaneous pickup and delivery. To our best knowledge, this is the first study that considers both features in 2E-VRP. In this section, we review two closely related problems: 2E-VRP and VRPSPD.

2E-VRP was studied since the pioneer work of Crainic et al. (2009). The authors proposed a general problem under the name two-echelon, synchronized, scheduled, multidepot, multiple-tour, heterogeneous VRPTW (2SS-MDMT-VRPTW), and 2E-VRP is a special case of this problem. The formal description and model of 2E-VRP were introduced by Perboli and Tadei (2010) and Perboli et al. (2011). Cuda et al. (2015) published a survey on 2E-VRP, which summarized the early development of 2E-VRP. We refer readers to this survey for further details. Later, several meta-heuristics were proposed to solve 2E-VRP. For example, Li et al. (2019) designed a two-stage heuristic algorithm. The first stage constructs an initial solution through the CW algorithm, and the second stage improves the solution by a variable neighborhood search (VNS). Enthoven et al. (2020) designed a tailored adaptive large neighborhood search (ALNS) heuristic for two-echelon vehicle routing problem with covering...
options (2E-VRP-CO), which considered the applications of 2E-VRP in e-commerce. Recently, Kitjacharoenchai et al. (2020) discovered an application of 2E-VRP in truck-drone delivery system. They proposed an ALNS heuristic to solve it. Breunig et al. (2019) considered a large neighborhood search (LNS) metaheuristic for the electric two-echelon vehicle routing problem (E2E-VRP), where the second-echelon customers are served by electric vehicles. Some specially designed operators, such as close satellite destroy operators, are used in the LNS. More literature about E2E-VRP can be found in the review paper (Qin et al. 2021). Exact algorithms are another mainstream for solving 2E-VRP. Santos et al. (2015) introduced a branch-and-price-and-cut (BPC) algorithm for 2E-VRP, in which a reformulation based on q-routes and several classes of valid inequalities are used. The algorithm obtained 10 new optimality certificates and several lower bounds compared with the previous branch-and-price (BP) algorithm in Santos et al. (2013). BPC algorithm was also used in Marques et al. (2020), which can solve up to 200 customers and 10 satellites instances. Liu et al. (2018) proposed a new variant of 2E-VRP, named two-echelon capacitated vehicle routing problem with grouping constraints (2E-VRP-GC), and a branch-and-cut algorithm is adopted.

It is worth noting that only a few researchers considered the time window constraints in 2E-VRP. Dellaert et al. (2019, 2021) proposed branch-and-price-based algorithms for 2E-VRPTW with multiple depots. Dellaert et al. (2019) compared the effectiveness of two set partitioning formulations. The first one is named 1-path formulation, which is similar to the traditional VRP model. The second one is 2-path formulation, which separates the first echelon routes and the second echelon routes, thus simplifying the subproblem. Experiments show that the second formulation has a better results. Dellaert et al. (2021) extended 2E-VRPTW to a multimmodity type, which is named multimmodity two-echelon capacitated vehicle routing problem with time windows (MC-2E-VRPTW). The results show that most instances with 50 customers can be solved to optimality, and most instances with 100 customers and 2 depots can be solved close to optimality within 3 h. Li et al. (2020) introduced a two-echelon vehicle routing problem with time windows and mobile satellites (2E-VRP-TM). In this problem, each vehicle carries several drones in the first-echelon route, while the drones can be launched from the vehicle to serve customers in the second-echelon route. The authors proposed an ALNS heuristic to solve the problem.

2E-VRPTWSPD can be treated as VRPSPD if the satellites and time windows are excluded. VRPSPD was first introduced by Min (1989). Then, many heuristic and metaheuristic approaches were proposed. We briefly introduced them as follows. Mu et al. (2016) introduced a parallel simulated annealing (SA). Kalayci and Kaya (2016) proposed a hybrid meta-heuristic algorithm based on the ant colony system (ACS) and variable neighborhood search (VNS). Park et al. (2021) designed a genetic algorithm (GA). Liu et al. (2021) devised a hybrid algorithm of the ACS method. Considering VRPSPD with time windows constraints, Wang and Chen (2012) proposed GA. Liu et al. (2013) presented both a GA and a TS method. Wang et al. (2015) defined a parallel SA algorithm with a special insertion method. For more related works on VRPSPD, we refer readers to the review paper (Çağrı Koç et al. 2020).

To date, there are merely studies combined 2E-VRP with VRPSPD (Belgin et al. 2018). The authors introduced a two-echelon vehicle routing problem with simultaneous pickup and delivery (2E-VRPSPD) and proposed a node-based mathematical model and a hybrid heuristic approach based on variable neighborhood descent (VND) and local search (LS). However, this work did not consider the time window constraints, which are common restrictions in reality. In addition, we find that there exists very limited up-to-date research considering tabu search (TS), which has strong convergence and has shown its great success in dealing with different VRPs. Thus, this paper considers a TS-based algorithm and adopts some state-of-the-art techniques to investigate 2E-VRPTWSPD.

### 3 Problem descriptions

2E-VRPTWSPD is defined on a directed graph $G = (V, A)$ with vertex set $V = V_0 \cup V_S \cup V_C$, where $V_0$ is the set of depot location (only one depot in this set), $V_S$ is the set of satellite locations, and $V_C$ is the set of customer locations. The arc set $A$ consists of two different sets $A_1$ and $A_2$, where $A_1 = \{(i, j) | i, j \in V_0 \cup V_S\}$ is the set of first-echelon arcs, and $A_2 = \{(i, j) | i, j \in V_S \cup V_C, i \neq j\} \backslash \{(i, j) | i, j \in V_S, i \neq j\}$ is the set of second-echelon arcs. There is a nonnegative transportation cost $c_{ij}$ associated with each arc $(i, j)$.

Each customer $i \in V_C$ has a delivery demand $d_i$ and a pickup demand $p_i$. As we have already mentioned in Sect. 1, we assume that the pickup demand of a customer is less than its delivery demand. Each satellite $s \in V_S$ has a delivery demand and a pickup demand, which are not known at the beginning, but they can be calculated once the assignment of customers to the corresponding satellites is determined. Specifically, the total delivery demand of a satellite $s$ can be calculated as $\sum_{i \in s} d_i$, where $S$ is the set of customers assigned to satellite $s$. Similarly, the total pickup demand is calculated as $\sum_{i \in s} p_i$.

The demand of each customer $i \in V_C$ must be satisfied within a time window $[e_i, l_i]$. The time window means that the service is not allowed to start either before or after a section. Waiting is permitted at all locations at no cost. Fur-
thermore, no time window is considered for either a satellite or the depot. Upon the arrival to customer $i$, delivery freight requires a service time $s_i$. Furthermore, the service time of a satellite is proportional to the quantity of cargoes unloaded from the first-echelon vehicles with a parameter $\tau$. Once a first-echelon vehicle has arrived at a satellite, its cargoes are loaded onto the second-echelon vehicles as soon as possible. The capacities of vehicles are denoted as $Q_1$ and $Q_2$ in the first echelon and second echelon, respectively. Also, let $K_1$ and $K_2$ be the sets of vehicles in the first echelon and second echelon, respectively.

2E-VRPTWSPD tries to find an assignment of customers to satellites in the second-echelon stage and determine the vehicle routes with the minimum total cost in both echelons. It is worth noting that no direct shipments from the depot to customers are allowed. Detailed mathematical formulation of 2E-VRPTWSPD is provided in Appendix. In actual, 2E-VRPTWSPD involves three stages of routing. As illustrated in Fig. 2, firstly, the first-echelon vehicles (i.e., $FV^d_i$) start from the depot to deliver cargoes to satellites. Secondly, the second-echelon vehicles (i.e., $SV_1$, $SV_2$, $SV_3$) start from satellites to serve customers with the simultaneously pickup and delivery manner, and finally return to their satellites with pickup cargoes. Thirdly, the first-echelon vehicles (i.e., $FV^{p}_i$) start from the depot to collect cargoes on satellites.

Different from most 2E-VRP where each first-echelon vehicle can only visit a satellite at most once, our problem relaxes such requirement. Since the service time of a satellite depends on the quantity of cargoes shifted from the first-echelon vehicle to the second-echelon vehicle(s), it is possible that a first-echelon vehicle visits a satellite more than once to potentially lower the total cost.

Figures 2 and 3 illustrate two 2E-VRPTWSPD examples. The numbers next to the line segments are the distance of arcs. Information about the distribution process is shown in the table in the right part, where $[e_i, l_i]$ is the time window, $a_i$ is the arrival time at each customer or satellite, $s_i$ is the service time, and $d_i$ is the demand on customers or the total demand for customers assigned to the satellites. In these two examples, we set $\tau = 0.1$.

Detailed explanations of the figures are as follows. In Fig. 2, a first-echelon vehicle $FV^d_1$ starts from $CD$, firstly arrives at $S_1$ at time $a_{S_1} = 12$. Then, in the second-echelon at $S_1$, $FV^{p}_1$ costs $(d_{C_1} + d_{C_5}) \cdot \tau = 5$ units of time to unload its cargoes for $C_1$ and $C_5$. The second-echelon vehicle, denoted as $SV_1$, starts from $S_1$ at 17, and arrives at $C_1$ at time 22. Back to $FV^{d}_1$, it immediately loads cargoes for $C_1$ and $C_4$, which cost 10 units of time. Hence, $SV_2$ can only start its delivery at time 27 and reach $C_3$ at time 32. For $FV^d_1$, after finishing its assignment of cargoes with the total service time $5 + 10 = 15$, it starts from $S_1$ at time 27 and reaches $S_2$ at time 47. Further information is shown in the table. In this routing plan, the time window of $C_2$ is violated.

In Fig. 3, $FV^d_1$ starts from $CD$ to visit $S_1$. Only the cargoes for $SV_1$ are transferred. Then, $FV^d_1$ serves $S_2$ and returns to $S_1$ to transfer cargoes for $SV_2$. Because of time-saving of the service time at $SV_2$, the arrival time of $C_2$ is advanced. This routing plan is feasible, and its cost is less than that using two first-echelon vehicles. This example shows that for 2E-VRPTWSPD with un-fixed service time, which is more general in real-world, it is reasonable to visit a satellite more than once. Note that the route of $FV^d_1$ is not influenced, because there is no time constraint for the pickup stage of first-echelon vehicles.

### 4 Solution approach

To solve medium-to-large sized 2E-VRPTWSPD instances, a variable neighborhood tabu search algorithm is implemented. Its main techniques are summarized as follows.

In order to improve the solution representation of satellites, we propose a new concept called dummy satellites. This representation can greatly simplify the design of operators and other algorithmic components. Besides, we add dummy time windows to these dummy satellites to accelerate the search. Two operators in each of three vehicle routing stages (i.e., a total of six neighborhood operators) are used in our tabu search to explore the search space. A greedy algorithm is provided to construct an initial solution and estimate whether a feasible solution exists or not. Another important feature of our approach is the possibility of exploring infeasible solutions during the search. We penalize the violation of time windows and vehicle capacities. The penalty strategy can facilitate the exploration of the search space and is particularly useful for those tightly constrained instances.

#### 4.1 Solution representation

Solution representation is an important factor that affects the performance of a heuristic algorithm. We propose dummy satellites by splitting each satellite into several dummy satellites. Each dummy satellite only connects with one second-echelon tour and can only be served once.

Figure 5 shows an example of a solution with dummy satellites corresponding to the original solution in Fig. 4. As depicted in Fig. 4, there are 3 satellites and 4 second-echelon routes. In particular, satellite $S_1$ has 2 second-echelon routes. Therefore, we divide $S_1$ into 2 dummy satellites, $DS_{S_1-1}$ and $DS_{S_1-2}$. Each of them has only 1 second-echelon route. To unify the expression, $S_2$ and $S_3$ are also represented as $DS_{S_2-1}$ and $DS_{S_3-1}$.

In addition, we impose time windows for these dummy satellites. When the second-echelon route of a dummy satel-
Fig. 2 An example of visiting a satellite once in one route

Fig. 3 An example of visiting a satellite twice in one route

Inspired by the method used in Li et al. (2020), we derive the computational formula to simplify the disposal of the first-echelon network as follows. For the $i$th customer delivered from a dummy satellite $j$ (or route $j$), we introduce a variable $TS_i^j$ to represent the maximum duration for the first-echelon vehicle that can postpone to arrive at satellite $j$ related to customer $i$’s time window:

$$TS_i^j = \sum_{k=1}^{i-1} \max\{e_{k}^{j} - a_{k}^{j}, 0\} + l_{i}^{j} - a_{i}^{j},$$

where $e_{i}^{j}, l_{i}^{j}$ and $a_{i}^{j}$ is the left end, right end time window and the arriving time for the $i$th customer in route $j$. 
We also use \( T S^j \) to represent the minimum value for all customers in the second-echelon route starting from satellite \( j \) (denoted as \( \text{route}(j) \)):

\[
TS^j = \min(TS_i^j \mid i \in \text{route}(j)).
\]

Then, we can get the latest arrival time at satellite \( j \) for first-echelon vehicles:

\[
l_j = a_i^j + TS^j - c_{jj^*} - s_j^j.
\]

where \( j^* \) is the first customer in \( j \)'s route, \( s_j^j \) is the service time of dummy satellite \( j \) (note that we use the superscript \( j \) to distinguish with the subscript \( i \) used in service time \( s_i \)). This equation connects the deadline time of customers at satellites by considering the time cost from dummy satellites to its first customer.

Figures 6 and 7 show the examples of how to construct dummy satellite time windows. Each example includes some second-echelon route \( j \) (denoted “DS customer 1 customer 2...”). Suppose that we have obtained a routing plan. The time line of each route is illustrated above the route. As shown in Fig. 6, \( TS_1^j = t_1, TS_2^j = t_2 + t_3, TS_3^j = t_2 + t_4, TS^j = \min(TS_1^j, TS_2^j, TS_3^j), l_j = a_1 + TS^j - c_{j1} - s_j^j \). In Fig. 7, no matter what time a first-echelon vehicle arrives at \( j \), customer 2’s time window constraints cannot be satisfied. For this situation, we set \( l_j = -1 \). The application of dummy satellite time windows will be further explained in Sect. 4.4.3.

### 4.2 Search space

In 2E-VRPTWSPD, if a route violates the capacity constraint or time window constraint, the corresponding solution is infeasible. Similar to Cordeau et al. (2001), we use a weighted penalty function to take these violations into account.

Consider the fitness function \( f(s) = c(s) + \alpha t(s) + \beta d(s) \) of a solution \( s \), where \( c(s) \) is the objective value of 2E-VRPTWSPD, \( t(s) \) and \( d(s) \), respectively, represent the violations of the time window and vehicle capacity, calculated as follows:

\[
t(s) = \sum_{i \in V_C} \sum_{v \in K_2} \max\{a_i^v - l_i, 0\},
\]

\[
d(s) = \sum_{k \in K_1} \max\{w_k - Q_k, 0\} + \sum_{k \in K_2} \max\{w_k - Q_k, 0\},
\]

where \( K_1, K_2 \) is the set of the first and second-echelon vehicles, \( a_i^v \) is the arrival time for the \( i \)th customer in route \( v \), \( w_k \) is the delivery or pickup demand for vehicle \( k \). \( \alpha \) and \( \beta \) are the corresponding weights. The weights are dynamically adjusted within an interval \([LB, UB]\), where \( LB \) and \( UB \) are predetermined by preliminary experiments.

Initially, \( \alpha \) and \( \beta \) are randomly chosen within the interval. Whenever the vehicle capacity constraint or the time window constraint is violated, the respective weight (\( \alpha \) or \( \beta \)) is multiplied by a parameter \( \delta > 1 \); when the solution is feasible, the respective weight is divided by \( \delta \). Note that the updated weight must also lie in the interval \([LB, UB]\).
Two-echelon vehicle routing problem with time windows and simultaneous pickup and delivery

4.3 Initial solution

We adopt a greedy algorithm to construct an initial solution and estimate whether a feasible solution exists. The greedy algorithm is shown in Algorithm 1, where SList, CList, and DSList represent the lists of available satellites, customers, and dummy satellites, respectively. feasibility is a binary indicator for the problem, and solution is the detailed routing plan.

Algorithm 1 Greedy algorithm to create initial solution

Require: SList, CList
Ensure: feasibility, solution
classify customers to satellites (SList, CList);
RS1, feasibility = construct RS1 (SList, CList);
DSList = generate dummy satellites(RS1);
RS2 = construct RS2(DSList);
RS3 = construct RS3(DSList);
solution = create solution by RS1, RS2, RS3;
return feasibility and solution;

4.3.1 Classification of customers

In order to serve a customer as soon as possible, we first classify the customers according to the distances to their closest satellite. Specifically, for each customer \( i \in V_C \), choose the satellite \( s \in V_S \) which minimizes the sum of the distance between \( C \) to \( s \) and \( s \) to \( i \).

4.3.2 Construction of routes

The problem for constructing RS1 is essentially a VRPTW. To construct a new route, the customer with the minimum latest-service-starting time is first inserted into the route. Then, the customer with the maximum saving is inserted stepwise. Before inserting customers into a route, a validity check is applied to guarantee the time window of each customer in the route and the load of the vehicle, since the insertion of customers will change the transfer time in dummy satellites. After the second-echelon route is determined, dummy satellites are generated.

Similarly, the problem for RS2 is a VRP with deadline constraints (no left end time window), and for RS3 is a classic CVRP. The corresponding routes are constructed in the same way.

4.4 Variable neighborhood tabu search algorithm

Our variable neighborhood tabu search algorithm is a mixture of both TS and VNS. Tabu Search (TS) is a memory-based search strategy to guide the local search to continue its search beyond local optimality (Glover 1990; Belhaiza et al. 2014). When a local optimum is encountered, a move to the best neighbor is made to explore the solution space, even though it may cause a deterioration in the objective function value.
TS seeks the best admissible move that can be determined in a reasonable amount of time.

Variable neighborhood search (VNS) is a generic local search methodology introduced by Mladenović and Hansen (1997). It has been successfully applied to a variety of contexts, including graph theory, packing problems, and location routing. The main idea of VNS is to define multiple neighborhoods to enlarge the search space.

4.4.1 Neighborhood structures

In the literature, $\lambda$-interchange is one of the best neighborhood structures for optimizing VRPTW. We consider two types of $\lambda$-interchange, which are denoted as $(l_1, l_2)$-interchange. $l_1$ and $l_2$ represent the number of continuous segments of two different routes.

We choose to adopt $(1, 0)$-interchange and $(1, 1)$-interchange. Figure 8a-d shows several examples of $\lambda$-interchange operators for the first- and second-echelon routes, where rectangles represent the depot and dummy satellites, and circles represent customers.

Figure 8a gives an example of $(1, 0)$-interchange operator in the second echelon. Two routes (i.e., “$DS_1-C_1-C_2-C_3-C_4-DS_1$” and “$DS_2-C_5-C_6-C_7-C_8-DS_2$”) are selected. A customer of the former route ($C_2$) is removed and then inserted into the latter route. Figure 8b shows an example of...
(1, 1)-interchange operator. A customer of the former route (C2) and a customer of the latter route (C0) are exchanged. For the first-echelon operators, when the dummy satellites are moved, its associated customers are moved as well. Take Fig. 8c as an example, C3 is inserted into the latter route with dummy satellite D52.

In sum, both (1, 0)-interchange and (1, 1)-interchange are applied in all the three route planning stages to obtain a total of six different neighborhood operators, named \(O_{1-0}, O_{1-1}, O_{2-0}, O_{2-1}, O_{p_{1-0}}, \) and \(O_{p_{1-1}}\).

### 4.4.2 Tabu search

The main process of the proposed method is tabu search, which is shown in Algorithm 2. A tabu list is defined as a two-dimensional array for each pair of nodes (a node can be either the depot, dummy satellite or customer). The element in the array indicates the number of iterations that the corresponding node-pair cannot be altered.

In each iteration, the algorithm randomly selects two neighborhood operators N1 and N2 and generates two neighboring solutions \(s_{N1}\) and \(s_{N2}\). The better one is recorded by \(s^*\), and the global best solution \(s_{best}\) is updated accordingly.

The associated operator is then used to update the tabu list by forbidding any operation on the related node-pairs for a few iterations. The red lines of each operator depicted in Fig. 8 indicate the node-pairs to be added into the tabu list. For example, if (1, 0)-interchange for the second-echelon route takes effect, the related node-pairs (C1, C2), (C2, C3) and (C1, C3) are taboo.

### 4.4.3 Solution evaluation strategy

In tabu search, the objective function should be evaluated when a new solution is created by neighborhood operators. In 2E-VRPTWSPD, the operators of different stages always have interaction effects. For example, when a second-echelon operator is applied, the transfer time of first-echelon vehicles at satellites will be changed, and the corresponding first-echelon route will be affected. In this case, it may take a lot of time to evaluate those newly generated solutions.

We hereby use the dummy time windows mentioned above to improve the solution evaluation process. For each dummy satellite, if its corresponding second-echelon route is determined, the dummy time window is kept unchanged. According to the definition of dummy satellite time windows, if a first-echelon vehicle arrives at the dummy satellite before its deadline, no time window constraints of customers will be violated, and thus, the fitness function will not change.

To be more specific, when an operator is applied, the arrival time of dummy satellites in the first-echelon routes may be affected. For a dummy satellite whose second-echelon route has not changed, we can check whether its new arrival time is earlier than the right end of its time window. If so, there is no time window violation in this dummy satellite, then we can omit the calculation of its associated customers.

### 5 Computational results

In this section, we present the computational study of our mathematical model and heuristic algorithm to examine their performances.

#### 5.1 Problem settings

To our best knowledge, there is no previous approach studied 2E-VRPTWSPD. We hereby consider a small and a large instance set modified from the literature.

The small-scale instances are originated from Dellaert et al. (2019) and used to compare the exact solution with the heuristic solution. The original instances are proposed to test multidepot 2E-VRPTW with up to 15 customers and 3 satellites. To ensure that the mathematical model can get the optimal solutions, we remove some customers, satellites, and depots to obtain 2E-VRPTWSPD instances. The pickup demand of customers is set to half of their delivery demand.

The scales of instances are 7, 10, 12 customers with 2 satellites; each has 6 instances. Each instance is named by a notation that consists of the number of satellites, number of customers, and instance id. For example, “7-2-1” denotes the first instance with 2 satellites and 7 customers.

For large-scale instances, we make use of the instances proposed by Hemmelmayr et al. (2012) for 2E-VRP. The scales of instances are 100 customers with 5 satellites, 100 customers with 10 satellites, and 200 customers with 10 satellites; each has 6 instances. To adapt to 2E-VRPTWSPD, we generate time windows for these instances using the method proposed by Solomon (1987) and randomly generate pickup
demands. Notice that we keep the original notations, like “100-10-1” or “100-10-1b.” There is no obvious connection between these two instances in Hemmelmayr et al. (2012).

We used the commercial solver CPLEX 12.6.3 to solve the mathematical formulation directly. The heuristic was coded in Java SE 1.8.0. All the experiments were conducted on an ASUS personal computer with an AMD Ryzen™ 7 4800H 2.90GHz CPU, 8G RAM, and Windows 10 operating system. For each small-scale instance, CPLEX ran with default settings until finding an exact solution or stopping due to the exhaustion of the predetermined maximum computation time, which was set to 1 h. For the heuristic algorithm, we set the tabu tenure as 30. The termination principle is reaching either the maximum number of iterations (I1) or the maximum number of iterations without improving the best solution (I2). For small-scale instances, we set I1 = 1000 and I2 = 200. For large-scale instances, we set I1 = 25000 and I2 = 5000.

### 5.2 Results on small-scale instances

A direct solution of the mathematical formulation can be obtained by the exact method of CPLEX 12.6.3 on small-scale instances. We then use these results to examine the performance of our heuristic algorithm. In Table 1, we list the exact and heuristic results on the small-scale instances. Column 1 indicates the instance name. Columns 2 and 3 show the objective value (ObjE) and computation time (TE) of the exact solution obtained by CPLEX. Columns 4 and 5 show the minimum result of objective value executed 10 times (ObjMinH) and the percentage gap between columns 2 and 4 (GAP1). Columns 6, 7, and 8 show the maximum objective value (ObjMaxH), percentage gap between columns 2 and 6 (GAP2), average objective value (ObjAH), and the percentage gap between columns 2 and 7 (GAP3). Column 10 shows the computation time (TH) of the heuristic solution.

The performance measurements considered in the comparison are: (1) percentage gap (Gap): calculated as 100% * (ObjH - ObjE)/ObjE, where ObjH is the objective value obtained by the heuristic; (2) CPU time of the mathematical model or the heuristic.

From Table 1, it is observed that the mathematical model can reach optimum on all 6 instances with 7 customers and 2 satellites, and on 8 out of all 18 instances. Our heuristic algorithm can also find all these 8 optimal results. For 17 out of 18 instances, it can find better or at least the same results compared with the mathematical model. On all the instances with 7, 10 or 12 customers and 2 satellites, our heuristic algorithm costs very tiny computing power, i.e., the maximum CPU time is smaller than 0.05 s.
Table 2  Heuristic results of large-scale instances

| Instance  | $Obj$ | $n_1$ | $n_2$ | $n_3$ | $Obj_1$ | $Obj_2$ | $Obj_3$ | $T$ |
|-----------|-------|-------|-------|-------|---------|---------|---------|-----|
| 100-5-1   | 2293.21 | 4     | 27    | 2     | 551.50  | 1445.38 | 296.34  | 44.37 |
| 100-5-1b  | 2060.28 | 5     | 17    | 2     | 692.86  | 1071.08 | 296.34  | 56.88 |
| 100-5-2   | 1952.62 | 5     | 28    | 2     | 539.62  | 1149.64 | 263.36  | 27.97 |
| 100-5-2b  | 1473.74 | 4     | 17    | 2     | 431.85  | 778.53  | 263.36  | 12.39 |
| 100-5-3   | 1781.76 | 5     | 27    | 2     | 604.11  | 1256.86 | 279.96  | 25.60 |
| 100-5-3b  | 1552.63 | 4     | 18    | 2     | 493.92  | 790.43  | 268.28  | 44.34 |
| 100-10-1  | 2135.61 | 6     | 27    | 2     | 598.79  | 1256.86 | 279.96  | 53.11 |
| 100-10-1b | 1845.14 | 6     | 19    | 2     | 620.35  | 968.58  | 256.20  | 12.39 |
| 100-10-2  | 2051.30 | 5     | 27    | 2     | 377.86  | 1494.38 | 179.06  | 24.87 |
| 100-10-2b | 1906.62 | 7     | 18    | 2     | 514.48  | 1212.47 | 179.67  | 63.46 |
| 100-10-3  | 1805.02 | 5     | 21    | 2     | 596.01  | 933.38  | 275.63  | 47.23 |
| 200-10-1  | 3376.97 | 9     | 48    | 2     | 978.80  | 2119.76 | 278.41  | 282.18 |
| 200-10-1b | 2877.07 | 9     | 36    | 2     | 937.91  | 1647.47 | 291.70  | 226.62 |
| 200-10-2  | 2947.96 | 8     | 51    | 2     | 784.07  | 1910.94 | 252.95  | 94.05 |
| 200-10-2b | 2747.89 | 10    | 33    | 2     | 992.45  | 1442.47 | 312.98  | 215.56 |
| 200-10-3  | 3444.24 | 8     | 48    | 2     | 752.27  | 2432.41 | 259.56  | 175.04 |
| 200-10-3b | 2814.80 | 8     | 33    | 2     | 768.21  | 1816.63 | 229.97  | 235.26 |

5.3 Results on large-scale instances

The comparison of heuristic solutions with exact solutions on small-scale instances has verified the effectiveness of our method. In general, the heuristic approach is applicable in tackling large-scale instances. Table 2 shows the results of our heuristic algorithm on large-scale instances with 100/200 customers and 10/15 satellites. The first column of Table 2 displays the instance name. The second to eighth columns report the objective value, the number of the first-echelon routes for delivery, the number of second-echelon routes, the objective value for first-echelon distribution phase, the objective value for second-echelon and the objective value for first-echelon collection phase, respectively.

The experimental results show that our algorithm has a good convergence performance, as we find that most instances are finished due to reaching the maximum number of iterations without improving the best solution. For the instances with 100 customers, the algorithm is converged within about 1 min; for 200 customers, the solutions are astringed within about 5 min. We further provide the convergence plot for two selected instances “100-10-3b” and “200-10-3” in Fig. 9. The plot indicates the fast convergence of the algorithm in the first few iterations.

Finally, we find that the running time of the algorithm depends more on the number of customers than the number of satellites, because the scale of customers is much larger than that of satellites. For large-scale instances, the distance traveled by the second-echelon vehicle is much larger than that by the first-echelon vehicle. Compared with the objective value in the third stage, the first stage delivery requires more vehicle routes due to the deadline constraints and larger delivery demand; thus, its objective function $Obj_1$ is generally larger than $Obj_3$. 

5.4 Effects of different operators

Our tabu search procedure depends on six operators to exploit the search space, as mentioned in Sect. 4. To verify the importance of each operator, we conducted ablation study. We, respectively, removed one of the operators in the set \{O_1d_0, O_1d_1, O_2l_0, O_2l_1, O_1p_0, O_1p_1\} from our algorithm to generate six variants and then compared the result of these variants with the original algorithm.

To carry out comparative experiments, we used the set of large instances as the benchmark. Table 3 reports the gap between our algorithm and its six variants for each tested instance. Column 1 indicates the instance name. Columns 2 to 7 indicate the percentage gap between our algorithm and its six variants, denoted as $Gap_1$ to $Gap_6$. A negative value of $Gap_n$ indicates that the result is better.

As shown in Table 3, our algorithm performs better than 6 variants in most of the test instances. Specifically, the result of our algorithm is the best one in 14 instances, except 100-5-3, 100-10-1b, 100-10-3b and 200-10-2b. The biggest percentage gap between the best solution with our heuristic is 1.21. Considering the random factor of the heuristic, this analysis
5.5 Impact of the satellite time windows

As described in Sect. 4, our heuristic algorithm uses dummy satellite time windows to speed up the search. To evaluate its merit, we removed this technique from our algorithm and tested the reduced execution time. We conducted experiments on 18 large-scale instances. Notice that removing the satellite time windows will not change the final solution, so we omit the presentation of objective values.

Table 4 lists the experimental results, which include the running time with/without satellite time windows ($T_0/T_1$), and the percentage gap ($GAP$) calculated by $100\% \times (T_1 - T_0)/T_1$.

![Convergence plot for two instances](image)

Fig. 9  The convergence plot for two instances

implies that excluding the use of any operator will impair the solution quality.
$T_0/T_0$. The last row summarizes the average result over 18 instances.

From this table, we find that the algorithm with satellite time windows executes faster on all 18 instances. To be specific, the satellite time windows are able to speed up the search procedure with 19.00% on average. For the best one (instance 200-10-3), the percentage gap can reach 45.72%. The experiments demonstrate the importance of introducing the satellite time windows in accelerating the algorithm.

6 Conclusions

In this paper, we introduce a two-echelon vehicle routing problem with time windows and simultaneous pickup and delivery (2E-VRPTWSPD). It extends the classic two-echelon vehicle routing problem (2E-VRP) and has several applications in practice. A mathematical model is proposed to describe the problem. We then present a variable neighborhood tabu search heuristic algorithm to solve the problem. To test our algorithm, we generate two instance sets with small and large scales based on the existing benchmark sets. The results show that our heuristic algorithm is effective and efficient to find good solutions for 2E-VRPTWSPD. Furthermore, we show by statistical analysis that the strategies of combing multiple neighborhood operators and introducing the dummy satellite time windows can significantly improve the performance and speed of the heuristic algorithm.

Our future research on 2E-VRPTWSPD will focus on the design of more powerful valid inequalities in branch-and-cut algorithms. In addition, branch-and-price and column generation algorithms are a class of the most successful exact algorithms to solve many routing problems. We believe that they can be applied to solve 2E-VRPTWSPD instances to optimality.

Author Contributions All the authors contributed to the study conception and design. The study direction and specific problem definition were proposed by H Qin and J Li. The algorithm and mathematical model were proposed by H Zhou and J Li and programmed by H Zhou. The first draft of the manuscript was written by Z Zhang and H Zhou, and all the authors commented on previous versions of the manuscript. All the authors approved the final manuscript.

Funding This research was supported by National Key R&D Program of China (No. 2018YFB1700600), National Natural Science Foundation of China (No. 71971090), and Guangdong Natural Science Funds (Nos. 2019A1515011169, 2021A1515011301).

Data availability The authors can provide the instance sets and detailed computational results upon requests. The datasets generated during and/or analyzed during the current study are available from the corresponding author on reasonable request.

Code Availability The experiments can be reproduced by using code_ aster.

Declarations

Conflict of interest The authors have no conflicts of interest to declare that are relevant to the content of this article.

Informed consent Informed consent was obtained from all individual participants included in the study.

A Arc-based formulation for 2E-VRPTWSPD

To model the 2E-VRPTWSPD, we introduce dummy node sets $V^s_{DS}$ for each satellite. $V^s_{DS}$ represents node set of the first echelon. Let $A'_1 = \{(i, j) | i, j \in V_0 \cup V'_s, i \neq j\}$ and $A'_2 = \{(i, j) | i, j \in V'_s \cup V_C, i \neq j\}\\{(i, j) | i, j \in V'_s, i \neq j\}$.

Inspired by the ideas from Liu et al. (2018) and Li et al. (2019), we formulate 2E-VRPTWSPD as a mixed-integer programming formulation based on the used vehicles in the first and second echelon. The variables in this model are defined as follows.

- $x_{ij}^{1k}$: a binary decision variable and relative to the first-echelon vehicles, which is equal to 1 if arc $(i, j) \in A'_1$ is traveled by vehicle $k \in K_1$ for distribution, and 0 otherwise;
- $x_{ij}^{2k}$: a binary decision variable and relative to the first-echelon vehicles, which is equal to 1 if arc $(i, j) \in A'_1$ is traveled by vehicle $k \in K_1$ for collection, and 0 otherwise;
- $w_{ij}^{1k}$: a decision variable representing the quantity delivered to satellite $s \in V'_s$ by vehicle $k \in K_1$;
- $w_{ij}^{2k}$: a decision variable representing the quantity collected to satellite $s \in V'_s$ by vehicle $k \in K_1$;
- $u_{ij}^{1k}$: a decision variable representing the position of satellite $s \in V'_s$ in the route of vehicle $k \in K_1$ for distribution;
- $u_{ij}^{2k}$: a decision variable representing the position of satellite $s \in V'_s$ in the route of vehicle $k \in K_1$ for collection;
- $f_{ij}$: a decision variable representing the amounts of the delivery commodities travel through arc $(i, j) \in A'_2$;
- $g_{ij}$: a decision variable representing the amounts of the pickup commodities travel through arc $(i, j) \in A'_2$;
- $y_{ij}$: a binary decision variable and relative to the second-echelon vehicles, which is equal to 1 if vehicle $v \in K_2$ travels through arc $(i, j) \in A'_2$, and 0 otherwise;
- $a_{ik}$: a decision variable representing the arrival time of the first-echelon vehicle to satellite $s \in V'_s$ in the route of vehicle $k \in K_1$.
- $a_{ij}^v$: a decision variable representing the arrival time of the second-echelon vehicle to customer $i \in V_C$ in the route of vehicle $v \in K_2$;
- $s_s^k$: a decision variable representing the service time of satellite $s \in V_S'$ in the route of vehicle $k \in K_1$.

With these variables and parameters, we can formulate the following mixed integer program:

$$\min \sum_{k \in K_1} \sum_{(i,j) \in A'_1} c_{ij}(x^1_{ij} + x^2_{ij}) + \sum_{v \in K_2} \sum_{(i,j) \in A'_2} c_{ij} y^v_{ij}$$  \tag{1}

subject to

$$\sum_{(i,j) \in A'_1} x^1_{ij} = \sum_{(i,j) \in A'_1} x^k_{ij}, \quad \forall i \in V_0 \cup V_S', k \in K_1$$  \tag{2}

$$\sum_{(i,j) \in A'_1} x^2_{ij} = \sum_{(i,j) \in A'_1} x^k_{ij}, \quad \forall i \in V_0 \cup V_S', k \in K_1$$  \tag{3}

$$\sum_{(i,j) \in A'_1} x^1_{ij} \leq 1, \quad \forall k \in K_1$$  \tag{4}

$$\sum_{(i,j) \in A'_1} x^2_{ij} \leq 1, \quad \forall k \in K_1$$  \tag{5}

$$\sum_{k \in K_1 (i,j) \in A'_1} x^1_{ij} \leq 1, \quad \forall i \in V_S'$$  \tag{6}

$$\sum_{k \in K_1 (i,j) \in A'_1} x^2_{ij} \leq 1, \quad \forall i \in V_S'$$  \tag{7}

$$u^1_{ij} + 1 \leq u^1_{ij} + M (1 - x^1_{ij}), \quad \forall i \in V_S', j \in V_S',$$  \tag{8}

$$1 \leq u^1_{ij} + M (1 - x^1_{ij}), \quad \forall j \in V_S', k \in K_1$$  \tag{9}

$$u^2_{ij} + 1 \leq u^2_{ij} + M (1 - x^2_{ij}), \quad \forall i \in V_S', j \in V_S'$$  \tag{10}

$$1 \leq u^2_{ij} + M (1 - x^2_{ij}), \quad \forall j \in V_S', k \in K_1$$  \tag{11}

$$u^1_{ij} \leq M \sum_{(i,j) \in A'_1} x^k_{ij}, \quad \forall s \in V_S', k \in K_1$$  \tag{12}

$$u^2_{ij} \leq M \sum_{(i,j) \in A'_1} x^k_{ij}, \quad \forall s \in V_S', k \in K_1$$  \tag{13}

$$\sum_{i \in V_S'} y^v_{ij} \leq Q_1, \quad \forall k \in K_1$$  \tag{14}

$$\sum_{i \in V_S'} y^v_{ij} \leq Q_1, \quad \forall k \in K_1$$  \tag{15}

$$\sum_{(i,j) \in A'_2} y^v_{ij} = \sum_{(i,j) \in A'_2} y^v_{ij}, \quad \forall i \in V_C \cup V_S', v \in K_2$$  \tag{16}

$$\sum_{i \in K_2 (i,j) \in A'_2} y^v_{ij} = 1, \quad \forall i \in V_C$$  \tag{17}

$$\sum_{i \in K_2 (i,j) \in A'_2} y^v_{ij} = \sum_{k \in K_1 (i,j) \in A'_1} x^k_{ij}, \quad \forall i \in V_S'$$  \tag{18}

$$\sum_{i \in V_S'} y^v_{ij} \leq 1, \quad \forall v \in K_2$$  \tag{19}

$$\sum_{(i,j) \in A'_2} f_{ij} = \sum_{(i,j) \in A'_2} f_{ij} + d_i, \quad \forall i \in V_C$$  \tag{20}

$$\sum_{(i,j) \in A'_2} g_{ij} = \sum_{(i,j) \in A'_2} g_{ij} - p_i, \quad \forall i \in V_C$$  \tag{21}

$$\sum_{k \in K_1 (i,s) \in A'_2} u^1_{is} = \sum_{(i,s) \in A'_2} f_{is}, \quad \forall s \in V_S'$$  \tag{22}

$$\sum_{k \in K_1 (i,s) \in A'_2} u^2_{is} = \sum_{(i,s) \in A'_2} g_{is}, \quad \forall s \in V_S'$$  \tag{23}

$$s^k_s = \tau \ast w^1_{is}, \quad \forall s \in V_S', k \in K_1$$  \tag{24}

$$(d_j - p_j) \ast \sum_{v \in K_2} y^v_{ij} \leq f_{ij} + g_{ij}$$  \tag{25}

$$a^v_j \geq a^v_i + c_{ij} + s^k_i - M \left(1 - x^1_{ij}\right),$$  \tag{26}

$$\forall i \in V_S', j \in V_S', (i,j) \in A'_1, k \in K_1$$

$$a^v_j \geq c_{ij} - M \left(1 - x^1_{ij}\right),$$  \tag{27}

$$\forall j \in V_S', k \in K_1$$

$$a^v_j \geq a^v_i + c_{ij} + s^k_i - M \left(2 - y^v_{ij} - \sum_{(h,i) \in A'_1} x^k_{hi}\right),$$  \tag{28}

$$\forall i \in V_S', j \in V_C, k \in K_1, v \in K_2$$

$$a^v_j \geq a^v_i + c_{ij} + s^k_i - M \left(1 - y^v_{ij}\right),$$  \tag{29}

$$\forall i \in V_C, j \in V_C, (i,j) \in A'_2, v \in K_2$$

$$a^v_i \geq e_i, \quad \forall i \in V_C, v \in K_2$$  \tag{30}

$$a^v_i \leq l_i, \quad \forall i \in V_C, v \in K_2$$  \tag{31}

$$\sum_{i \in V_S'} x^1_{ij}, x^2_{ij} \in \{0, 1\}, \quad \forall (i,j) \in A'_1, k \in K_1$$  \tag{32}

$$w^1_{is}, w^2_{is}, u^1_{is}, u^2_{is} \geq 0, \quad \forall s \in V_S', k \in K_1$$  \tag{33}

$$f_{ij}, g_{ij} \geq 0, \quad \forall (i,j) \in A'_2$$  \tag{34}

$$y^v_{ij} \in \{0, 1\}, \quad \forall (i,j) \in A'_2, v \in K_2$$  \tag{35}

$$s^k_i, a^v_i \geq 0, \quad \forall s \in V_S', k \in K_1$$  \tag{36}

$$a^v_i \geq 0, \quad \forall i \in V_C, v \in K_2$$  \tag{37}

The objective function (1) minimizes the sum of the first-echelon and second-echelon traveling cost. Constraints
Constraints (2)–(3) are the flow conservation constraints for each satellite. Constraints (4)–(7) ensure that a dummy satellite can be visited at most once. Constraints (8)–(11) avoid the presence of sub-tours in the first echelon. Constraints (12)–(13) guarantee that a first-echelon vehicle can conduct distribution or collection at a satellite, only if the vehicle visits that satellite. Constraints (14)–(15) are the capacity constraints of each first-echelon vehicle. Constraints (16) are the flow conservation constraints in the second echelon. Constraints (17) ensure that each customer is visited only by one vehicle. Constraints (18) ensure that each dummy satellite is served by one second-echelon vehicle. Constraints (19) ensure that each second-echelon vehicle is used at most once. Constraints (20)–(21) are the flow conservation constraints for distribution and collection, respectively. Constraints (25) bound the flow of goods traveling on each arc not exceeded the capacity of the second-echelon vehicle. Constraints (22) ensure that the amount of distribution to the customers from a satellite is equal to that of delivery to this satellite from the depot. Constraint (23) guarantee that the amount of collections from the customers to a satellite is equal to that of pickup from this satellite to the depot. Constraints (24) build the relation between outturn and service time on each satellite. Constraints (26)–(29) calculate the arrival time of vehicles to satellite and customers. Constraints (28) relate the arrival time of a first-echelon vehicle and the departure time of a second-echelon vehicle if they meet at a satellite to carry a demand. It denotes that a second-echelon vehicle can depart from a satellite only after the freight is delivered to the satellite and ready to be delivered. Constraints (30) and (31) are hard time window constraints for the customers. Constraints (32)–(37) are the domain constraints.

References

Belgin O, Karaoglan I, Altiparmak F (2018) Two-echelon vehicle routing problem with time windows and simultaneous pickup and delivery: mathematical model and heuristic approach. Comput Ind Eng 115:1–16

Belhaiza S, Hansen P, Laporte G (2014) A hybrid variable neighborhood tabu search heuristic for the vehicle routing problem with multiple time windows. Comput Oper Res 52:269–281

Breunig U, Baldacci R, Hartl RF, Vidal T (2019) The electric two-echelon vehicle routing problem. Comput Oper Res 103:198–210

Cordeau JF, Laporte G, Mercier A (2001) A unified tabu search heuristic for vehicle routing problems with time windows. J Oper Res Soc 52(8):928–936

Crainic TG, Ricciardi N, Storchi G (2009) Models for evaluating and planning city logistics systems. Transp Sci 43(4):432–454

Cuda R, Guataroba G, Speranza MG (2015) A survey on two-echelon routing problems. Comput Oper Res 55:185–199

Dellaert N, Dashy Saridarq F, Van Woensel T, Crainic TG (2019) Branch-and-price-based algorithms for the two-echelon vehicle routing problem with time windows. Transp Sci 53(2):463–479

Dellaert N, Van Woensel T, Crainic TG, Saridarq FD (2021) A multi-commodity two-echelon capacitated vehicle routing problem with time windows: model formulations and solution approach. Comput Oper Res 127:105154

Entchoven DL, Jargalsaikhan B, Roodebergen KJ, uit het Broek MA, Schrotenboer AH (2020) The two-echelon vehicle routing problem with covering options: city logistics with cargo bikes and parcel lockers. Comput Oper Res 118:104919

Glover F (1990) Tabu search-part II. ORSA J Comput 2(1):4–32

Hemmelmayr VC, Cordeau JF, Crainic TG (2012) An adaptive large neighborhood search heuristic for two-echelon vehicle routing problems arising in city logistics. Comput Oper Res 39(12):3215–3228

Kalayci CB, Kaya C (2016) An ant colony system empowered variable neighborhood search algorithm for the vehicle routing problem with simultaneous pickup and delivery. Expert Syst Appl 66:163–175

Kitjacharoenchai P, Min BC, Lee S (2020) Two echelon vehicle routing problem with drones in last mile delivery. Int J Prod Econ 225:107598

Koç Ç, Laporte G, Türkmeniz İ (2020) A review of vehicle routing with simultaneous pickup and delivery. Comput Oper Res 122:104987

Li H, Bai M, Zhao Y, Dai C (2019) Vehicle flow formulation for two-echelon time-constrained vehicle routing problem. J Manag Sci Eng 4(2):75–90

Li H, Wang H, Chen J, Bai M (2020) Two-echelon vehicle routing problem with time windows and mobile satellites. Transp Res Part B: Methodol 138:179–201

Liu R, Xie X, Augusto V, Rodriguez C (2013) Heuristic algorithms for a vehicle routing problem with simultaneous delivery and pickup and time windows in home health care. Eur J Oper Res 230(3):475–486

Liu T, Luo Z, Qin H, Lim A (2018) A branch-and-cut algorithm for the two-echelon capacitated vehicle routing problem with grouping constraints. Eur J Oper Res 266(2):487–497

Liu W, Zhou Y, Liu W, Qin J, Xie N, Chang X, Chen J (2021) A hybrid ACS-VTM algorithm for the vehicle routing problem with simultaneous delivery & pickup and real-time traffic condition. Comput Ind Eng 162:107747

Marques G, Sadykov R, Deschamps JC, Dupas R (2020) An improved branch-cut-and-price algorithm for the two-echelon capacitated vehicle routing problem. Comput Oper Res 114:104833

Min H (1989) The multiple vehicle routing problem with simultaneous delivery and pick-up points. Transp Res Part A: Gen 23(5):377–386

Mladenović N, Hansen P (1997) Variable neighborhood search. Comput Oper Res 24(11):1097–1100

Mu D, Wang C, Zhao F, Sutherland JW (2016) Solving vehicle routing problem with simultaneous pickup and delivery using parallel simulated annealing algorithm. Int J Shipp Transp Logist 8(1):81–106

Park H, Son D, Koo B, Jeong B (2021) Waiting strategy for the vehicle routing problem with simultaneous pickup and delivery using genetic algorithm. Expert Syst Appl 165:113959

Perboli G, Tadei R (2010) New families of valid inequalities for the two-echelon vehicle routing problem. Electron Notes Discrete Math 36:639–646

Perboli G, Tadei R, Vigo D (2011) The two-echelon capacitated vehicle routing problem: models and math-based heuristics. Transp Sci 45(3):364–380

Qin H, Su X, Ren T, Luo Z (2021) A review on the electric vehicle routing problems: variants and algorithms. Front Eng Manag 8(3):370–389

Santos FA, da Cunha AS, Mateus GR (2013) Branch-and-price algorithms for the two-echelon capacitated vehicle routing problem. Optim Lett 7(7):1537–1547

Santos FA, Mateus GR, da Cunha AS (2015) A branch-and-cut-and-price algorithm for the two-echelon capacitated vehicle routing problem. Transp Sci 49(2):355–368
Solomon MM (1987) Algorithms for the vehicle routing and scheduling problems with time window constraints. Oper Res 35(2):254–265
Wang HF, Chen YY (2012) A genetic algorithm for the simultaneous delivery and pickup problems with time window. Comput Ind Eng 62(1):84–95
Wang C, Mu D, Zhao F, Sutherland JW (2015) A parallel simulated annealing method for the vehicle routing problem with simultaneous pickup-delivery and time windows. Comput Ind Eng 83:111–122

Publisher’s Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.