A variable neighborhood search algorithm with constraint relaxation for the two-echelon vehicle routing problem with simultaneous delivery and pickup demands

Ran Liu¹ • Shan Jiang²

Accepted: 16 December 2021 / Published online: 10 January 2022
© The Author(s), under exclusive licence to Springer-Verlag GmbH Germany, part of Springer Nature 2022

Abstract
This paper considers a special vehicle routing problem, the two-echelon vehicle routing problem with simultaneous delivery and pickup demands (2E-VRPSDP). The 2E-VRPSDP differs from classic transportation and vehicle routing problems in two ways. First, freight delivery from the depot to the customers is managed by shipping the freight through intermediate satellites. Second, each customer in the 2E-VRPSDP may have simultaneous delivery and pickup demands. The 2E-VRPSDP is an extension of the two-echelon vehicle routing problem (2E-VRP) and the vehicle routing problem with simultaneous delivery and pickup (VRPSDP). A variable neighborhood search algorithm is designed to solve the 2E-VRPSDP in which both feasible and infeasible solutions can be explored. Numerical results show that the proposed algorithm is effective and that the algorithm can provide reasonable solutions within an acceptable computational time.

Keywords Constraint relaxation · Variable neighborhood search · Vehicle routing · Two-echelon · Delivery and pickup

1 Introduction
In the past decade, the two-echelon vehicle routing problem (2E-VRP) has become a new interest in the vehicle routing problems (VRPs) research field. The logistics network of 2E-VRP is composed of two echelons, i.e., the freight is delivered from the depot to the satellites by large capacity vehicles in the first echelon and then transported from the satellites to the customers by the second echelon vehicles of relatively small capacity. The objective of the 2E-VRP is to minimize the total transportation costs of both echelons. The 2E-VRP problem arises in many practical transportation and distribution contexts, such as city logistics applications (Feliu et al. 2007; Jepsen et al. 2013). The 2E-VRP has been studied by many researchers, and some very good studies can be found for this problem (Perboli et al. 2011; Jepsen et al. 2013; Breunig et al. 2016; Liu et al. 2017). In the existing research, generally, it is assumed that one customer only has either delivery or pickup demand. However, in some practical applications, the vehicle needs to distribute and collect freight simultaneously. We consider a practical two-echelon logistical problem arising in the home healthcare (HHC) industry in Shanghai, China. Many HHC companies have been built in Shanghai for patients who require long and regular healthcare services at their homes. The typical services in the HHC involve various logistic activities, including delivering medicines and medical instruments to patients, picking up biological samples from patients’ homes, and collecting medical waste for disposal. Considering the HHC company, the core component in-home healthcare problems is to find a feasible working schedule for their drivers and vehicles to reduce the operating cost. For a large city, because customers spread in a very large area, one HHC company establishes some subcompanies in different districts, and daily logistic services such as delivering medicines and collecting medical waste are implemented as a two-echelon logistics system. First, in the morning, the
goods to be delivered to the customers (e.g., the medicines for the customers) are transported from the company (or center warehouse) to the subcompanies. Next, small subcompany vehicles deliver goods to customers and pick up goods from customers (e.g., collecting medical waste) to take back to the subcompanies. After all the pickup goods are collected in subcompanies, finally, the vehicles at the HHC company pick up such goods from subcompanies to return to the HHC company and complete one day’s operations. Clearly, this is a special two-echelon transportation network in which the first echelon consists of the HHC company and subcompanies, and the second echelon contains subcompanies and customers. We can further observe that, unlike the classical 2E-VRP, the operations by the HHC company consist of three phases. The HHC company must design a set of routes for vehicles in this two-echelon network. According to the HHC company investigations, it is not easy for the company’s manager and planner. This is because this routing optimization problem is rather complex, which can be seen as a combination of two challenging problems, i.e., the two-echelon vehicle routing problem (2E-VRP) and the vehicle routing problem with simultaneous delivery and pickup (VRPSDP), because the problem has a two-echelon network structure and the customers have both delivery and pickup demands simultaneously. Since both the VRPSDP and the 2E-VRP are NP-hard, the problem stated above is also NP-hard and even more complex and harder than these two problems.

In this paper, we focus on this special logistical problem, which is named the two-echelon vehicle routing problem with simultaneous delivery and pickup (2E-VRPSDP). Although the 2E-VRP and VRPSDP have been studied for more than ten years, as stated above, the 2E-VRPSDP solution structure differs from the classical problems (the detailed differences are shown and analyzed in Sect. 2.) To our knowledge, there is no literature about the 2E-VRPSDP proposed in this paper. Because of the difficulties in solving instances of practical interest, we propose a variable neighborhood search (VNS) heuristic in which both feasible and infeasible solutions are explored to enhance the algorithmic search ability. Numerical results show that the VNS algorithm can efficiently solve the 2E-VRPSDP.

The rest of this paper is organized as follows. Section 2 defines the problem and discusses the differences with existing problems that have been studied in the literature. Section 3 builds the mathematical model for the 2E-VRPSDP. Section 4 reviews the relevant literature. Section 5 describes the VNS algorithm. Computational results are reported in Sect. 6. Section 7 concludes the paper.

2 Problem definition

The 2E-VRPSDP can be defined formally as follows. Let $G = (V, E)$ be a graph with the node set $V = V_0 \cup V_S \cup V_C$ and edge set $E = E_1 \cup E_2$. In set $V$, $V_0 = \{0\}$ is the depot, set $V_S = \{1, \ldots, |V_S|\}$ represents the set of satellites where $|V_S|$ is the cardinality of set $V_S$, and set $V_C = \{|V_S|+1, \ldots, |V_S|+|V_C|\}$ is the set of customers. Each customer $i \in V_C$ has a delivery demand and a pickup demand simultaneously, denoted as $d_i$ and $p_i$, respectively. A fleet of $K_1$ homogeneous first echelon vehicles with a capacity of $Q_1$ is located at depot $V_0$, which can visit the depot and satellites. Additionally, a total of $K_2$ homogeneous second echelon vehicles with a capacity of $Q_2$ are located at all satellites, which can only visit the satellite and customers, and $Q_1 > Q_2$. Set $E$ is divided into two subsets, $E_1$ and $E_2$. Set $E_1 = \{(i, j) : i, j \in V_S \cup V_C, i, j \not\in V_S \times V_S\}$ corresponds to the edges between the depot and the satellites and between different satellites, which can only be traveled by the first echelon vehicles. Set $E_2 = \{(i, j) : i, j \in V_S \cup V_C, i, j \not\in V_S \times V_S\}$ represents the edges connecting the satellites and the customers and those connecting different customers. Each edge $(i, j) \in E$ has a nonnegative cost (distance), denoted as $c_{ij}$, and for each pair $(i, j) \in E$, $c_{ij} = c_{ji}$.

Note that a solution of the 2E-VRPSDP consists of three phases. In the first phase, a number of first echelon vehicles start from depot $V_0$ and deliver goods to a sequence of satellites and then return to the depot. For notational convenience, each of these routes is called the first phase delivery route. Then, in the second phase, second echelon vehicles start from the satellites, each meeting the delivery and pickup demands of one or several customers, and return to the starting satellites. Now, all the pickup demands from the customers are stored at the satellites. Similarly, each route at this echelon is referred to as a second phase delivery and pickup route. Finally, again, the first echelon vehicles depart from the depot and collect all pickup goods from the satellites and return back to the central depot. Such routes are called third phase pickup routes. Note that at each phase, direct vehicle routes between the depot and customers are forbidden. At each phase, some constraints must be respected as follows.

1. Each first phase delivery route and third phase pickup route start and end at the depot;
2. Each route is assigned to exactly one vehicle;
3. At any time, the total load of the vehicle cannot exceed the vehicle capacity;
4. A customer can be served by only one second phase delivery and pickup route; a satellite can be served by at most one first phase delivery route and at most one third phase pickup route. This constraint means that the demands of both echelons cannot be split.
The objective of 2E-VRPSDP is to determine the vehicle routes of both echelons to the sum of the routing costs, i.e., the total travel distances of all vehicle routes in the above three phases are minimized.

Compared with the 2E-VRPSDP, the 2E-VRP has only two-phase operations: first, the goods to be delivered to the customers are transported from the depot to the satellites by large vehicles; next, small vehicles starting from satellites deliver goods to customers and complete the operations (Feliu et al. 2007; Jepsen et al. 2013; Hemmelmayr et al. 2012). To clearly illustrate the 2E-VRPSDP and compare it with the 2E-VRP, a solution to the 2E-VRP with one depot, three satellites (S₁–S₃), and seven customers (C₁–C₇) is shown in Fig. 1, which consists of three parts A–C. Each customer has pickup and delivery demands. First, as shown in part A of Fig. 1, the company distributes the goods to the satellites by two first echelon delivery routes (blue lines). Next, in part B, three vehicles start from each satellite and deliver goods to each customer and collect pickup goods to deliver back to the satellites (black dotted line). Finally, part C shows two routes that collect all the goods to deliver to the depot (red lines).

Belgin et al. (2018) introduced an interesting two-echelon vehicle routing problem with simultaneous pickup and delivery requests. In their problem, the customers have pickup and delivery demands, and the pickup and delivery activities are performed simultaneously by the same vehicles through satellites to customers in the second echelon. Belgin et al. (2018) assumed that in the first echelon, goods are delivered from the central depot to each satellite and collected from satellites to deliver back to the depot by the same vehicles. That is, a solution to the problem of Belgin et al. (2018) consists of only two steps: The delivery routes and pickup routes between the depot and satellites (as shown in parts A and C of Fig. 1) are combined. Furthermore, we note that many other researchers also state that pickup and delivery in two-echelon city logistics can improve the efficiency if it is performed simultaneously on the first echelon vehicles. For example, Gianessi et al. (2016) solved the related tactical planning problems; for tactical planning problems, please refer to Fontaine et al. (2017, 2021). We agree that the operations in these works are reasonable in practical applications. We investigate many logistics and home healthcare companies and find that they do not adopt a two-step model because, in some scenarios, pickup and delivery operations for a satellite are not easy to perform at the same time. For example, one first echelon vehicle a reaches satellite i at 8:00 and unloads the goods for this satellite. Next, a second echelon vehicle b delivers such goods to customers and returns to this satellite at 10:00, with all pickup demands from its customers. Following the assumption of Belgin et al. (2018), vehicle a has to wait at satellite i until vehicle b returns to this satellite, i.e., two hours in this example. In practice, some companies deliver goods from a central depot to satellites in the morning and collect goods from satellites to deliver back to the depot in the afternoon, consistent with the problem definition in this paper.

### 3 Mathematical formulation

The mathematical formulation of the 2E-VRPSDP (mixed-integer linear program, MILP) is presented below.

**Decision variables:**

\[ y_{i,j} \] binary variable equal to 1 if a vehicle travels directly from node i to node j at a first phase delivery route;

\[ x_{i,j} \] binary variable equal to 1 if a vehicle travels directly from node i to node j at a second phase delivery and pickup route;
\( l_{ij} \) binary variable equal to 1 if a vehicle travels directly from node \( i \) to node \( j \) at a third phase pickup route;

\( z_{ij} \) binary variable equal to 1 if customer \( i \) is assigned to satellite \( j \);

\( q_i \) quantity of goods to be delivered and loaded on the second phase vehicle until customer \( i \) is visited;

\( b_i \) quantity of pickup goods loaded on the second phase vehicle immediately after customer \( i \) is visited;

\( a_i \) quantity of demands to be delivered at satellite \( i \);

\( \tilde{a}_i \) quantity of demands to be picked up at satellite \( i \);

\( A_i \) quantity of goods to be delivered to satellites loaded on the first phase vehicle until satellite \( i \) is visited;

\( B_i \) quantity of pickup goods loaded on the third phase vehicle immediately after satellite \( i \) is visited;

**Objective function:**

\[
\begin{align*}
\text{Min} & \quad \sum_{i \in V_C} \sum_{j \in V_C} c_{ij}x_{ij} + \sum_{i \in V_C} \sum_{j \in V_C} c_{ij}(y_{ij} + l_{ij}) \\
\text{Subject to,} & \quad \sum_{j \in V_C} x_{ij} = 1 \quad \forall j \in V_C \quad (2) \\
& \quad \sum_{j \in V_C} x_{ij} = \sum_{j \in V_C} x_{ji} \quad \forall j \in V_C \quad (3) \\
& \quad a_i + d_i + Q_2 x_{ij} \leq Q_2 + a_i \quad \forall i, \quad j \in V_C, \quad i \neq j \quad (4) \\
& \quad b_i + p_j + Q_2 x_{ij} \leq Q_2 + b_j \quad \forall i, \quad j \in V_C, \quad i \neq j \quad (5) \\
& \quad a_i - d_i + b_i \leq Q_2 \quad \forall i \in V_C \quad (6) \\
& \quad d_i \leq a_i \leq Q_2 \quad \forall i \in V_C \quad (7) \\
& \quad p_i \leq b_i \leq Q_2 \quad \forall i \in V_C \quad (8) \\
& \quad \sum_{j \in V_S} z_{ij} = 1 \quad \forall i \in V_C \quad (9) \\
& \quad x_{ij} \leq z_{ij} \quad \forall i \in V_C, j \in V_S \quad (10) \\
& \quad x_{ji} \leq z_{ij} \quad \forall i \in V_C, j \in V_S \quad (11) \\
& \quad x_{e,m} + z_{e,i} + \sum_{j \in V_S} z_{mj} \leq 2 \quad \forall e, \quad m \in V_C, \quad e \neq m, \quad \forall i \in V_S \quad (12) \\
& \quad \sum_{i \in V_C} \sum_{j \in V_S} x_{ij} \leq K_2 \quad (13) \\
& \quad \sum_{i \in V_S} \sum_{j \in V_C} x_{ij} \leq K_2 \quad (14) \\
& \quad \tilde{a}_i \geq 0, \quad \tilde{b}_i \geq 0 \quad \forall i \in V_C \quad (15) \\
& \quad \tilde{a}_i \geq 0, \quad \tilde{b}_i \geq 0, \quad A_i \geq 0, \quad B_i \geq 0 \quad \forall i \in V_S \quad (16)
\end{align*}
\]

The objective function (1) minimizes the total transportation cost. Constraints (2) ensure that each customer is visited exactly once. Constraints (3) guarantee flow equations for customers, i.e., every vehicle that arrives at a customer must leave that customer. Vehicle load constraints in the second phase delivery and pickup route are explained in Constraints (4)–(8). Such constraints also eliminate the subtours in the second echelon routes. Constraint (6) ensures that in any customer's area after the vehicle delivers and picks up goods, the loaded quantity does not exceed the vehicle capacity. Constraint (7) denotes that the demand of customer \( i \) must not exceed the load quantity on the vehicle before arriving, and this load
quantity value must not exceed the capacity of the vehicle. In Constraint (4) if \( x_{i,j} = 1 \), i.e., customer \( j \) is next to customer \( i \), and \( a_i - d_i = a_j \), which indicates the delivery quantity on the second phase vehicle is decreased by the demand quantity of customer \( i \) after visiting, so that \( a_j + d_i + Q_2 \leq Q_2 + a_i \). Otherwise, when \( x_{i,j} = 0 \), the inequality also holds since \( d_i \leq a_i \) and \( a_j \leq Q_2 \) according to the definition of these notations. Constraints (5) and (8) correspond to the pickup goods case and are of a similar form to Constraints (4) and (7). Constraints (9) force each customer to be assigned to one satellite. Constraints (10)–(11) ensure that there is no arc connecting customer \( i \) and satellite \( j \) if customer \( i \) is not allocated to satellite \( j \). Constraint (12) avoids two connected customers belonging to two different satellites. Constraints (13) and (14) ensure the available fleet size for the second echelon transportation. Constraints (15) and (16) equalize the total pickup and delivery demands of the customers and corresponding satellites. Constraints (17)–(22) are related to the first phase delivery route, while Constraints (23)–(28) are about the third phase pickup route; they are similar to some early constraints. Constraints (17)–(18) and (23)–(24) ensure that every satellite is visited by exactly one vehicle from the first phase and the third phase, respectively. Constraints (19) and (20) are vehicle load constraints in the first phase delivery route; Constraints (25)–(26) are corresponding constraints in the third phase pickup route. Constraints (21), (22), (27), and (28) impose fleet size constraints in the first echelon transportation. Finally, constraints (29)–(34) define the domain of decision variables.

4 Literature review

Although many variants of the VRP and the 2E-VRP have been studied in the existing literature, little research has been performed on the 2E-VRPSDP. As stated above, two main bodies, the 2E-VRP and the VRPSDP, are relevant to the 2E-VRPSDP. We survey the literature in these two bodies of research.

4.1 The 2E-VRP literature

The 2E-VRP has drawn much attention in the past decade. Both exact and heuristic algorithms have been designed to solve this problem. In terms of exact algorithms, first, Felici et al. (2007) proposed a flow-based mathematical model for the 2E-VRP and developed two families of valid inequalities. Felici et al. (2007) used an exact branch-and-cut algorithm to solve instances containing up to 32 customers and 2 satellites. Perboli et al. (2011) introduced some new optimality cut classes and improved the algorithm of Felici et al. (2007). Jepsen et al. (2013) built a directed three index formulation similar to that of Perboli et al. (2011); they derived a relaxation from it to avoid giving incorrect upper bounds when more than two satellites were included in the solution. The branch-and-cut algorithm of Jepsen et al. (2013) performed better than that of Perboli et al. (2011). A branch-and-cut-and-price algorithm was presented by Santos et al. (2014), which overcame symmetry issues and was further strengthened by valid inequalities. Baldacci et al. (2013) proposed a new mathematical formulation. Their exact method decomposed the problem into a limited set of multiple-depot vehicle routing problems with side constraints. Computational results on benchmarks proved that their method outperformed previously published exact methods in terms of size, number of problems solved to optimality, and computing time. Considering some variants of the classical 2E-VRP, Liu et al. (2018) introduced a 2E-VRP with grouping constraints, in which customers were divided into several disjoint groups, and the grouping constraints ensured that customers from the same group were served by vehicles from the same satellite. They formulated the problem as a mixed-integer program and proposed valid inequalities to strengthen the model. A branch-and-cut algorithm was implemented to solve the problem, which could solve more instances to optimality than CPLEX. Darvish et al. (2019) studied a flexible 2E-VRP in which a supplier delivered a commodity to its customers through a two-echelon supply network. Two sources of flexibility were analyzed: flexibility in network design and flexibility in due dates. The former was related to the possibility of renting any of the satellites in any period of the planning horizon, whereas the latter was related to the possibility of serving a customer between the period an order was set and a due date. A mathematical programming formulation to this problem was presented, and an exact method was proposed that was based on the interplay between two branch-and-bound algorithms. Del-laert et al. (2019) studied the 2E-VRP with time windows. Different from the classical 2E-VRP, the second echelon consisted of transferring freight from satellites to the final customers within their time windows. They proposed two path-based mathematical formulations for the problem and developed branch-and-price–based algorithms to solve the problem. The algorithms solved instances of up to five satellites and 100 customers to optimality. Breunig et al. (2019) formulated an extension of the 2E-VRP, called the electric 2E-VRP, which involved electric vehicles for second echelon deliveries, battery capacity constraints, and possible visits to charging stations, and used it as a prototypical problem for the study of multiechelon battery-powered supply chains. They designed an efficient exact algorithm based on the enumeration of candidate solutions for the first echelon and on bounding functions and route enumeration for the second echelon, along with a problem-
tailored large neighborhood search metaheuristic. Marques et al. (2020) proposed a branch-cut-and-price algorithm for the 2E-VRP. The authors introduced a new route-based formulation for the problems that do not use variables to determine product flows in satellites. They also introduced a new branching strategy that significantly decreased the size of the branch-and-bound tree and introduced a new family of satellite supply inequalities. Because exact algorithms usually cannot effectively solve the large-size 2E-VRP in a reasonable computation time, many researchers have resorted to heuristic methods. Crainic et al. (2011) developed a multistart heuristic based on separating the problem by solving customer assignments heuristically and then dealing with the remaining VRPs. In their method, a perturbation mechanism was adopted to iteratively build new solutions, and a feasibility search procedure was used to bring the solution back into the feasible region. In addition to the exact methods, Perboli et al. (2011) also presented two math heuristics based on the information obtained by solving the linear relaxation of the mathematical formulation model. Hemmelmayr et al. (2012) designed an adaptive large neighborhood search heuristic for the problem. They introduced some large-scale instances with up to 200 customers. Zeng et al. (2014) solved the problem using a greedy randomized adaptive search procedure embedded with a route-first cluster-second procedure and a variable neighborhood descent. Their approach was tested on instances not larger than 50 customers. Breunig et al. (2016) designed an effective large neighborhood-based heuristic for solving the 2E-VRP. Their algorithm improved 18 best-known solutions. Grangier et al. (2016) addressed a variant of the 2E-VRP that integrated constraints arising in city logistics such as time window constraints, synchronization constraints, and multiple trips at the second echelon. They proposed an adaptive large neighborhood search to solve this problem. Amarouche et al. (2018) proposed a new hybrid heuristic method for solving the 2E-VRP that relied on two components. The first component effectively explored the search space to discover a set of interesting routes, and the second recombined the discovered routes into high-quality solutions. Rohmer et al. (2019) presented a two-echelon inventory routing problem for perishable products. Products were delivered from a supplier to an intermediary depot, where storage may occur and from which they were delivered by smaller vehicles to the customer locations. The objective was to minimize the total transportation and holding costs. An adaptive large neighborhood search metaheuristic was designed to address the problem. Jie et al. (2019) considered the 2E-VRP with battery swapping stations, which aimed to determine the delivery strategy under battery driving range limitations for city logistics. The electric vehicles operating in the different echelons had different load capacities, battery driving ranges, power consumption rates, and battery swapping costs. The authors proposed a hybrid algorithm that combined column generation and an adaptive large neighborhood search to solve the problem. Mühlbauer and Fontaine (2021) studied a variant of the 2E-VRP that used cross-docking from vans to cargo bicycles at so-called satellites. To solve large instances, the authors introduced a new efficient parallelized large neighborhood search algorithm. The algorithm was tested using symmetric 2E-VRP benchmark instances from the literature.

4.2 The VRPSDP literature

In addition to the 2E-VRP, the VRPSDP has also been studied intensively (Koç et al. 2020). The VRPSDP was first proposed by Min (1989). Since VRPSDP is NP-hard, exact algorithms have difficulty solving large-scale real-life cases. Only a few exact methods have been developed for the problem. Dell’Amico et al. (2006) proposed an exact branch-and-price approach for the VRPSDP. Their method solved instances containing up to 40 customers. Subramanian et al. (2011) designed a branch-and-cut-based method and later Subramanian et al. (2013) developed a branch-and-cut-and-price method for solving the VRPSDP. Hernández-Pe´rez et al. (2021) addressed a generalization of the one-commodity pickup and delivery traveling salesman problem where each customer supplied or demanded a given quantity of a certain product. The authors presented three mathematical models for the problem and designed an exact branch-and-cut algorithm to solve it.

Compared with the exact algorithms, heuristic approaches dominated the solution methodologies of the VRPSDP. Nagy and Salhi (2005) proposed a heuristic that allowed infeasibilities to occur and guided the search toward strong feasibility through search routines. Crispim and Branda´o (2005) presented a hybrid algorithm for the VRPSDP that was comprised of the two metaheuristics of tabu search and variable neighborhood descent. Alfredo Tang Montané and Galvao (2006) proposed a tabu search approach that utilized relocation, interchange and crossover movements to obtain interroute adjacent solutions and used a 2-opt procedure to obtain alternative intraroute solutions. Bianchessi and Righini (2007) presented and compared constructive algorithms, local search algorithms and tabu search algorithms on the VRPSDP. Gajpal and Abad (2009) designed an ant colony system (ACS) for solving the VRPSDP, which used a construction rule as well as two multi route local search schemes. Zachariadis et al. (2010) introduced an adaptive memory algorithmic framework to solve the VRPSDP, which combined promising solution features to generate high-quality solutions. Avci and Topaloglu (2015) proposed an adaptive local search
solution approach for the VRPSDP, which hybridized a simulated annealing inspired algorithm with variable neighborhood descent. A perturbation-based variable neighborhood search heuristic for solving the VRPSDP was designed by Polat et al. (2015). Zachariadis et al. (2016) introduced a special VRPSDP with two-dimensional loading constraints, which covered cases where customers posed delivery and pick up requests for transporting nonstackable rectangular items. The authors proposed an optimization framework that employed memorization techniques to accelerate the solution methodology. Computational results were reported on the VRPSPD, the VRP with two-dimensional constraints, and newly constructed benchmark problems. Qiu et al. (2018) studied a variant of VRPSDP, in which customers’ demands were discrete in terms of batches (or orders) and each customer could be visited by a variety of vehicles or several times by one vehicle. A tabu search algorithm with specifically designed batch combinations and item creation operations was proposed to solve the problem. Majidi et al. (2018) dealt with the pollution routing version of VRPSDP, where the goal was to minimize fuel consumption and emissions by scheduling and routing customers. A nonlinear mix integer programming model was presented for this problem, and an adaptive large neighborhood search heuristic was proposed for the solution method including new removal and insertion operators. Zhang et al. (2019) addressed a multicommodity many-to-many vehicle routing problem with simultaneous pickup and delivery for a fast-fashion retailer in Singapore. To solve large-scale instances, an adaptive memory programming-based algorithm combined with techniques such as the regret insertion method for initializing the solution pool, the segment-based evaluation scheme, and the advanced pool management method was proposed in this work.

Although as stated above, many good exact algorithms and heuristics were proposed for the 2E-VRP and VRPSDP, to our knowledge, only Belgin et al. (2018) studied a special two-echelon vehicle routing problem with simultaneous pickup and delivery. However, the basic operation scheme of the problem in Belgin et al. (2018) differs from the problem studied in this paper. Belgin et al. (2018) assumed that goods of the first echelon are delivered from the central depot to each satellite and collected from satellites back to the depot simultaneously by the same vehicles. As stated in Sect. 2, our 2E-VRPSDP separates the routes in the first echelon into one delivery route and one pickup route. To our knowledge, this is the first work on 2E-VRPSDP.

5 Variable neighborhood search algorithms

Since the VRPSDP and the 2E-VRP are two difficult NP-hard problems, most research on these two problems resorts to heuristics methods, such as tabu search (TS), variable neighborhood search (VNS), adaptive large neighborhood search heuristic (ALNS), and genetic algorithm (GA). Because the 2E-VRP can be seen as a special case of the 2E-VRPSDP studied in this paper, the 2E-VRPSDP is also an NP-hard problem and even more complex than VRPSDP and 2E-VRP. Thus, exact algorithm or commercial solver behavior is highly unpredictable. For example, we attempt to solve the above 2E-VRPSDP mathematical formulation using the Cplex and Gurobi. We find that the solvers cannot find the high-quality solution in a reasonable computational time (e.g., 2–3 h) even for moderate-size test instances (refer to the numerical results in Sect. 6.4). Thus, in this paper, a variable neighborhood search algorithm (VNS) is designed for the 2E-VRPSDP. The principle of VNS is a systematic change in neighborhoods within a local search procedure. As a local search-based algorithm, the main advantage of VNS is not the computation time. For example, it is slower than some simple greedy or hill-climbing algorithms. However, it has a great ability to find high-quality solutions, especially for some very complex NP-hard optimization problems. This has been widely proven by solving the VRP as well as its variants (Kytöjoki et al. 2007; Fleszar et al. 2009; Hemmelmayr et al. 2009). For a more thorough description of VNS, refer to Mladenović and Hansen (1997) and Hansen and Mladenović (2001). In this paper, we build a new VNS approach, the outline of which is shown in Algorithm 1. First, the algorithm identifies a set of neighborhood structures $N_k$ ($k = 1, 2, \ldots, k_{\text{max}}$) and an initial solution $S$. Parameter $k$ equals 1. Then, a shaking step is performed by randomly selecting a solution $S'$ of solution $S$ based on the $k$th neighborhood $N_k$. Next, the algorithm applies a local search improvement procedure to the current solution $S'$. This procedure is repeated once a new incumbent solution $S''$ is found. If this local optimum $S''$ is better than $S$, the algorithm sets $S \leftarrow S''$ and searches with $k = 1$. Otherwise, VNS switches to the next neighborhood with $k = k + 1$ to perform another shaking step, followed by local search improvement. Note that our approach extends the search to infeasible solutions, i.e., both feasible and infeasible solutions are allowed by the VNS. Each solution to the problem corresponds to a set of routes. Infeasibility occurs for a VNS solution if the total demand of a vehicle exceeds the vehicle’s specified capacity. The algorithm uses a weighted linear penalty function for violations of this constraint. The details of the implementation of an infeasible solution are presented in the following sections.
5.1 General functioning of the algorithm

Since the 2E-VRPSDP is a rather complicated problem, the VNS approach extends the search space to the infeasible region. It is frequently conjectured that infeasible solutions enable a better transition during the search between structurally different feasible solutions (Cordeau et al. 2001). In particular, purposeful management of penalties may enable us to focus the search toward borders of feasibility, a place where high-quality solutions are more likely to be located (Glover and Hao, 2011). Thus, in our VNS, the vehicle capacity constraint is temporarily violated during the search. A weighted and linear penalty function with a penalty weight $a$ is introduced into the cost function for the violation of vehicle capacity constraints. For a solution $S$, a general cost function $c(S)$ is defined as $c(S) = d(S) + \alpha \times e(S)$ for solution evaluation, where $d(S)$ denotes the total travel distances of vehicle routes (original objective function), $e(S)$ denotes the total violations of vehicle load of all the routes in three phases, and $\alpha$ is the penalization parameter. Term $e(S)$ is defined as follows:

$$
e(S) = \sum_{i \in V_r} (load_{i1} - Q_1)^+ + \sum_{i \in V_r} (load_{i2} - Q_2)^+ + \sum_{i \in V_r} (load_{i3} - Q_1)^+
$$

(35)

where $x^+ = \max(0, x)$, $load_{i1}$ and $load_{i3}$ are the vehicle loads after a first phase delivery route and a third phase pickup route serves satellite $i$, respectively; $load_{i2}$ is the load of a vehicle after a second phase delivery and pickup route serves customer $i$. Clearly, if solution $S$ is feasible, i.e., the vehicle capacity is strictly satisfied at each node, $e(S) = 0$ and $c(S) = d(S)$; otherwise, $e(S) > 0$ and $c(S) > d(S)$. In our VNS penalty parameter, $\alpha$ is adjusted dynamically to facilitate the exploration of the search space. If the solution $S'$ obtained after shaking and local search is feasible (referred to Algorithm 1), the value of $\alpha$ is divided by a factor of $1 - u(S)$; otherwise, $\alpha$ is multiplied by this factor. This penalization parameter $\alpha$ is initialized with $\alpha_0$ and is limited between an upper bound $\alpha_{\text{max}}$ and a lower bound $\alpha_{\text{min}}$ which limit the maximum and minimum values of this parameter during the search process.

5.2 Construction of an initial solution

Based on the savings algorithm (Clark and Wright 1964), we propose a method to obtain an initial solution to our problem. The initial solution method has three stages.

- **Stage 1: customer assignment** First, a random sequence of all the customers is generated. Following this sequence, each customer is assigned to its nearest

---

**Algorithm 1. Variable neighborhood search algorithm**

1. Identify neighborhood structures set $N_k (k=1,...,k_{\text{max}})$, penalty parameter values
2. Generate an initial solution $S$, $k=1$
3. while stop criterion is not reached, do
4.     while $k\neq k_{\text{max}}$, do
5.         generate one random solution $S'$ from $S$, based on $N_k$/shaking
6.         if $S'$ is feasible then
7.             apply local search (restricted in feasible region) to $S'$ and obtain $S''$
8.         else // $S'$ is infeasible
9.             apply local search, extending to the infeasible region to $S'$ and obtain $S''$
10.        end if
11.    adjust the penalty parameters
12.    if $S''$ is better than $S$, or the acceptance criterion is satisfied, then
13.        $S' = S''$, $k=1$
14.    else
15.        $k = k+1$
16.    end if
17. end while
18. end while
satellite. During this assignment procedure, the algorithm ensures that for each satellite, the total delivery or pickup demands from all associated customers cannot be greater than the first echelon vehicle capacity \( Q_1 \). If an assignment of a customer to a satellite violates this constraint, this customer is assigned to its second nearest satellite, and so on until all the customers are assigned to satellites. Because the vehicle capacity is limited, the above procedure may fail to assign each customer to a satellite. In this case, the initial customer sequence is regenerated until all customers have been assigned. Note that in the first step, the random sequence introduces randomness to the algorithm, similar to the works of Cordeau et al. (1997, 2001).

- **Stage 2: second echelon routing** The algorithm applies the savings algorithm (Clark and Wright 1964) to each satellite to service the customers who have been assigned to this satellite. For satellite \( s \), a separate back and forth route for each customer is initially created. For each customer pair \( i \) and \( j \) of satellite \( s \), the savings is defined as \( s_{ij} = c_{ii} + c_{jj} - 2c_{ij} \). Starting from the largest and nonnegative saving \( s_{ij} \), two routes (\( s, \ldots, i, s \)) and (\( s, j, \ldots, s \)) are merged into a single second phase delivery and pickup route (\( s, \ldots, i, j, \ldots, s \)). The above route combinations are executed until the number of second phase delivery and pickup routes is no more than \( K_2 \) (the total number of second echelon vehicles). Note that the above procedures may construct an infeasible solution that violates the vehicle capacity constraint. For example, part A of Fig. 2 shows four routes that start and end at satellites \( S_1 \) and \( S_2 \). Based on such routes, the savings values of the merging route \((S_1, C_1, C_2, S_1)\) and \((S_1, C_3, S_1)\) and combining \((S_2, C_4, S_2)\) and \((S_2, C_5, S_2)\) are computed. If the former savings value is larger, then the two routes of satellite \( S_1 \) are merged, and one larger route \((S_1, C_1, C_2, C_3, S_1)\) is obtained, as shown in part B of Fig. 2.

- **Stage 3: first echelon routing** After obtaining the second phase delivery and pickup routes, similar to stage 2, the algorithms apply the savings algorithm to obtain the first phase delivery routes and the third phase pickup routes.

### 5.3 Shaking

Shaking is the foundation of the VNS, which diversifies the solution and avoids it from being trapped in a local optimum. Each neighborhood should determine a proper balance between perturbing the incumbent solution and retaining the good parts of the incumbent solution. The algorithm employs ten interroute neighborhood structures (namely, \( N_1-N_{10} \)) in VNS shaking. The set of neighborhoods is summarized in Table 1.

Neighborhoods \( N_1 \) to \( N_4 \) relocate or exchange satellite(s) among two first phase delivery routes. Neighborhoods \( N_5 \) to \( N_8 \) similarly deal with the third phase pickup routes. Note that if a satellite is moved, the customers and second phase routes associated with this satellite are moved with this satellite. Figure 3 shows an example of a neighborhood \( N_4 \) operator: Satellite \( S_1 \) is removed from a first phase delivery route (Depot–\( S_2 \)–\( S_1 \)–Depot) and inserted into another route (Depot–\( S_3 \)–Depot).

The last two neighborhoods, \( N_9 \) and \( N_{10} \), involve the change in the second phase delivery and pickup route and the first echelon routes (pickup route and delivery route), simultaneously. They are more complex than the above neighborhoods. To clarify \( N_9 \) and \( N_{10} \), two satellite statuses close and open are used. In this paper, when one or more customers are assigned to a satellite, the status of this satellite is open. For a satellite, when all its customers are assigned to other satellite(s), this satellite becomes closed. Neighborhoods \( N_9 \) first randomly close a satellite. The customers assigned to this satellite are removed from this satellite one by one. The algorithm inserts each removed customer into a random position of another second phase delivery and pickup route or constructs a new second phase delivery and pickup route that only contains this customer. In terms of constructing a new second phase route, it may be built on an open satellite or a currently closed satellite. Once this new route is built on a closed satellite, this satellite changes to open. Thus, this satellite is simultaneously inserted into a first phase delivery route and third phase pickup route. For example, in Fig. 4, after shaking, satellite \( S_2 \) is closed. Customers \( C_4 \) and \( C_5 \) are assigned to the current open satellite \( S_1 \) and the closed satellite \( S_4 \), respectively. Thus, satellite \( S_4 \) now becomes open and is inserted into a first phase delivery route and a third phase pickup route (Depot–\( S_3 \)–\( S_4 \)–Depot).

### 5.4 Local search

The solution obtained from the shaking procedure needs to be further improved to obtain a local optimum. The local search procedures are successfully hybrid in the metaheuristics. For example, in Prins (2004) and Vidal et al. (2012), local search procedures were applied to improve the quality of the offspring solutions in the genetic algorithms. Prins (2004) noted that the local search improvement procedure was a key idea of algorithm design. Thus, in our VNS, four kinds of local search methods are used, including interroute and intraroute methods, i.e., interroute 1–0 relocation move, interroute 1–1 exchange move, intraroute 1–0 relocation move and intraroute 1–1 exchange move. Such operators are widely used by VRP.
studies (Bräysy and Gendreau, 2005; Vidal et al. 2012). The interroute 1–0 move refers to relocating a customer from its current position to a position in another route or constructing a new route, while the intraroute 1–0 move...
denotes relocating a customer from its current position to another position on the same route. Similarly, the interroute 1–1 move involves exchanging two customers’ positions of different routes, and the intraroute 1–1 move exchanges two customers’ positions on the same route. The four procedures are used for all three vehicle route phases. Note that for the second phase delivery and pickup route, the interroute 1–0 move may change the status of satellites. For example, now satellite \( j \) has only one customer, and satellite \( k \) is closed. When this customer is moved from satellite \( j \) to satellite \( k \), satellite \( j \) becomes closed, and satellite \( k \) is open. During the local search procedure, the algorithm adopts the “first-accept” strategy, i.e., during a local search procedure once a better solution is found, it is adopted as the new seed for repeating this local search. The local search procedure continues until no improvements can be made.

In terms of the classic local search methods, if a local search starts from a feasible solution seed, the algorithm limits the search in a feasible solution area. Otherwise, if it starts from an infeasible seed, the whole feasible and infeasible solution areas are allowed to be searched. In this paper, the algorithm adopts a new local search strategy. If the solution obtained from shaking (the origin of the whole local search) is feasible, then the local search is restricted to the feasible region, which means that the vehicle capacity cannot be exceeded at any move of the local search. Otherwise, if the shaking result is infeasible, the local search is extended to the infeasible area where the generalized cost, including penalties, is used to evaluate each move.

5.5 Acceptance criterion

After the shaking and the local search procedures have been performed, the solution obtained at the current iteration is compared to the incumbent solution to decide whether it is accepted. To prevent the VNS from quickly becoming stuck in a local optimum, inspired by simulated annealing (SA) (Kindervater and Savelsbergh 1997), the scheme of accepting a new but worse solution is adopted. If after shaking and local search procedures, a new solution \( S'' \) that is better than the incumbent solution \( S \) is identified, \( S'' \) is accepted directly to replace \( S \). Otherwise, this new solution is accepted with a probability \( \exp\left(-\frac{c(S'') - c(S)}{T}\right) \), where \( c(S'') \) and \( c(S) \) are the generalized costs of two solutions. The annealing temperature \( T \) decreases linearly from an initial value \( T_0 \), i.e., after each VNS iteration, \( T \) is reduced by \( T \times (1/\text{ite}) \), where \( \text{ite} \) is the VNS maximum iteration time.

6 Computational results

In this section, computational experiments are conducted to assess the performance of the proposed approach. All algorithms in this paper were coded in C++. The computational experiments were conducted on an Intel E5-2670 processors clocked at 2.6 GHz with 2 GB of memory running Linux. All the algorithms are run 10 times for each test instance. The best and the average solution costs and the average algorithmic running time are obtained from 10 runs for each test instance.

---

**Fig. 4** Neighborhood \( N_0 \) operator
6.1 Test instances from the 2E-VRPSDP benchmarks

Since there is no benchmark instance for our problem, the 2E-VRP benchmark instances from the literature are modified. Thanks to Breunig et al. (2016), the existing 2E-VRP instances were summarized and are available at https://www.univie.ac.at/prolog/research/TwoEVRP. The 2E-VRPSDP test instances are derived from existing 2E-VRP benchmark instances as follows. We select 46 basic 2E-VRP instances in which twenty 2E-VRP instances contain 50 customers, twenty have 100 customers, and the remaining six instances have 200 customers. For each basic 2E-VRP instance, we derive a new instance as follows. We select first 50% customers and set 1/3 of the original demand as the delivery demands and 2/3 of the original demand as the number of pickups. For the left 50% customers, we set 1/3 and 2/3 of the original demand as the pickup and delivery demands, respectively. Other parameters, including locations of all points, vehicle number limits, and vehicle capacity, are the same as those of the original 2E-VRP instance. Since there are only six 2E-VRP instances of 200 customers available in the literature, we change the vehicle capacity and the vehicle number of an echelon and derive another fourteen 2E-VRPSDP instances with 200 customers (with signs of c-g in Table 8). In total, we have sixty 2E-VRPSDP instances for computational experiments. The original 2E-VRP benchmark set, which each of our new 2E-VRPSDP instances comes from, is marked at the beginning of the instance name, and “SDP” is added at the end. For example, the new instance generated based on “E-n51-k5-s2-17” (2E-VRP benchmark, Set 2b) is named “Set2b_E-n51-k5-s2-17_SDP.”

6.2 Parameter tuning

Based on the above test data, we select 18 2E-VRPSDP test instances to tune the parameters in our algorithm, consisting of six small instances with 50 customers, six medium instances with 100 customers, and six large instances with 200 customers. Each instance is solved by the algorithm with a one-parameter setting ten times. We obtain ten computational results for each test instance, including the best solution cost and mean solution cost among the 10 running results. Then, we calculate the average of all best solution costs and the average of the mean solution cost across all 18 test instances. We use these two statistical results to assess the solution quality of such parameter settings. Table 2 summarizes the final parameter settings of our VNS algorithm used in the experiments. In the following subsections, some parameters are tuned, and a sensitivity analysis is executed to elucidate the effects of the components of the proposed algorithm.

6.2.1 Parameter \( T_0 \)

Parameter \( T_0 \) is the initial temperature of the simulated annealing-based acceptance criterion (Sect. 4.5). Clearly, this parameter has an important effect on the VNS, which controls the probability of accepting a relatively worse solution in the algorithm iterations. We test parameter \( T_0 \) on the set of \{50, 75, 100, 125, 150, 200\} while using the values in Table 2 for other algorithmic parameters. The results are summarized in Table 3. In this table, row “Avg-Best” shows the average value of all best solution costs to a total of 18 test instances, and row “Avg-Mean” gives the average of all mean solution costs. We find that the solution quality improves with the increase in parameters from a minimum value of 50 and peaks at a value of 100, and then the accuracy of the algorithm decreases with the value of this parameter.

6.2.2 Parameter \( \varphi \)

In the VNS, for each solution \( S \), an extended cost \( c(S) \) is defined as \( c(S) = d(S) + \varphi \times e(S) \) for solution evaluation (see Sect. 5.1). At each VNS iteration, when a solution \( S' \) is obtained after shaking and local search, the algorithm judges its feasibility. If this solution is feasible, the value of \( \alpha \) is divided by \( 1 + \varphi \); otherwise, \( \alpha \) is multiplied by \( 1 + \varphi \). Clearly, this parameter \( \varphi \) controls the change frequency of parameter \( \alpha \) and the frequency at which the algorithm jumps between feasible and infeasible solution areas. We test \( \varphi \) in the set of \{1.01, 1.05, 1.1, 1.2, 1.3, 1.5\}. The results are shown in Table 4. The best algorithm performance is yielded at \( \varphi \) equal to 1.05.

6.3 Sensitivity analysis of algorithmic components

Furthermore, a set of experiments is designed to elucidate the effects of VNS algorithm components. We attempt to remove one or a set of components from the algorithm, and the remaining algorithm’s performance is explored. Based on these experiments, the roles of the algorithm components are tested and verified.

The first is the shaking component (see Sect. 4.3). Ten interroute neighborhood structures (namely, \( N_1-N_{10} \)) are adopted in VNS as shaking. In this part of the experiments, such neighborhood structures are classified into four sets. The first set contains \( N_1, N_3, N_5, \) and \( N_7 \), which relocate a satellite to another position; the second set consists of \( N_2, N_4, N_6, \) and \( N_8 \), which exchange the positions of two satellites. The third is \( N_9 \), which closes a satellite, and the
final fourth is $N_{10}$, which removes a number of sequential customers and inserts them into another route. We sequentially delete one set of neighborhood structures from the whole algorithm and yield four versions of algorithms, i.e., “no-relocation,” “no-exchange,” “no-close,” and “no-remove.” We also aim to test the component of the local search (Sect. 4.4). The solution obtained from the shaking procedure is further improved by a set of local search methods. The first set of local search methods is \texttt{1-0 relocation}, and the second is \texttt{1-1 exchange} (details are referenced in Sect. 4.4). Again, each time one set of local search methods is shielded, the remaining algorithms “no-\texttt{1-0 LS}” and “no-\texttt{1-1 LS}” are tested. Each version of the modified algorithm is executed 10 times over each instance. The results are shown in Table 5. The results show that all these algorithmic components enhance the performance of the algorithm because “Whole VNS” yields the best results. Therefore, it is concluded that the algorithm performance will be compromised if one component is deleted from the algorithm.

Furthermore, in terms of each of the above operators, we also want to know how many times it is used, or whether it truly improves the solution during the VNS iteration process. For example, considering four types of Shaking operators (relocation, exchange, close, and remove), we record the mean execution times of each operator during VNS iterations. They contribute 12.7%, 10.7%, 37.2%, and 39.4% of shaking executions, respectively. This statistical result shows that each shaking operator is executed and has a considerable positive effect on the algorithm performance. We also record the success rate of two local search methods \texttt{1-0 relocation} and \texttt{1-1 exchange} (i.e., the number of times that one LS operator improves the solution during VNS whole iterations, over the total improvement times of all LS operations). It is found that 1–0 relocation contributes approximately 53% local search improvement, whereas 1–1 exchange contributes 47% improvement. This result also indicates that two local search methods are necessary in the algorithm. Furthermore, we find that the most running time of VNS is used by local search procedures, and they account for approximately 90% of the running time of the whole algorithm. The remaining time is mainly used by shaking operators. Furthermore, the computational time of each shaking operator accounts for a similar percentage of the total shaking time as their executions (i.e., approximately 10%, 10%, 35%, and 45%). Similarly, each local search operator’s running times also account for a similar percentage of the total consumption time as their contributions stated above.

### 6.4 Computational results for 2E-VRPSDP instances

We cannot find other algorithms for the 2E-VRPSDP, so in this paper, we first develop a simplified version of our approach, named VNS1, to test the design and performance of the proposed VNS. In the VNS1 approach, the basic framework of the algorithm is the same as that of the above approach, including the construction of an initial solution, shaking, local search and the acceptance criterion. However, the search space of VNS1 is always restricted to the feasible region, which means that in the shaking and local search procedure, the vehicle capacity constraints are always obeyed. Thus, the solution obtained at any iteration

### Table 2 Parameter setting in the experiment

| Symbol | Explanation | Value |
|--------|-------------|-------|
| $T_0$  | The initial temperature in the acceptance criterion | 100   |
| $\alpha_0$, $\alpha_{\text{min}}$, $\alpha_{\text{max}}$ | Dynamic penalization parameter in $c(s)$ | $\alpha_0$: 1, $\alpha_{\text{max}}$: 10,000, $\alpha_{\text{min}}$: 0.001 |
| $\varphi$ | Parameter used to adjust penalization parameter $\alpha$ | 1.05  |
| $\text{ite}$ | The maximum number of VNS iterations | 10,000 |

### Table 3 Sensitivity of the algorithm performances to $T_0$

| $T_0$ | 50     | 75     | 100    | 125    | 150    | 200    |
|-------|--------|--------|--------|--------|--------|--------|
| Avg-Best | 1089.98 | 1091.59 | 1088.00 | 1088.52 | 1090.65 | 1090.56 |
| Avg-Mean | 1118.27 | 1117.41 | 1116.44 | 1117.68 | 1123.37 | 1119.24 |

### Table 4 Sensitivity of the algorithm performances to $\varphi$

| $\varphi$ | 1.01 | 1.05 | 1.1 | 1.2 | 1.3 | 1.5 |
|-----------|------|------|-----|-----|-----|-----|
| Avg-Best  | 1088.94 | 1088.00 | 1110.57 | 1099.20 | 1101.57 | 1101.52 |
| Avg-Mean  | 1112.68 | 1116.44 | 1136.89 | 1125.63 | 1130.40 | 1137.72 |
of VNS1 is always feasible, and its penalty cost equals zero. Such a version of VNS is more classic and standard. The VNS approach proposed above (extending the algorithm to an infeasible solution area) is denoted as VNS2. To give a fair comparison between two approaches, VNS1 and VNS2, for each test instance, we change the stop criterion of algorithm VNS1, i.e., we do not use maximum iterations to terminate VNS1; it is stopped after a special total running time that equals the computational time of VNS2. That is, the two approaches have the same running times for solving each test instance.

Additionally, we also try to solve the problem using the commercial solver Gurobi and a decomposition-based approach. In terms of the Gurobi solver, we implement the mathematical model in Sect. 3 and solve each test instance with a maximum of 3 h running time much longer than that of the VNS. The decomposition-based approach is built on the basis that the 2E-VRPSDP is decomposed into two levels, in each of which a special vehicle routing problem is solved. That is to say, the classical VRPs are solved at the first level, and a pickup and delivery problem is optimized at the second level (refer to Fig. 1). Finally, the solutions of two levels are combined to obtain a whole solution to our 2E-VRPSDP problem. For the routing problem at each level, a simplified VNS algorithm of this paper is used to optimize the vehicle routes. The details of the simplified VNS algorithm of this paper is used to optimize the vehicle routes. The details of the simplified VNS algorithm are presented in Appendix.

We tested a total of sixty 2E-VRPSDP instances on Gurobi, decomposition-based approach, VNS1 and VNS2. The numerical results are aggregated in Table 6, according to the scales of the test instances. The detailed results are presented in Table A1–A3 in Appendix. In Table 6, columns “Avg” and “Best” give the average solution cost and best solution cost over ten runs of the corresponding algorithm, respectively.

As shown in Table 6 and Tables A1–A3 in Appendix, for all instances, VNS2 yields better solutions from the perspective of solution cost (quality). VNS2 can find a better solution for each test instance than Gurobi, decomposition-based approach, and VNS1. As shown in Table 6, compared with Gurobi, VNS2 can improve mean solution costs quality 17.72%, 27.77%, and 40.43% for 50-customer, 100-customer, and 200-customer instances, respectively. The superiority of our VNS2 over the solver becomes clearer for larger-scale tests. Compared with the decomposition-based approach, the superiority of our VNS is still clear: For three scales of test instance, the mean solution costs of our VNS2 is smaller than that of the decomposition-based approach 25.38%, 15.00%, and 12.42%.

Finally, we compare two VNS algorithms. The deviations between the average solution costs of VNS1 and VNS2 are 17.76%, 8.17%, and 5.05% for 50-customer, 100-customer, and 200-customer instances, respectively. Concerning the best solution costs, VNS2 deviates from VNS1 by 16.09%, 6.30%, and 6.45%, respectively.

6.5 Computational results for 2E-VRP benchmarks

To further assess the proposed VNS approach, we tested it on existing 2E-VRP benchmark instances from the literature. The 2E-VRP can be regarded as a special 2E-VRPSDP when the pickup demand of each customer of the 2E-VRPSDP is zero. In such cases, the second phase delivery and pickup route is reduced to the second phase delivery route; the third phase pickup routes do not exist. Note that the classic 2E-VRP assumes that each satellite can be visited by more than one first phase delivery route; however, in the 2E-VRPSDP, a satellite is limited to service from no more than one vehicle. Thus, we select twenty

| Table 5 | Sensitivity analysis of algorithmic components |
|---------|----------------------------------------------|
| No-relocation | No-exchange | No-close | No-remove | No-1–0 LS | No-1–1 LS | Whole VNS |
| Avg-Best | 1092.99 | 1089.23 | 1094.77 | 1115.30 | 1158.40 | 1100.32 | **1088.00** |
| Avg-Mean | 1122.07 | 1118.60 | 1137.93 | 1148.26 | 1216.62 | 1144.40 | **1116.44** |

| Table 6 | Computational results of 2E-VRPSDP instances (aggregated) |
|---------|----------------------------------------------|
| Instance scale | Gurobi | Decomposition approach | VNS1 | VNS2 |
| | | | Avg | Best | Avg | Best |
| 50 customers | 1150.36 | 1268.39 | 1200.12 | 1155.24 | 959.09 | 946.52 |
| 100 customers | 1380.19 | 1172.84 | 1101.11 | 1067.35 | 1009.11 | 996.94 |
| 200 customers | 2030.39 | 1380.96 | 1309.58 | 1291.99 | 1244.42 | 1209.49 |
2E-VRP benchmark instances in which the best-known solutions (BKS) of the 2E-VRP from the literature (Breunig et al. 2016; Mühlbauer and Fontaine 2021) do not assign more than one vehicle to visit a satellite. To fully compare with other 2E-VRP methods, the results of the following existing state-of-the-art algorithms are presented, i.e., the large neighborhood search-based heuristic (LNS-2E) of Breunig et al. (2016), the adaptive large neighborhood search heuristic (ALNS) of Hemmelmayr et al. (2012), the hybrid heuristic (GRASP + VND) from Zeng et al. (2014), and the parallelized large neighborhood search heuristic (PLNS) from Mühlbauer and Fontaine (2021) when applicable. The LNS-2E of Breunig et al. (2016) is insensitive to its parameters, which are presented in Table 2 of Breunig et al. (2016). The numerical experiments of Breunig et al. (2016) are executed on an Intel E5-2670v2 CPU at 2.5 GHz with 3 GB RAM. The parameters of ALNS in Hemmelmayr et al. (2012) are detailed in their Sect. 6.2, and the ALNS algorithm is carried out on a 2.2 GHz AMD Opteron 275 Processor. The algorithm of Zeng et al. (2014) is carried out using a single core of an Intel Pentium Dual-Core E5500 processor 2.8 GHz and 2 GB of memory. The PLNS in Mühlbauer and Fontaine (2021) is performed on a computer with 8G RAM and an Intel i5–6200 processor with 2.4 GHz, two cores and four threads. The PLNS is a parallelized algorithm, which is executed on the four threads in parallel. In addition to the results of the above algorithms, we also compare our results with the current BKS of each test instance from the literature.

Table 7 aggregates and compares detailed computational results on the existing 2E-VRP benchmark instances, according to the scale of test instance (customer number). The detailed results for each test instance are shown in Table A4 in Appendix. In Table 7, Column “BKS” refers to the best-known solution of the instance from the existing literature. For the other approaches (LNS-2E, ALNS, GRASP + VND, PLNS), subcolumn “Best” gives the best objective value found within their five runs; column “Avg” shows the average objective value of these five runs. Note that Zeng et al. (2014) only provided the average running time (in seconds) to find the best solutions (not the whole algorithm running time) in their paper. Therefore, for their results, Table 7 gives such times, whereas for other approaches, the whole algorithm running times (average run one time, in seconds) for solving each instance are shown.

First, we compare the solution quality of different algorithms. As shown in Table 7 and Table A4, the first four approaches can find BKS for the first ten instances that contain 50 customers. The VNS algorithm and the LNS-2E of Breunig et al. (2016) have better performances since they can obtain BKS for every algorithm running, i.e., the average solution (column Avg) equals the best solution
Considering larger-scale test instances (100 customers), we observe that VNS has the highest quality mean of the best solution (1128.42). Considering the largest instances (200 customers), the mean of the VNS best solution (1372.73) is only worse than that of the PLNS (1355.27) and is better than the other algorithms. Regarding the number of BKS found by each approach, as shown in Table A4, VNS finds five existing BKS; LNS-2E and PLNS both obtain three existing BKS. Compared with LNS-2E, VNS can find six better solutions, and LNS-2E can only find one better solution than VNS. Compared with PLNS, VNS finds three better solutions (100–5–1, 100–5–1b, and B-n101–5), and for 3 test instances, PLNS obtains better solutions (200–10–1, 200–10–1b, and 200–10–2). These results indicate that for such large test instances, VNS is better than LNS-2E, and the two algorithms VNS and PLNS, are similar in solution quality.

Additionally, Table 8 aggregates the best objective value found by each approach during all experiments, including those used for parameter calibration. We can observe the superiority of the VNS and PLNS over other approaches. The detailed results of Table 8 are shown in Table A5 in Appendix. As shown in Table A5, for 15 out of a total of 20 test instances, the VNS can find the BKS from the literature. The BKS found by each other algorithm is much less than VNS. Such numerical results indicate that the VNS can obtain high-quality solutions for the two-echelon vehicle routing problem with simultaneous delivery and pickup.

In addition to the solution quality, in terms of runtime for such test instances, as shown in Table 7, the mean running time of VNS is shorter than that of LNS-2E and ALNS. For example, the mean computational times of VNS and LNS-2E over all test instances are 284 s and 480 s, respectively. Note that the total runtime of GRASP + VND is not given in the literature; its running time shown in Table 7 is only the running time to find the best solutions (not the whole algorithm running time). We also found that the running time of PLNS is shorter than that of our VNS, especially for large-scale test instances with 100 + customers. However, please note that the PLNS is a parallelized metaheuristic, and our VNS is not designed as a parallel algorithm. The parallelized technique can greatly reduce the running time of PLNS. In summary, compared with existing state-of-the-art approaches, the VNS in this paper does not have an obvious superiority in computation time. However, its running time is acceptable and reasonable for all these test instances.

7 Conclusions and future work

In this paper, a new two-echelon transportation problem was investigated, the two-echelon vehicle routing problem with simultaneous deliveries and pickups (2E-VRP-PSD). The 2E-VRP-PSD considers the distribution and collection of freight on two echelons. It is more complex than the classic two-echelon vehicle routing problem (2E-VRP) and the vehicle routing problem with simultaneous deliveries and pickups (VRP-PSD). To address this challenging problem, a VNS algorithm was developed, which extends the search space into the infeasible region and strikes a balance between infeasible solutions and feasible solutions by adding a linear penalty cost function. To prevent the algorithm from becoming stuck in a local optimum, an acceptance criterion inspired by simulated annealing was embedded into the proposed method. To examine the algorithmic performances, new 2E-VRP-PSD test instances were generated based on the existing 2E-VRP benchmarks. Our approach was compared with its simplified version, which restricts the search in the feasible region. The results show the priority of the proposed method in obtaining higher-quality solutions. The proposed approach was also compared with existing state-of-the-art algorithms on the 2E-VRP benchmarks from the literature. The numerical results prove the efficiency and effectiveness of the proposed approach.

The work presented in this paper can be extended in two directions. The first is that uncertain elements can be added to the current framework, such as customers with random delivery and pickup demands. Stochastic demands may make the problem more similar to real-life applications. Additionally, this paper focuses on a heuristic algorithm, and the optimal solution cannot be guaranteed. It is interesting to consider exact decomposition algorithms such as Bender decomposition or branch-and-price algorithms to solve this problem optimally in future work.

| Instance scale | VNS | LNS-2E | ALNS | GRASP + VND | PLNS | BKS |
|----------------|-----|--------|------|-------------|------|-----|
| 50 customers   | 1181.78 | 1181.78 | 1181.78 | 1181.78 | 1184.25 | 1181.78 |
| 100 customers  | 1128.42 | 1129.87 | –     | –          | 1130.34 | 1128.34 |
| 200 customers  | 1370.05 | 1370.05 | 1383.54 | –          | 1355.27 | 1354.49 |
Supplementary Information The online version contains supplementary material available at https://doi.org/10.1007/s00500-021-06692-3.

Acknowledgements This work is supported by Research Grants from the National Natural Science Foundation of China (71672112 and 71972133).

Declarations

Conflict of interest The authors declare that there are no conflicts of interest regarding the publication of this paper.

Ethical approval This article does not contain any studies with human participants or animals performed by any of the authors.

References

Alfredo Tang Montané F, Galvão RD (2006) A tabu search algorithm for the vehicle routing problem with simultaneous pick-up and delivery service. Comput Oper Res 33:595–619

Amarouche Y, Guibadj RN, Moukrim A (2018) A neighborhood search and set cover hybrid heuristic for the two-echelon vehicle routing problem. In: 18th Workshop on algorithmic approaches for transportation modelling, optimization, and systems (ATMOS 2018). Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik

Aveti M, Topaloglu S (2015) An adaptive local search algorithm for vehicle routing problem with simultaneous and mixed pickups and deliveries. Comput Ind Eng 83:15–29

Baldacci R, Mingozzi A, Roberti R, Calvo RW (2013) An exact algorithm for the two-echelon capacitated vehicle routing problem. Oper Res 61(2):298–314

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

Bianchessi N, Righini G (2007) Heuristic algorithms for the vehicle routing problem with simultaneous pick-up and delivery. Comput Oper Res 34:578–594

Bräysy O, Gendreau M (2005) Vehicle routing problem with time windows, part I: route construction and local search algorithms. Transp Sci 39(1):104–118

Breunig U, Schmid V, Hartl RF, Vidal T (2016) A large neighborhood based heuristic for two-echelon routing problems. Comput Oper Res 76:208–225

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

Clark G, Wright JW (1964) Scheduling of vehicles from a central depot to a number of delivery points. Oper Res 12:568–581

Cordeau JF, Gendreau M, Laporte G (1997) A Tabu search heuristic for periodic and multi-depot vehicle routing problems. Networks 30(2):105–119

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, Mancini S, Perboli G, Tadei R (2011) Multi-start heuristics for the two-echelon vehicle routing problem. In: Merz P, Hao J-K (eds) Evolutionary computation in combinatorial optimization: lecture notes in computer science, vol 6622. Springer, Torino, pp 179–190

Crispim J, Brandao J (2005) Metaheuristics applied to mixed and simultaneous extensions of vehicle routing problems with backhauls. J Oper Res Soc 56:1296–1302

Darvish M, Archetti C, Coelho LC, Speranza MG (2019) Flexible two-echelon location routing problem. Eur J Oper Res 277(3):1124–1136

Dellaert N, Sariaird Q, Woonse 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

Dell’Amico M, Righini G, Salani M (2006) A branch-and-price approach to the vehicle routing problem with simultaneous distribution and collection. Transp Sci 40(2):235–247

Feliu JG, Perboli G, Tadei R, Vigo D (2007) The two-echelon capacitated vehicle routing problem. Technical report DEIS OR.INGCE 20072(R). Department of Electronics, Computer Science, and Systems, University of Bologna, Bologna, Italy

Fleischer K, Osman IH, Hindi KS (2009) A variable neighbourhood search algorithm for the open vehicle routing problem. Eur J Oper Res 195(3):803–809

Fontaine P, Crainic TG, Jabali O, Rei W (2017) The impact of combining inbound and outbound demand in city logistics systems. In: 2017 IEEE 41st annual computer software and applications conference (COMPSAC)

Fontaine P, Crainic TG, Jabali O, Rei W (2021) Scheduled service network design with resource management for two-tier multi-modal city logistics. Eur J Oper Res 294(2):558–570

Gajpal Y, Abad P (2009) An ant colony system (ACS) for vehicle routing problem with simultaneous delivery and pickup. Comput Oper Res 36(2009):3215–3223

Gianessi P, Alfandari L, Lérocalt L, Calvo RW (2016) The multimodality-ring location routing problem. Transp Sci 50(2):541–558

Glover F, Hao JK (2011) The case for strategic oscillation. Ann Oper Res 183(1):163–173

Grangier P, Gendreau M, Lehéudé F, Rousseau L-M (2016) An adaptive large neighborhood search for the two-echelon multiple-trip vehicle routing problem with satellite synchronization. Eur J Oper Res 254(1):80–91

Hansen P, Mladenović N (2001) Variable neighborhood search: principles and applications. Eur J Oper Res 130(3):449–467

Hemmelmayr VC, Doerner KF, Hartl RF (2009) A variable neighborhood search heuristic for periodic routing problems. Eur J Oper Res 195(3):791–802

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 39:3215–3228

Hernández-Pérez H, Landete M, Rodríguez-Martín I (2021) The single-vehicle two-echelon one-commodity pickup and delivery problem. Comput Oper Res 127:105152

Jepsen M, Spoorendonk S, Ropke S (2013) A branch-and-cut algorithm for the symmetric two-echelon capacitated vehicle routing problem. Transp Sci 47(1):23–37

Jie W, Yang J, Zhang M, Huang Y (2019) The two-echelon capacitated electric vehicle routing problem with battery swapping stations: formulation and efficient methodology. Eur J Oper Res 272(3):879–904

Kindervater GAP, Savelsbergh MWP (1997) Vehicle routing: handling edges exchanges windows. In: Aarts E, Lenstra JK (eds)

Koç Ç, Laporte G, Tükenmez I (2020) A review of vehicle routing with simultaneous pickup and delivery. Comput Oper Res 122:104–987

Kytöjoki J, Nuortio T, Bräysy O, Gendreau M (2007) An efficient variable neighborhood search heuristic for very large scale vehicle routing problems. Comput Oper Res 34(9):2743–2757
Liu R, Tao YY, Hu QY, Xie XX (2017) Simulation-based optimisation approach for the stochastic two-echelon logistics problem. Int J Prod Res 55(1):187–201
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
Majidi S, Hosseini-Motlagh SM, Ignatius J (2018) Adaptive large neighborhood search heuristic for pollution-routing problem with simultaneous pickup and delivery. Soft Comput 22(9):2851–2865
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 23:377–386
Mladenović N, Hansen P (1997) Variable neighborhood search. Comput Oper Res 24:1097–1100
Mühlbauer F, Fontaine P (2021) A parallelised large neighborhood search heuristic for the asymmetric two-echelon vehicle routing problem with swap containers for cargo-bicycles. Eur J Oper Res 289(2):742–757
Nagy G, Salhi S (2005) Heuristic algorithms for single and multiple depot vehicle routing problems with pickups and deliveries. Eur J Oper Res 162:126–141
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
Polat O, Kalayci CB, Kulak O, Günther HO (2015) A perturbation based variable neighborhood search heuristic for solving the vehicle routing problem with simultaneous pickup and delivery with time limit. Eur J Oper Res 242:369–382
Prins C (2004) A simple and effective evolutionary algorithm for the vehicle routing problem. Comput Oper Res 31(12):1985–2002
Qiu M, Fu Z, Egles R, Tang Q (2018) A Tabu Search algorithm for the vehicle routing problem with discrete split deliveries and pickups. Comput Oper Res 100:102–116
Rohmer SUK, Claassen GDH, Laporte G (2019) A two-echelon inventory routing problem for perishable products. Comput Oper Res 107:156–172
Santos FA, Mateus GR, Salles da Cunha A (2014) A branch-and-cut-and-price algorithm for the two-echelon capacitated vehicle routing problem. Transp Sci 49(2):355–368
Subramanian A, Uchoa E, Pessoa AA, Ochi LS (2011) Branch-and-cut with lazy separation for the vehicle routing problem with simultaneous pickup and delivery. Oper Res Lett 39:338–341
Subramanian A, Uchoa E, Pessoa AA, Ochi LS (2013) Branch-cut-and-price for the vehicle routing problem with simultaneous pickup and delivery. Optim Lett 7:1569–1581
Vidal T, Crainic TG, Gendreau M, Lahrichi N, Rei W (2012) A hybrid genetic algorithm for multidepot and periodic vehicle routing problems. Oper Res 60(3):611–624
Zachariadis EE, Tarantilis CD, Kiranoudis CT (2010) An adaptive memory methodology for the vehicle routing problem with simultaneous pick-ups and deliveries. Eur J Oper Res 202:401–411
Zachariadis EE, Tarantilis CD, Kiranoudis CT (2016) The vehicle routing problem with simultaneous pick-ups and deliveries and two-dimensional loading constraints. Eur J Oper Res 251(2):369–386
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. Math Probl Eng 2014:1–11
Zhang Z, Cheang B, Li C, Lim A (2019) Multi-commodity demand fulfilment via simultaneous pickup and delivery for a fast fashion retailer. Comput Oper Res 103:816

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