Two-Echelon Vehicle Routing Problem with Time Windows and Simultaneous Pickup and Delivery

Hang Zhou (zh20010728@126.com)
Huazhong University of Science and Technology https://orcid.org/0000-0003-3286-341X
Hu Qin
Huazhong University of Science and Technology - Main Campus: Huazhong University of Science and Technology
Zizhen Zhang
Sun Yat-Sen University https://orcid.org/0000-0003-0320-9355
Jiliu Li
Huazhong University of Science and Technology - Main Campus: Huazhong University of Science and Technology

Research Article

Keywords: Two-echelon vehicle routing problem, time windows, pickup and delivery, tabu search

Posted Date: December 2nd, 2021

DOI: https://doi.org/10.21203/rs.3.rs-604710/v1

Version of Record: A version of this preprint was published at Soft Computing on January 10th, 2022. See the published version at https://doi.org/10.1007/s00500-021-06712-2.

Soft Computing manuscript No.
(will be inserted by the editor)

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

Hang Zhou [1] · **Hu Qin** [1] · **Zizhen Zhang** [✉, 2] · **Jiliu Li** [1]

Received: date / Accepted: date

Abstract In this paper, we propose 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-VRP) by considering the time window constraints and simultaneous pickup and delivery. In 2E-VRPTW SPD, the pickup and delivery activities are performed simultaneously by the same vehicles through the depot to satellites in the first echelon and satellites to customers in the second echelon, where each customer has a specified time window. To solve this problem, firstly, we formulate the problem with a mathematical model. Then, we implement a variable neighborhood tabu search algorithm with the proposed solution representation of dummy satellites to solve large-scale instances. Dummy satellites time windows are used in our algorithm to speed up the algorithm. Finally, we generate two instance sets based on the existing 2E-VRP and 2E-VRPTW benchmark sets and conduct additional experiments to analyze the performance of our algorithm.

Keywords Two-echelon vehicle routing problem · time windows · pickup and delivery · tabu search

1 Introduction

In recent years, the transportation cost in logistics is increasing rapidly. The vehicle routing problem (VRP), which aims to determine the best routing plan for vehicles to serve a set of customers, are widely used to fit 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 final 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 in the second level are usually accomplished by second-echelon vehicles with a small capacity. Freight is first transported from the depot to satellites by first-echelon vehicles. Then, the cargoes on the first-echelon vehicles are loaded into 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.

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

In this study, we consider a two-echelon vehicle routing problem with time windows and simultaneous pickup and delivery problem (2E-VRPTWSPD), which is a new variant of 2E-VRP. This problem can easily be applied in some real-world circumstances. For example, consider the delivery of some medical supplies in a multi-modal 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 house for convenience. Some part of the medical product, such as the wrapper for some liquid or the used syringe needle, need to be called back immedi-

✉ Zizhen Zhang
E-mail: zhangzizhen@gmail.com

1 School of Management, Huazhong University of Science and Technology.
2 School of Computer Science and Engineering, Sun Yat-sen University.

ately, because the abandon wrapper with remnant medicine will pollute the environment, or even be illicitly purchased by drug traffickers and then caused harms. Thus, the reverse logistic becomes an important part in the delivery system. Many reverse logistic examples in multi-modal 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 smaller than the delivery demand. This is a practical assumption, although it is easy to adjust our method to adapt to the general problem, in which the pickup demand may be larger. Second, we find that in most literature on 2E-VRPTW, the service duration in satellites is always fixed. However, in practice, the time used to transfer cargoes is related to the quality of cargoes. In this paper, we assume that the service time in satellites is positively 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, and propose a mixed integer programming mathematical model to formulate the problem. We then provide a heuristic algorithm that includes a greedy algorithm and 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, Section 2 reviews studies on the related work. Section 3 formally defines 2E-VRPTWSPD and introduces a mixed-integer linear programming model. Section 4 presents the 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 briefly review two 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, multi-depot, multiple-tour, heterogeneous VRPTW (2SS-MDMT-VRPTW), and 2E-VRP is the special case of this problem. Formal description and model of 2E-VRP were introduced by Perboli and Tadei (2010) and Perboli et al. (2011). The authors proposed an MIP formulation and derived valid inequalities. Several meta-heuristics were proposed to solve 2E-VRP after that. Crainic et al. (2011) proposed multi-start heuristics by separating the two echelons apart to solve the two routing sub-problems. Hemmelmayr et al. (2012) proposed an Adaptive Large Neighborhood Search (ALNS) combined with a local search. Crainic et al. (2013); Zeng et al. (2014) proposed heuristic algorithms based on Greedy Randomizied Adaptive Search Procedure (GRASP), respectively. Breunig et al. (2016) developed a Large Neighborhood Search (LNS) for 2E-VRP. Cuda et al. (2015) published a survey on two-echelon routing problems, which summarized the development of 2E-VRP. Only a few researchers considered the time window constraints of these problems, which named 2E-VRPTW. Dellaert et al. (2019, 2021) proposed a branch-and-price-based algorithm. Li et al. (2020) introduced a two-echelon vehicle routing problem with time windows and mobile satellites (2E-VRP-TM) and proposed an Adaptive Large Neighborhood Search (ALNS) to solve it.

2E-VRPTWSPD can be treated as VRPSPD when the satellites and time windows are removed. VRPSPD was first introduced by Min (1989). Many heuristic and meta-heuristic approaches for solving VRPSPD have been proposed. For example, Bianchessi and Righini (2007); Chen and Wu (2006); Crispim and Brandão (2005) devised several Tabu Search (TS) algorithms for VRPSDP. Ropke and Pisinger (2006) designed Large Neighborhood Search (LNS). Mu et al. (2016) introduced parallel Simulated Annealing (SA). Considering the time windows constraints, Angelelli and Mansini (2002) proposed an exact method and Mingyong and Erbao (2010); Wang and Chen (2012) proposed Genetic Algorithms. Liu et al. (2013) proposed both a Genetic Algorithm (GA) and a Tabu Search (TS) method. A few of researchers combined 2E-VRP with VRPSPD, such as 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) to solve it.

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 $A_1 = \{(i,j) \mid i,j \in V_0 \cup V_S\}$ is the set of first-echelon arcs, and $A_2 = \{(i,j) \mid i,j \in V_S \cup V_C, i \neq j\} \setminus \{(i,j) \mid i,j \in V_S, i \neq j\}$ 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 Section 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. Furthermore, no time window is considered for either satellites 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 τ. 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 set 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 a 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 Figure 1, firstly, the first-echelon vehicles (i.e., FV_1^d) 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_1^p) 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.

Figure 1 and 2 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 Figure 1, a first-echelon vehicle FV_1^d starts from CD, firstly arrives at S_1 at time $a_{S_1} = 12$. Then, in the second-echelon at S_1, FV_1^d 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, denote as SV_1, starts from S_1 at 17, and arrives at C_1 at time 22. Back to FV_1^d, it next immediately unloads cargoes for C_3 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_1^d, 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 Figure 2, FV_1^d starts from CD to visit S_1. Only the cargoes for SV_1 are transferred. Then, FV_1^d 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-VRPTW 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_1^p is not influenced, because there is no time constraint for the pickup stage of first-echelon vehicles.

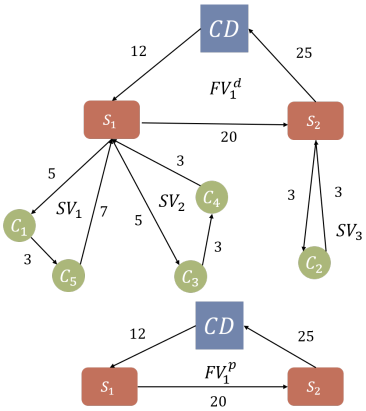
4 Solution Approach

To solve medium-to-large size 2E-VRPTWSPD instances, a variable neighborhood tabu search algorithm is implemented. The key ideas of the algorithm are as follows.

We try to improve the solution representation of satellites by providing 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 three vehicle routing stages, 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

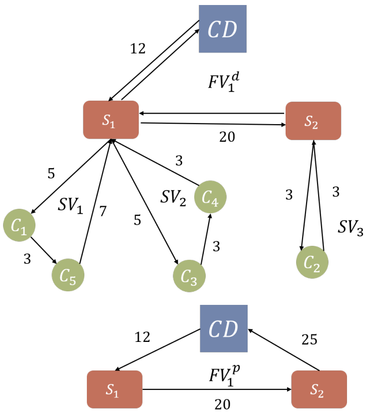
Solution representation is an important factor that affects the performance of a heuristic algorithm. We propose dummy satellites by splitting each satellite into several dummy satellites. Each dummy satellite only connects with one second-echelon tour and can only be served once.



	C_1	C_2	C_3	C_4	C_5
e_i	19	44	50	50	23
l_i	23	**49**	90	100	27
a_i	22	**56**	32	54	26
s_i	1	1	1	1	1
d_i	20	50	50	50	30

	S_1	S_2
a_i	12	47
s_i	15	6
d_i	150	60

Fig. 1 Example for visiting a satellite twice in one route(a)



	C_1	C_2	C_3	C_4	C_5
e_i	19	44	50	50	23
l_i	23	49	90	100	27
a_i	22	46	78	82	26
s_i	1	1	1	1	1
d_i	20	60	50	50	30

	S_{1-1}	S_2	S_{1-2}
a_i	12	37	63
s_i	5	6	10
d_i	50	60	100

Fig. 2 Example for visiting a satellite twice in one route(b)

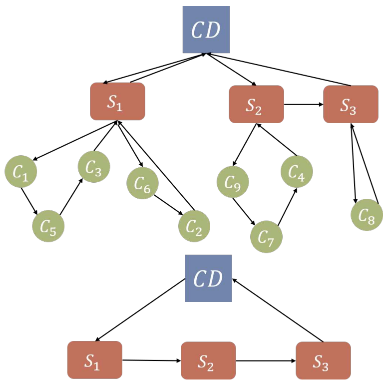


Fig. 3 Solution without split dummy satellites

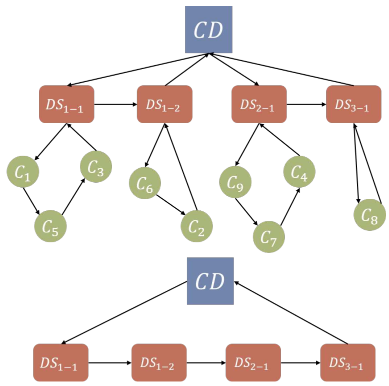


Fig. 4 Solution with split dummy satellites

Figure 4 shows an example of the solution with dummy satellites corresponding to the original solution in Figure 3. As in Figure 3, 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_{1-1} and DS_{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_{2-1} and DS_{3-1}.

In addition, we propose time windows for these dummy satellites. When the second-echelon route of a dummy satellite is determined, the latest arrival time of first-echelon vehicles to this satellite can also be determined. In other words, there is a deadline restriction for the first-echelon vehicle, and the deadline depends on how the second-echelon vehicles 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:

$$TS_i^j = \sum_{k=1}^{i-1} \max\{l_k^j - a_k^j, 0\} + l_i^j - a_i^j$$

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 TS^j to represent the minimum value for all customers in the second-echelon route starting from satellite j (denoted as $route(j)$):

$$TS^j = \min\{TS_i^j \mid i \in route(j)\}$$

Then we can get the latest arrival time to satellite j for first-echelon vehicles:

$$l_j = a_1^j + TS^j - c_{jj^*} - s^j$$

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

Figure 5 and 6 show examples of how to construct dummy satellite time windows. Each example includes some second-echelon route j (denoted "DS $\to$ customer 1 $\to$ customer 2 $\to$ ..."). Assume that we already have the route plan. The time line of each route is illustrated above the route. As shown in Figure 5, $TS_1^j = t_1$, $TS_2^j = t_2 + t_3$, $TS_3^j = t_2 + t_4$, $TS^j = \min\{TS_1^j, TS_2^j, TS_3^j\}$, $l_j = a_1 + TS^j - c_{j1} - s^j$. In Figure 6, 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 is explained in Section 4.4.2.

4.2 Search Space

In 2E-VRPTWSPD, if a route violates the maximum load constraint or time window constraints, 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:

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

Algorithm 1 Greedy create initial solution

Input: SList, CList
Output: feasibility, solution

```
classify customers to satellites (SList, CList);
RS1, feaisibility = 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;
```

$$d(s) = \sum_{k \in K_1} \max\{w^k - Q_1, 0\} + \sum_{k \in K_2} \max\{w^k - Q_1, 0\}$$

, 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. α and β are the corresponding weights. The weights are dynamically adjusted within an interval $[LB, UB]$, where LB and UB are predetermined by preliminary experiments.

Initially, α and β are randomly chosen within the interval. Whenever the vehicle capacity constraint or the time window constraint is violated, the respective weight (α or β) is multiplied by a parameter $\delta > 1$; when the solution is feasible, the respective weight is divided by δ. Note that the updated weight must also lie in the interval $[LB, UB]$.

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. $solution$ is the routing plan.

Three route sets RS1–RS3 are built. RS1 includes the second-echelon routes starting from a dummy satellite to serve customers. RS2 is the set of first-echelon routes starting from the depot to delivery. RS3 is the set of first-echelon routes from the depot to pickup cargoes on dummy satellites. The algorithm includes a classification phase and a construction phase.

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 CD to s and s to i.

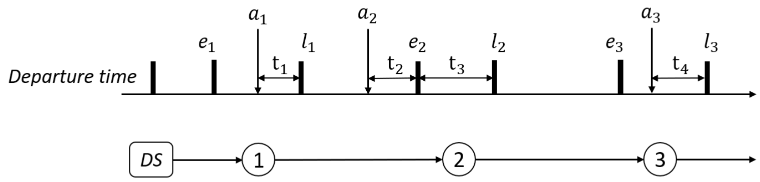


Fig. 5 Example of dummy satellite time windows (feasible)

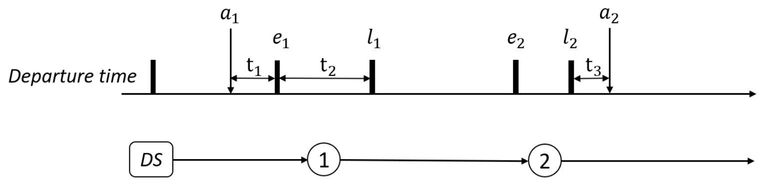


Fig. 6 Example of dummy satellite time windows (infeasible)

4.3.2 Construction of Routes

The problem for constructing RS1 is essentially a VRPTW. To construct a new route, a customer with the minimum of the latest-service-starting time is firstly inserted into the route. Then, the customer with the maximum savings is inserted step-wise. Before inserting customers into the route, a validity check is applied for the time windows of each customer in the route and the load for 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

Tabu Search (TS) is a memory-based search strategy to guide the local search to continue its search beyond local optimality Belhaiza et al. (2014); Glover (1990). 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.

Our Variable Neighborhood Tabu Search Algorithm is based on both TS and VNS. In each iteration of tabu search, it randomly selects two neighborhoods N1 and N2, and generates two neighboring solutions s_{N1} and s_{N2}. The better one is recorded by s^*, and then updates the global best solution s_{best}. Detailed procedures are shown in Algorithm 2.

Algorithm 2 Variable Neighborhood Tabu Search Algorithm

Input: An instance
Output: Best solution
  s_0 = Greedy();
  $s^* = s_0$, $s_{best} = s_0$;
  while the stopping condition is not met **do**
    N1 = a random neighborhood;
    N2 = another random neighborhood;
    s_{N1} = choose a best solution in N1(s^*);
    s_{N2} = choose a best solution in N2(s^*);
    s^* = best solution between s_{N1} and s_{N2};
    if s^* better than s_{best} **then**
      $s_{best} = s^*$;
    end if
    update tabu list;
  end while

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 two continuous segments of two different routes.

We choose to adopt $(1,0) - interchange$ and $(1,1) - interchange$. Figure 7 to 10 show several examples of $\lambda - interchange$ for the first-echelon and second-echelon routes, where rectangles represent the depot and dummy satellites, and circles represent customers.

Figure 7 shows 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 8 shows an example of $(1,1)-interchange$ operator. A customer of the former route (C_2) and a customer of the latter route (C_6) are exchanged. For the first-echelon operators, when the dummy satellites are moved, its associated customers are moved as well. Take Figure 9 as an example, C_3 is inserted into the latter route with dummy satellite DS_2.

In sum, $(1,0)-interchange$ and $(1,1)-interchange$ are applied in all the three route planning stages to obtain 6 different operators, named O^{1d}_{1-0}, O^{1d}_{1-1}, O^{2}_{1-0}, O^{2}_{1-1}, O^{1p}_{1-0}, and O^{1p}_{1-1}.

4.4.2 Solution Evaluate 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 first-echelon vehicles' transfer time in satellites will be changed, and the corresponding first-echelon route will be affected. Therefore, it will cost a lot of time to evaluate new generated solutions.

We therefore use the dummy time windows mentioned above to improve the evaluation process of solutions. For each dummy satellite, if its corresponding second-echelon route is determined, the dummy time window is kept unchanged. As 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 some dummy satellite whose second-echelon route is 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 for 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 approaches 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 used to solve multi-depot 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 obtained 2E-VRPTWSPD instances. The pickup demand of customers is half of their delivery demand. We set $Q_1 = 125$,$Q_2 = 50$ and $\tau = 0.5$. The scales of instances are 7, 10, 12 customers with 2 satellites, each type has 6 instances. Each instance is represented 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 instances proposed by Hemmelmayr et al. (2012) for 2E-VRP. The instances scale is 100 customers with 5 satellites, 100 customers with 10 satellites, 200 customers with 10 satellites, each type 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 demand. Notice that we keep the original notations, like "100-10-1" or "100-10-1b". There is no obvious similarity 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 hour. For the heuristic algorithm, we set the stop principle of both maximum number of iterations (I_1) and maximum number of iterations without improving the best solution (I_2). For small-scale instances, we set $I_1 = 1000$ and $I_2 = 200$. For large-scale instances, we set $I_1 = 25000$ and $I_2 = 5000$.

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 the results of small-scale instances 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 (Obj_E) and computation time (T_E) of the exact solution obtained by CPLEX. Columns 4 and 5 show the minimum result of objective value executed 10 times (Obj_{MinH}) and the percentage gap between columns 2 and 4 (GAP1). Columns 6, 7, 8 and 9 show the maximum objective value (Obj_{MaxH}),

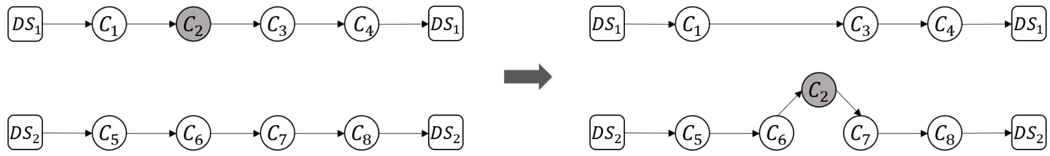


Fig. 7 $(1-0)-$*interchange* operators for the second-echelon route.

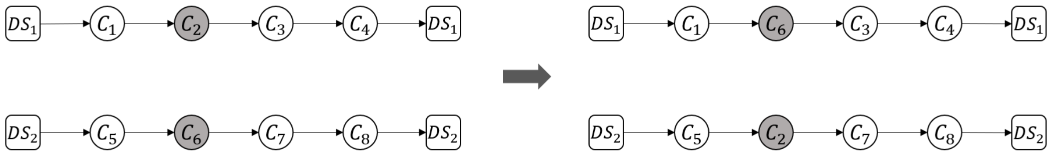


Fig. 8 $(1-1)-$*interchange* operators for the second-echelon route.

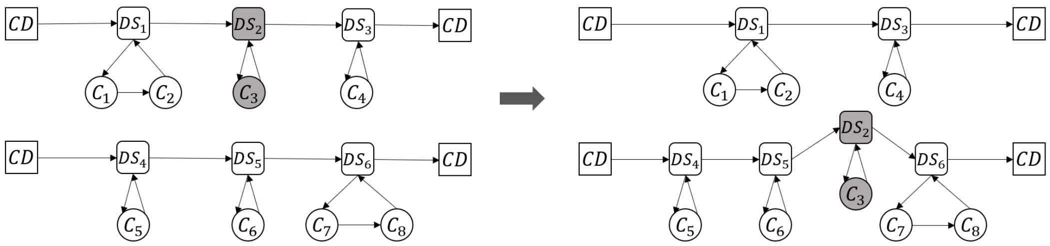


Fig. 9 $(1-0)-$*interchange* operators for the first-echelon route.

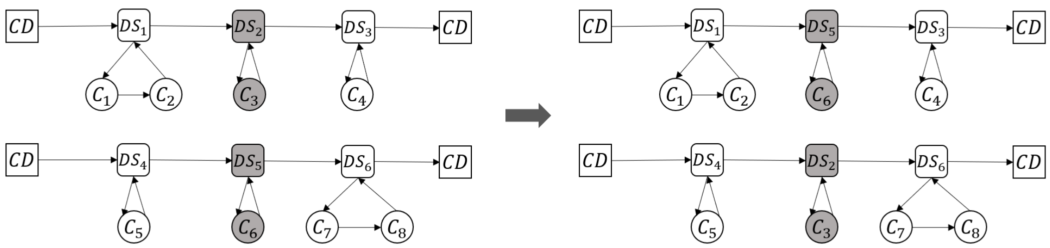


Fig. 10 $(1-1)-$*interchange* operators for the first-echelon route.

percentage gap between columns 2 and 6 (GAP2), average objective value (Obj_{MaxH}), and the percentage gap between columns 2 and 7 (GAP3). Column 10 shows the computation time (T_H) of the heuristic solution.

The performance measurements considered in the comparison are: (1) Percentage Gap (Gap): calculated as $100\% * (Obj_H - Obj_E)/Obj_E$, where Obj_H 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 instances 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 0.05s.

5.3 Results on Large-scale Instances

The comparison of heuristic solutions with exact solutions on small-scale instances shows 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 or 200 customers and 10 or 15 satellites. The first column of Table 2 displays the instance name. The second to eighth columns report the objective value, number of the first-echelon routes for delivery, number of second-echelon routes, the objective value for first-echelon in distribution, the objective value for second-echelon and the objective value for first-echelon in collection, 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 minute; for 200 customers, the solutions are

Table 1 Exact and heuristic results on small-scale instances.

Instance	Formulation		Heuristic						
	Obj_E	T_E	Obj_{MinH}	GAP1	Obj_{MaxH}	GAP2	Obj_{AveH}	GAP3	T_H
7-2-1	315.74	2.44	318.61	0.91	318.61	0.91	318.61	0.91	0.002
7-2-2	310.90	1.94	310.90	0.00	310.90	0.00	310.90	0.00	0.004
7-2-3	253.20	1.02	253.20	0.00	253.20	0.00	253.20	0.00	0.002
7-2-4	303.02	89.51	303.02	0.00	303.02	0.00	303.02	0.00	0.016
7-2-5	326.45	153.96	326.45	0.00	326.45	0.00	326.45	0.00	0.006
7-2-6	355.23	137.53	355.23	0.00	355.23	0.00	355.23	0.00	0.010
Ave				0.15		0.15		0.15	
10-2-1	280.33	2052.14	280.33	0.00	280.33	0.00	280.33	0.00	0.010
10-2-2	353.31	3600.00	353.31	0.00	353.31	0.00	353.31	0.00	0.013
10-2-3	286.64	3600.00	286.64	0.00	286.64	0.00	286.64	0.00	0.029
10-2-4	385.36	3600.00	385.36	0.00	385.36	0.00	385.36	0.00	0.015
10-2-5	417.73	3600.00	406.51	-2.69	413.40	-1.04	407.28	-2.50	0.020
10-2-6	453.01	3600.00	422.82	-6.66	468.02	3.31	427.44	-5.65	0.041
Ave				-1.56		0.38		-1.36	
12-2-1	505.96	3600.00	362.98	-28.26	362.98	-28.26	362.98	-28.26	0.023
12-2-2	484.73	3600.00	464.18	-4.24	521.18	7.52	499.73	3.10	0.020
12-2-3	231.81	8.90	231.81	0.00	231.81	0.00	231.81	0.00	0.018
12-2-4	488.63	3600.00	375.20	-23.21	379.12	-22.41	377.95	-22.65	0.023
12-2-5	469.75	3600.00	410.37	-12.64	483.54	2.93	445.40	-5.19	0.023
12-2-6	505.96	3600.00	362.98	-28.26	362.98	-28.26	362.98	-28.26	0.021
Ave				-16.10		-11.41		-13.54	

Table 2 Heuristic results of large-scale instances

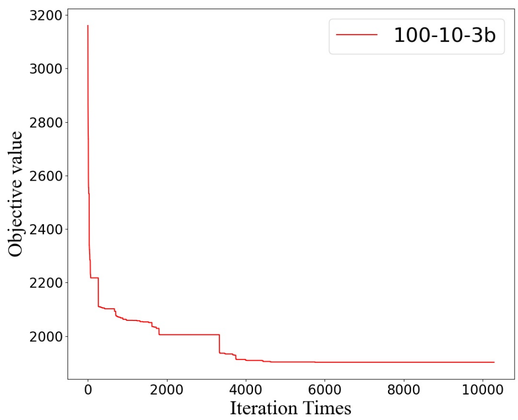
Instance	Obj	n_1	n_2	n_3	Obj_1	Obj_2	Obj_3	T
100-5-1	2293.21	4	27	2	551.50	1445.38	296.34	44.37
100-5-1b	2060.28	5	17	2	692.86	1071.08	296.34	56.88
100-5-2	1952.62	5	28	2	539.62	1149.64	263.36	27.97
100-5-2b	1473.74	4	17	2	431.85	778.53	263.36	12.39
100-5-3	1781.76	5	27	2	604.11	909.37	268.28	36.52
100-5-3b	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	25.60
100-10-1b	1845.14	6	19	2	620.35	968.58	256.20	53.11
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	1686.65	4	25	2	479.07	926.49	281.09	45.50
100-10-3b	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

astringed within about 5 minutes. We further provide the convergence plot for two selected instances "100-10-3b" and "200-10-3" in Figure 11. The plot indicates the fast convergence of the algorithm in the first few iterations.

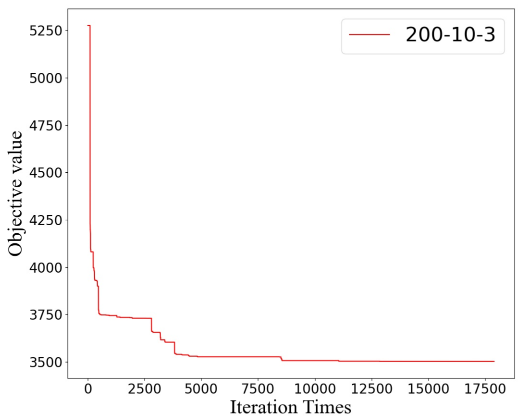
Finally, we find that the running time of the algorithm depends more on the number of customers than the number of satellites, because the scale of customers is much larger than that of satellites. For large-scale instances, the distance traveled by the second-echelon vehicle is larger than the first-echelon vehicle. Compared with the third stage, the first stage delivery requires more vehicle paths due to the deadline constraints and larger delivery demand, thus its objective function Obj_1 is 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 Section 4. To verify the importance of each operator, we respectively removed one of the operators in $\{O^{1d}_{1-0}, O^{1d}_{1-1}, O^{2}_{1-0}, O^{2}_{1-1}, O^{1p}_{1-0}, O^{1p}_{1-1}\}$ from our algorithm to generate 6 variants, and then compared the result of these variants with the original algorithm.



(a) Instance 100-10-3b



(b) Instance 200-10-3

Fig. 11 The convergence plot for two instances.

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. Columns 2 to 7 indicate the percentage gap between our algorithm and its six variants, denoted as $GAP1$ to $GAP6$. A negative value of $GAPn$ 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 implies that excluding the use of any operator will impair the solution quality.

Table 3 Gaps of the proposed algorithm and its six variants.

Instance	GAP1	GAP2	GAP3	GAP4	GAP5	GAP6
100-5-1	32.20	7.84	22.00	10.88	11.88	4.47
100-5-1b	33.63	6.10	35.96	8.73	9.20	10.03
100-5-2	21.45	5.11	32.67	4.10	6.74	7.66
100-5-2b	19.90	0.67	25.59	2.68	9.49	6.51
100-5-3	25.87	-0.24	26.87	7.96	13.79	6.07
100-5-3b	25.04	1.34	21.41	11.05	7.88	9.85
100-10-1	11.04	2.45	18.67	5.24	3.58	6.55
100-10-1b	11.79	-0.15	29.84	2.05	12.39	4.80
100-10-2	16.35	4.72	21.91	14.38	7.02	10.00
100-10-2b	15.03	5.88	34.64	6.99	2.69	3.21
100-10-3	6.63	6.54	18.30	3.38	8.88	2.88
100-10-3b	16.23	-1.21	29.35	2.62	5.71	2.07
200-10-1	9.60	1.96	24.53	12.28	10.94	5.87
200-10-1b	18.18	10.66	36.68	9.81	13.41	10.73
200-10-2	10.91	1.26	15.39	3.69	3.26	3.37
200-10-2b	9.55	-0.02	26.95	7.92	5.78	4.97
200-10-3	10.02	3.90	17.87	11.07	5.78	7.24
200-10-3b	29.28	10.34	35.89	10.51	16.54	10.43

Table 4 Time cost for heuristic with or without satellite time windows

Instance	T_0	T_1	GAP
100-5-1	58.27	72.00	23.57
100-5-1b	59.32	63.82	7.58
100-5-2	56.23	69.74	24.02
100-5-2b	55.63	61.52	10.60
100-5-3	58.31	67.73	16.16
100-5-3b	55.63	60.58	8.90
100-10-1	56.92	67.89	19.26
100-10-1b	58.46	64.01	9.51
100-10-2	56.57	65.72	16.19
100-10-2b	58.22	61.20	5.12
100-10-3	56.25	67.24	19.52
100-10-3b	58.87	68.30	16.01
200-10-1	251.96	324.34	28.73
200-10-1b	259.35	290.85	12.15
200-10-2	249.78	335.35	34.26
200-10-2b	255.33	286.17	12.08
200-10-3	253.92	370.01	45.72
200-10-3b	255.88	339.62	32.73
Ave			19.00

5.5 Impact of the Satellite Time Windows

As described in Section 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 $100\% * (T_1 - T_0)/T_0$. The last row summarizes the average result over 18 instances.

From this table, we find that the algorithm with satellite time windows is faster on all 18 instances. To be specific, the satellite time windows is 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 to accelerate 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 of small and large scale based on the existing instance sets. The results show that our heuristic approach is effective and efficient to find good solutions for 2E-VRPTWSPD. Furthermore, we show by statistical analysis that our strategies of combing multiple neighborhood operators and including the usage of satellite time windows can significantly improve the performance and speed of the heuristic.

Our future research on 2E-VRPTWSPD will focus on the design of more powerful valid inequalities and exact algorithms. Branch-and-price or other algorithms based on column generation 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 to optimality.

Funding This work was supported by Guangdong Natural Science Funds (No. 2019A1515011169, 2021A1515011301).

Compliance with ethical standards

Conflicts 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.

Authorship contributions

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

A Arc-Based Formulation for 2E-VRPTWSPD

To model the 2E-VRPTWSPD, we introduce dummy node sets V_{DS}^{s} for each satellite s. $V_S' = \underset{s\in V_S}{\cup} V_{\mathrm{DS}}^{s}$ represents node set of the first echelon. Let $A_1' = \left\{(i,j) \mid i,j \in V_0 \cup V_S', i \neq j\right\}$ and $A_2' = \{(i,j) \mid i,j \in V_S' \cup V_C, i \neq j\} \setminus \{(i,j) \mid i,j \in V_S', i \neq j\}$.

Inspired by the ideas from Liu et al. (2018) and Li et al. (2019), we first 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_s^{1k}: a decision variable representing the quantity delivered to satellite $s \in V_S'$ by vehicle $k \in K_1$;
- w_s^{2k}: a decision variable representing the quantity collected to satellite $s \in V_S'$ by vehicle $k \in K_1$;
- u_s^{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_s^{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}^{v} : 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_s^k : 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_i^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:

$$\min \sum_{k\in K_1} \sum_{(i,j)\in A_1'} c_{ij}\left(x_{ij}^{1k} + x_{ij}^{2k}\right) + \sum_{v\in K_2} \sum_{(i,j)\in A_2'} c_{ij} y_{ij}^{v} \tag{1}$$

subject to

$$\sum_{(i,j)\in A_1'} x_{ij}^{1k} = \sum_{(j,i)\in A_1'} x_{ji}^{1k}, \quad \forall i \in V_0 \cup V_S', k \in K_1 \tag{2}$$

$$\sum_{(i,j)\in A_1'} x_{ij}^{2k} = \sum_{(j,i)\in A_1'} x_{ji}^{2k}, \quad \forall i \in V_0 \cup V_S', k \in K_1 \tag{3}$$

$$\sum_{(0,j)\in A_1'} x_{0j}^{1k} \leq 1, \quad \forall k \in K_1 \tag{4}$$

$$\sum_{(0,j)\in A_1'} x_{0j}^{2k} \leq 1, \quad \forall k \in K_1 \tag{5}$$

$$\sum_{k\in K_1} \sum_{(i,j)\in A_1'} x_{ij}^{1k} \leq 1, \quad \forall i \in V_S' \tag{6}$$

$$\sum_{k\in K_1} \sum_{(i,j)\in A_1'} x_{ij}^{2k} \leq 1, \quad \forall i \in V_S' \tag{7}$$

$$u_i^{1k} + 1 \leq u_j^{1k} + M\left(1 - x_{ij}^{1k}\right), \quad \forall i \in V_S', j \in V_S', k \in K_1 \tag{8}$$

$$1 \leq u_j^{1k} + M\left(1 - x_{0j}^{1k}\right), \quad \forall j \in V_S', k \in K_1 \tag{9}$$

$$u_i^{2k} + 1 \leq u_j^{2k} + M\left(1 - x_{ij}^{2k}\right), \quad \forall i \in V_S', j \in V_S', k \in K_1 \tag{10}$$

$$1 \leq u_j^{2k} + M\left(1 - x_{0j}^{2k}\right), \quad \forall j \in V_S', k \in K_1 \tag{11}$$

$$w_s^{1k} \leq M \sum_{(s,i)\in A_1'} x_{si}^{1k}, \quad \forall s \in V_S', k \in K_1 \tag{12}$$

$$w_s^{2k} \leq M \sum_{(s,i)\in A_1'} x_{si}^{2k}, \quad \forall s \in V_S', k \in K_1 \tag{13}$$

$$\sum_{s\in V_S'} w_s^{1k} \leq Q_1, \quad \forall k \in K_1 \tag{14}$$

$$\sum_{s\in V_S'} w_s^{2k} \leq Q_1, \quad \forall k \in K_1 \tag{15}$$

$$\sum_{(i,j)\in A_2'} y_{ij}^{v} = \sum_{(j,i)\in A_2'} y_{ji}^{v}, \quad \forall i \in V_C \cup V_S', v \in K_2 \tag{16}$$

$$\sum_{v\in K_2} \sum_{(j,i)\in A_2'} y_{ij}^{v} = 1, \quad \forall i \in V_C \tag{17}$$

$$\sum_{v\in K_2} \sum_{j\in V_C} y_{ij}^{v} = \sum_{k\in K_1} \sum_{(i,j)\in A_1'} x_{ij}^{1k}, \quad \forall i \in V_S' \tag{18}$$

$$\sum_{i\in V_S'} \sum_{(i,j)\in A_2'} y_{ij}^{v} \leq 1, \quad \forall v \in K_2 \tag{19}$$

$$\sum_{(j,i)\in A_2'} f_{ji} = \sum_{(i,j)\in A_2'} f_{ij} + d_i, \quad \forall i \in V_C \tag{20}$$

$$\sum_{(j,i)\in A_2'} g_{ji} = \sum_{(i,j)\in A_2'} g_{ij} - p_i, \quad \forall i \in V_C \tag{21}$$

$$\sum_{k\in K_1} w_s^{1k} = \sum_{(s,i)\in A_2'} f_{si}, \quad \forall s \in V_S' \tag{22}$$

$$\sum_{k\in K_1} w_s^{2k} = \sum_{(s,i)\in A_2'} g_{is}, \quad \forall s \in V_S' \tag{23}$$

$$s_s^k = \tau * w_s^{1k}, \quad \forall s \in V_S', k \in K_1 \tag{24}$$

$$\begin{aligned}
&(d_j - p_j) * \sum_{v\in K_2} y_{ij}^{v} \leq f_{ij} + g_{ij} \\
&\leq Q_2 * \sum_{v\in K_2} y_{ij}^{v} + (p_i - d_i) * \sum_{v\in K_2} y_{ij}^{v} \\
&\forall (i,j) \in A_2'
\end{aligned} \tag{25}$$

$$\begin{aligned}
&a_j^k \geq a_i^k + c_{ij} + s_i^k - M\left(1 - x_{ij}^{1k}\right), \\
&\quad \forall i \in V_S', j \in V_S', k \in K_1
\end{aligned} \tag{26}$$

$$\begin{aligned}
&a_j^k \geq c_{0j} - M\left(1 - x_{0j}^{1k}\right), \\
&\quad \forall j \in V_S', k \in K_1
\end{aligned} \tag{27}$$

$$\begin{aligned}
&a_j^v \geq a_i^k + c_{ij} + s_i^k - M\left(2 - y_{ij}^{v} - \sum_{(h,i)\in A_1'} x_{hi}^{1k}\right), \\
&\quad \forall i \in V_S', j \in V_C, k \in K_1, v \in K_2
\end{aligned} \tag{28}$$

$$\begin{aligned}
&a_j^v \geq a_i^v + c_{ij} + s_i - M\left(1 - y_{ij}^{v}\right), \\
&\quad \forall i \in V_C, j \in V_C, v \in K_2
\end{aligned} \tag{29}$$

$$a_i^v \geq e_i, \quad \forall i \in V_C, v \in K_2 \tag{30}$$
$$a_i^v \leq l_i, \quad \forall i \in V_C, v \in K_2 \tag{31}$$
$$x_{ij}^{1k}, x_{ij}^{2k} \in \{0,1\}, \quad \forall (i,j) \in A_1', k \in K_1 \tag{32}$$
$$w_s^{1k}, w_s^{2k}, u_s^{1k}, u_s^{2k} \geq 0, \quad \forall s \in V_S', k \in K_1 \tag{33}$$
$$f_{ij}, g_{ij} \geq 0, \quad \forall (i,j) \in A_2' \tag{34}$$
$$y_{ij}^v \in \{0,1\}, \quad \forall (i,j) \in A_2', v \in K_2 \tag{35}$$
$$s_s^k, a_s^k \geq 0, \quad \forall s \in V_S', k \in K_1 \tag{36}$$
$$a_i^v \geq 0, \quad \forall i \in V_C, v \in K_2 \tag{37}$$

The objective function (1) minimizes the sum of the first-echelon and second-echelon traveling cost. Constraints (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

Angelelli E, Mansini R (2002) The vehicle routing problem with time windows and simultaneous pick-up and delivery. In: Quantitative approaches to distribution logistics and supply chain management, Springer, pp 249–267

Belgin O, Karaoglan I, Altiparmak F (2018) Two-echelon vehicle routing problem with simultaneous pickup and delivery: Mathematical model and heuristic approach. Computers & Industrial Engineering 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. Computers & Operations Research 52:269–281

Bianchessi N, Righini G (2007) Heuristic algorithms for the vehicle routing problem with simultaneous pick-up and delivery. Computers & Operations Research 34(2):578–594

Breunig U, Schmid V, Hartl RF, Vidal T (2016) A large neighbourhood based heuristic for two-echelon routing problems. Computers & Operations Research 76:208–225

Chen JF, Wu TH (2006) Vehicle routing problem with simultaneous deliveries and pickups. Journal of the Operational Research Society 57(5):579–587

Cordeau JF, Laporte G, Mercier A (2001) A unified tabu search heuristic for vehicle routing problems with time windows. Journal of the Operational research society 52(8):928–936

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

Crainic TG, Mancini S, Perboli G, Tadei R (2011) Multi-start heuristics for the two-echelon vehicle routing problem. In: European Conference on Evolutionary Computation in Combinatorial Optimization, Springer, pp 179–190

Crainic TG, Mancini S, Perboli G, Tadei R (2013) Grasp with path relinking for the two-echelon vehicle routing problem. In: Advances in Metaheuristics, Springer, pp 113–125

Crispim J, Brandão J (2005) Metaheuristics applied to mixed and simultaneous extensions of vehicle routing problems with backhauls. Journal of the Operational Research Society 56(11):1296–1302

Cuda R, Guastaroba G, Speranza MG (2015) A survey on two-echelon routing problems. Computers & Operations Research 55:185–199

Dellaert N, Dashty Saridarq F, Van Woensel T, Crainic TG (2019) Branch-and-price–based algorithms for the two-echelon vehicle routing problem with time windows. Transportation Science 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. Computers & Operations Research 127:105154

Glover F (1990) Tabu search—part ii. ORSA Journal on computing 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. Computers & operations research 39(12):3215–3228

Li H, Bai M, Zhao Y, Dai C (2019) Vehicle flow formulation for two-echelon time-constrained vehicle routing problem. Journal of Management Science and Engineering 4(2):75–90

Li H, Wang H, Chen J, Bai M (2020) Two-echelon vehicle routing problem with time windows and mobile satellites. Transportation Research Part B: Methodological 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. European Journal of Operational Research 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. European Journal of Operational Research 266(2):487–497

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

Mingyong L, Erbao C (2010) An improved differential evolution algorithm for vehicle routing problem with simultaneous pickups and deliveries and time windows. Engineering Applications of Artificial Intelligence 23(2):188–195

Mladenović N, Hansen P (1997) Variable neighborhood search. Computers & operations research 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. International Journal of Shipping and Transport Logistics 8(1):81–106

Perboli G, Tadei R (2010) New families of valid inequalities for the two-echelon vehicle routing problem. Electronic notes in discrete mathematics 36:639–646

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

Ropke S, Pisinger D (2006) A unified heuristic for a large class of vehicle routing problems with backhauls. European Journal of Operational Research 171(3):750–775
Solomon MM (1987) Algorithms for the vehicle routing and scheduling problems with time window constraints. Operations research 35(2):254–265
Wang HF, Chen YY (2012) A genetic algorithm for the simultaneous delivery and pickup problems with time window. Computers & Industrial Engineering 62(1):84–95
Zeng Zy, Xu Ws, Xu Zy, Shao Wh (2014) A hybrid grasp+ vnd heuristic for the two-echelon vehicle routing problem arising in city logistics. Mathematical Problems in Engineering 2014