https://dx.doi.org/10.1016/j.trb.2022.06.003

Two-echelon capacitated vehicle routing problem with grouping constraints and simultaneous pickup and delivery

Jiliu Li[a,b], Min Xu[b,*], Peng Sun[c]

[a]*School of Management, Huazhong University of Science and Technology, Wuhan 430074, China*
lijiliu1992@outlook.com
[b]*Industrial and Systems Engineering, The Hong Kong Polytechnic University, Kowloon, Hong Kong*
min.m.xu@polyu.edu.hk
[c]*College of Management and Economics, Tianjin University, Tianjin, 300072, China*
sunpeng1317@tju.edu.cn

Abstract

This paper investigates the two-echelon capacitated vehicle routing problem with grouping constraints and simultaneous pickup and delivery (2E-VRPGS), which is a new variant of the classical two-echelon capacitated vehicle routing problem (2E-VRP). In the 2E-VRPGS, customers from the same administrative region are served by vehicles from the same satellite so as to ensure service consistency, with pickup and delivery being performed simultaneously in the second echelon. To solve this problem to optimality, we formulate it as a path-based model and develop a tailored branch-and-cut-and-price algorithm, which can also exactly solve two closely related variants of 2E-VRPGS in the literature: the 2E-VRP with grouping constraints (2E-VRPG), and the 2E-VRP with simultaneous pickup and delivery (2E-VRPS). In particular, a novel dominance rule in the labeling algorithm, together with several customized valid inequalities, has been put forward to effectively accelerate the solution method by exploiting the problem characteristics. To evaluate the efficacy of the proposed algorithm on the problems 2E-VRPG, 2E-VRPS, and 2E-VRPGS, extensive numerical experiments have been conducted on three types of benchmark instances. Computational results on the 2E-VRPGS show that our dominance rule can significantly reduce the number of generated labels and all families of valid inequalities have a great impact on strengthening the path-based model. The algorithm is found to be highly competitive when compared with the existing exact algorithm for the 2E-VRPG and some new findings and managerial insights are derived from sensitivity analysis.

Keywords: two-echelon vehicle routing, grouping constraints, simultaneous pickup and delivery, valid inequalities, branch-and-cut-and-price.

* Corresponding author

1. Introduction

In logistics management, the vehicle routing problem (VRP) and many other variants have been studied extensively in the literature over the past few decades since it was proposed by Dantzig and Ramser (1959). Given a set of customers with delivery demands and a fleet of homogeneous (capacitated) vehicles, the VRP aims to determine a set of routes for vehicles with the minimal traveling cost to serve the customers, such that each customer is to be visited by a single vehicle and capacity constraints of vehicles are respected (Toth and Vigo, 2014). However, these shipment activities often cause traffic congestion and negative environmental impacts such as air pollution and noise. Hence, many cities have deployed the environmentally-friendly policies that prevent large-capacity trucks from entering into urban areas (Franceschetti et al., 2017). To comply with such legal restrictions, an increasing number of logistics companies have opted for an emerging two-echelon transportation system in which different types of vehicles are used on different echelons. Motivated by this, Crainic et al. (2009) proposed a well-known two-echelon capacitated vehicle routing problem (2E-VRP), and this problem has been widely investigated ever since. The 2E-VRP calls for a set of minimal-cost routes for two types of vehicles in different echelons, where the cargo is delivered from a depot to the satellites by large-capacity vehicles in the first echelon, and then transferred from the satellites to customers using small-capacity vehicles in the second echelon (Crainic et al., 2009; Cuda et al., 2015).

In many practices, customers usually have both demands of pickup and delivery, and vehicles are not only required to deliver cargos to customers but also to pick up other cargos from the customers. Taking beverage shipment for example, the suppliers regularly provide their bottled beverages to a large number of customers every day (restaurants, hotels, school canteens, etc.), and bring back the empty bottles generated on the previous day, either for reuse or for recycling. Since the separation of pickup and delivery will not only cause inconvenience to customers but also hinder the cost savings and efficiency gains achieved by synergizing the pickup and delivery operations, the simultaneous pickup and delivery by vehicles have received much attention from industry and academia in various transportation systems (e.g., Fazi et al., 2020; Hu et al., 2015). This kind of operation considers delivery and pickup activities at the same time by the same vehicles and all delivered cargos are originated from the depot and all pickup cargos are transported back to the depot. Likewise, the activities of simultaneous pickup and delivery should conform to the legal restrictions in many cities, which arouses a new two-echelon transportation system with simultaneous pickup and delivery. Although such a transportation system is practically relevant, so far no one has ever investigated it except for Belgin et al. (2018) and Zhou et al. (2022). The former introduced a 2E-VRP with simultaneous pickup and delivery (2E-VRPS) where vehicles are

allowed to pick up and deliver cargos at the same time in both echelons. The latter proposed an alternative transportation mode in a two-echelon system where pickups and deliveries are separated in the first echelon but combined in the second echelon.

With the continuous growth of economy and the substantial increase of demands of customers, for some logistics companies the capacities of satellites for temporarily storing cargos in a two-echelon transportation system can no longer meet the current requirements. To adapt to the large volume of demand, the original structure of two-echelon system needs to be reorganized by establishing large-capacity satellites to replace a mass of small-capacity satellites (Liu et al., 2018). Moreover, the reorganization can also considerably reduce the management cost incurred by the large number of existing small-capacity satellites. To ensure service consistency, the operators of new satellites had better manage their original customers, resulting in customers from the same small-capacity satellite or the same administrative region being assigned to the same new large-capacity satellite. This business-level operation will bring a new kind of constraints into the 2E-VRP, referred to as the *grouping constraints*, that is, customers in the same group are served by the same satellite. Motivated by this, Liu et al. (2018) proposed the 2E-VRP with grouping constraints (2E-VRPG). In addition, maintaining service quality throughout has been emphasized in the studies of consistent vehicle routing problems (conVRPs). In the conVRPs, each customer can be served by the same driver (vehicle, path) at approximately the same time over multiple days (e.g., Groër et al., 2009; Yao et al., 2021). The grouping constraints coincide with the concept of conVRP, but the detailed contents are distinct.

To the best of our knowledge, the existing literature has not jointly considered the above two realistic features – simultaneous pickup and delivery and grouping constraints – in the two-echelon transportation system. To close the gap, we address in the paper a variant called the 2E-VRP with grouping constraints and simultaneous pickup and delivery (2E-VRPGS). In this problem, the pickups and deliveries are separated in the first echelon, whereas they are performed simultaneously in the second echelon. Without loss of generality, the proposed model and solution method can be extended to handle the general case with simultaneous pickup and delivery in both echelons (see Sections 2.2 and 5.3). More importantly, we further analyse the influence of various system parameters on the new transportation system, and provide some managerial insights and useful suggestions for logistics companies.

1.1. Literature review

As the 2E-VRPGS has never appeared in the literature, in this section we will first review two types of relevant classical problems: the vehicle routing problem with simultaneous pickup and delivery (VRPSPD) and the 2E-VRP. In addition, the two most related problems of the 2E-

VRPGS, i.e., 2E-VRPS and 2E-VRPG, will also be discussed.

The literature on optimization problems with simultaneous pickup and delivery has increased substantially over the past decade. However, most papers essentially studied single-echelon transportation systems, referred to as the VRPSPD. The VRPSPD was first introduced by Min (1989) for solving a book distribution and collection problem at a library and has since evolved into a rich and active research field. Compared with the classical VRP, this kind of problems is more difficult to handle, because the simultaneous distribution and collection highly complicate the cargo flows. Many heuristic methods have been proposed to solve it or its variants, including a hybrid heuristic algorithm developed by Wang et al. (2013), an adaptive large neighborhood search proposed by Qu and Bard (2013), an adaptive local search algorithm developed by Avci and Topaloglu (2015), and a greedy clustering method put forward by Nadizadeh and Kafash (2019). A few attempts for exact methods have also been made, such as the branch-and-cut algorithm proposed by Subramanian et al. (2011) and the branch-and-cut-and-price algorithms put forward by Subramanian et al. (2013) and Qu and Bard (2015). We refer the interested readers to the review paper by Koç et al. (2020) for more algorithms and variants of the VRPSPD.

Our problem can be treated as an extension of the classical optimization problems on two-echelon transportation systems, 2E-VRP, which considers simultaneous pickup and delivery and grouping constraints together. The literature on the 2E-VRP started with an integrated short-term scheduling problem in a two-tiered distribution system in city logistics (Crainic et al., 2009). Compared with the classical VRP, the interaction and interdependence between the two levels make the 2E-VRP more complex and challenging. Therefore, most of the existing studies on the 2E-VRP sought heuristic methods to address the problem. For example, to solve the classical 2E-VRP, Perboli et al. (2011), Kergosien et al. (2013), Zeng et al. (2014), and Breunig et al. (2016) have proposed some high-efficiency heuristics. Apart from heuristic approaches, some exact methods for the 2E-VRP were also investigated. To solve the classical 2E-VRP to optimality, Gonzalez-Feliu (2008) introduced a set covering formulation and developed a column generation based approach, Perboli and Tadei (2010) and Perboli et al. (2011) came up with the branch-and-cut algorithms with some specific families of valid inequalities. Baldacci et al. (2013) put forward a tailor-made exact method with several bounding procedures. Santos et al. (2013), Santos et al. (2015), and Marques et al. (2020) introduced the branch-and-price algorithms, where the set of all first-echelon routes was known. The latter two algorithms are essentially branch-and-cut-and-price algorithms as they used some valid inequalities to enhance lower bounds.

There are also many studies for variants of 2E-VRP that consider a variety of features such as the synchronization of time, the special natures of satellites, the uses of specific vehicles, the demands of pickup and delivery, and time windows. For example, Li et al. (2016) investigated the

time-constrained 2E-VRP in a synchronous linehaul-delivery systems and put forward a Clarke and Wright savings heuristic to solve it. Li et al. (2018) later introduced the 2E-VRP with real-time transshipment capacity and designed a two-stage heuristic for it. Subsequently, by introducing electric vehicles and unmanned aerial vehicles, Breunig et al. (2019) and Li et al. (2020a) proposed the electric 2E-VRP and the 2E-VRP with time windows and mobile satellites, respectively, and both of them developed the large neighborhood search heuristics for the problems. As aforementioned, Belgin et al. (2018) first investigated the 2E-VRPS where the activities of pickup and delivery simultaneously happen in both echelons by a hybrid heuristic based on variable neighborhood descent and local search. Besides, Zhou et al. (2022) addressed a different variant with time windows where the activities of simultaneous pickup and delivery were only allowed in the second echelon by a variable neighborhood tabu search. As for the exact algorithms for the variants of 2E-VRP, Jepsen et al. (2013) and Liu et al. (2018) proposed the symmetric version of 2E-VRP and the 2E-VRPG, respectively. To solve the problems to optimality, they both came up with the branch-and-cut algorithms with several customized families of inequalities based on arc-flow-based models. Most recently, the 2E-VRP with time windows was studied by Dellaert et al. (2019) and Mhamedi et al. (2022). They proposed a branch-and-price-based algorithm and a branch-and-cut-and-price algorithm to optimally solve it, respectively. Computational results showed that the later has the greater advantage. In addition, Marques et al. (2022) studied the variant of the 2E-VRP with precedence constraints for unloading and loading freight at satellites. In this variant allows for storage and consolidation of freight at satellites. Routes at the second echelon connecting satellites and customers may consist of multiple trips and visit several satellites. They proposed a branch-and-cut-and-price algorithm to solve efficiently the problem.

Table 1. The summary of exact methods and heuristics for the 2E-VRP variants

Reference	Limited number of vehicles	Handling costs at satellites	Time windows	Simultaneous pickup and delivery	Grouping constraints	Algorithm	
						Exact	Heuristic
Jepsen et al. (2013)	✓	✓				BC	
Santos et al. (2013)	✓	✓				BP	
Baldacci et al. (2013)	✓	✓				EM	
Santos et al. (2015)	✓	✓				BCP	
Li et al. (2016)	✓	✓	✓				✓
Li et al. (2018)	✓						✓
Belgin et al. (2018)				1^{st}+2^{nd}			✓
Liu et al. (2018)	✓	✓			✓	BC	
Dellaert et al. (2019)			✓			BPB	
Breunig et al. (2019)	✓						✓
Li et al. (2020a)	✓		✓				✓
Marques et al. (2020)	✓	✓				BCP	
Mhamedi et al. (2022)			✓			BCP	
Zhou et al. (2022)	✓		✓	2^{nd}			✓
Marques et al. (2022)			✓			BCP	
This paper	✓	✓		2^{nd} and 1^{st}+2^{nd}	✓	BCP	

⋆ BC: branch-and-cut algorithm; BPB: branch-and-price-based algorithm; BCP: branch-and-cut-and-price algorithm; EM: tailor-made exact method.

Table 1 summarizes the above mentioned exact methods and heuristics for the 2E-VRP variants.

In the Table 1, we classify the literature by the following assumptions: (i) whether the number of vehicles is limited; (ii) whether the handling cost for cargo transshipment at satellites should be paid; (iii) whether the time windows are considered; (iv) which echelon allows the activities of simultaneous pickup and delivery; (v) whether the grouping constraints are considered; and (vi) which algorithm is used. As mentioned before, the 2E-VRPS and 2E-VRPG are two problems closely related to the 2E-VRPGS. We can see from Table 1 that only Belgin et al. (2018) and Zhou et al. (2022) investigated the 2E-VRPS. Due to the complex nature of the problem, they all focused on heuristics and up to now no one has studied any exact methods for the 2E-VRPGS. Besides this, although very relevant in practice, so far no one other than Liu et al. (2018) has ever considered the grouping constraints in the two-echelon transportation system. Regrettably, the study of Liu et al. (2018) ignored the simultaneous pickup and delivery requests of customers, which are commonly seen in real-world logistics transportation systems.

1.2. Objective and contributions

Based on the real-world transport scenario, this study introduces a new and practical variant of the 2E-VRP combining grouping constraints and simultaneous pickup and delivery (2E-VRPGS). To ensure the original level of service after satellite reorganization, customers from the same original administrative region will be served by the second-echelon vehicles from the same satellite. The objective of the problem is to minimize the total operating cost by determining the optimal vehicle routes for both of the two echelons and the assignment of customer groups to the satellites without violating the capacity constraints (of vehicles and satellites). The contributions are summarized as follows:

- This paper is the first to consider both the grouping constraints and simultaneous pickup and delivery in the 2E-VRP. The problem is formulated into a tight path-based model, which is also capable of formulating the 2E-VRPG and 2E-VRPS after minor modifications.
- To solve the 2E-VRPGS to optimality, an exact branch-and-cut-and-price (BCP) algorithm is developed. Specifically, a novel dominance rule in the labeling algorithm is proposed to remove more non-potential labels than the standard one. Furthermore, several families of customized valid inequalities extended from the classical ones in the literature are put forward. The computational results show that both the proposed dominance rule and valid inequalities have a positive impact on the algorithm.
- The proposed algorithm is able to solve the two most closely related variants of 2E-VRPGS in the literature, i.e., the 2E-VRPS and 2E-VRPG. To the best of our knowledge, this paper is the first to provide an exact algorithm for the 2E-VRPS even though it has already been investigated in the literature. Besides this, the numerical experiments have demonstrated that

the proposed algorithm is highly competitive compared with the existing exact algorithm on the 2E-VRPG.

- A comprehensive system analysis is conducted on some sensitive factors of the new transportation system. Finally, we provide some interesting findings and meaningful suggestions for logistics companies.

The remainder of this paper is organized as follows. In Section 2, we provide a problem description and a path-based model for the 2E-VRPGS. The ways to formulate the problems 2E-VRPG and 2E-VRPS based on the proposed model are also discussed. Section 3 develops a column generation with a novel dominance rule to solve the LP-relaxation of the path-based model, and then Section 4 comes up with four families of valid inequalities to strengthen it. Section 5 designs a BCP algorithm to address the path-based model, and adjustments to the proposed algorithm for solving the problems 2E-VRPG and 2E-VRPS are also described. In Section 6, numerical experiments on three types of instances are conducted to evaluate the effectiveness of the algorithm, and the performance of dominance rule, the performances of valid inequalities, and the impacts of system parameters are systematically analysed. Finally, Section 7 concludes the paper with main findings and a brief discussion of future research directions.

2. Problem description and mathematical formulation

The 2E-VRPGS is defined on a directed graph $G = (V, A)$, where V and A are node set and arc set, respectively. Node set V consists of a depot $V_0 := \{0\}$, a set of satellites V_S, and a set of customers V_C, i.e., $V = V_0 \cup V_S \cup V_C$. Each satellite $s \in V_S$ has a capacity Q^s and a handling cost h_s for every unit of freight. Each customer $i \in V_C$ has a delivery demand d_i and a pickup demand p_i. In addition, customer set V_C is made up by l disjoint groups such that $V_C = C_1 \cup C_2 \cup ... \cup C_l$ and $C_i \cap C_j = \varnothing$ for $\forall i \neq j$. Customers in the same group should be served by vehicles from the same satellite. The index set of customer groups is denoted by L, namely $L = \{1, 2, ..., l\}$. Arc set A consists of two types: the first-echelon arcs connecting the depot and the satellites, grouped in set $A_1 = \big\{(i,j) \mid i,j \in V_0 \cup V_S, i \neq j\big\}$, and the second-echelon arcs between satellites and customers, grouped in set $A_2 = \big\{(i,j) \mid i,j \in V_S \cup V_C, i \neq j\big\} \backslash \big\{(i,j) \mid i,j \in V_S, i \neq j\big\}$. Each arc $(i,j) \in A = A_1 \cup A_2$ is associated with a traveling cost denoted by c_{ij}. As for the vehicles, a homogeneous vehicle fleet will be utilized in each echelon, but the capacities of vehicles for the first and the second echelons, denoted by Q_1 and Q_2, respectively, can be different. The number of available vehicles in each echelon is also limited. Let K_1^d and K_1^p be the set of available first-echelon vehicles for distribution and collection, respectively, K_2 be the set of available second-echelon vehicles, and T_s be the set of available second-echelon vehicles departing from each satellite $s \in V_S$. The first-echelon vehicles fall into two classes: (i) one class of vehicles are used to deliver

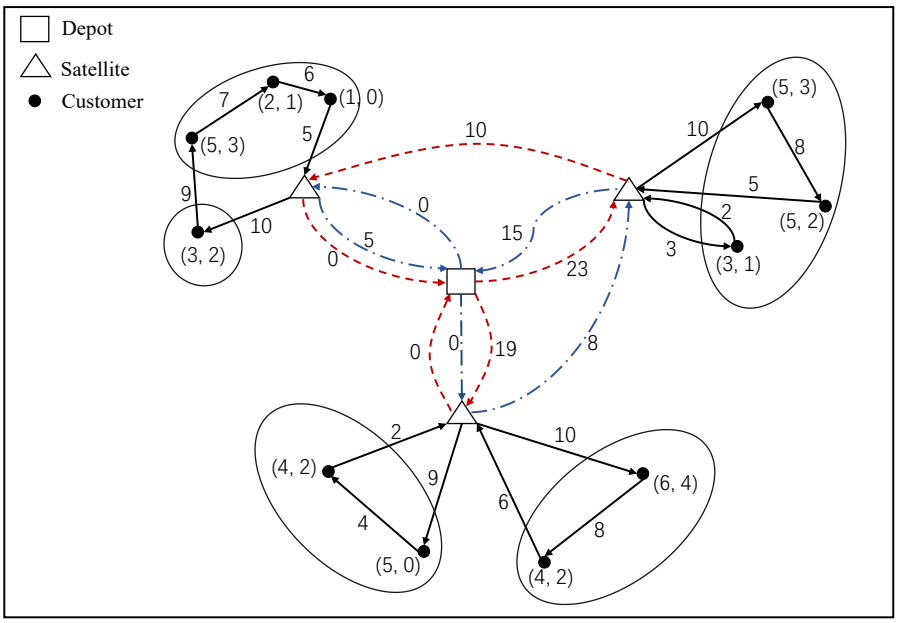

Figure 1. An example for 2E-VRPGS

the cargo from the depot to the satellites, the routes they use being called *distribution routes*; (ii) the other class of vehicles are employed to pick up the cargo from satellites to depot, these routes being called *collection routes*.

The 2E-VRPGS aims to determine a set of routes, including the first-echelon distribution routes, the first-echelon collection routes, and the second-echelon routes. The objective is to minimize the total operating cost, including the traveling cost and the handling cost at satellites, under the following assumptions:

- Each satellite can be visited more than once in the first echelon, whereas each customer is visited exactly once in the second echelon;
- Each first-echelon vehicle is assigned to at most one distribution route or at most one collection route;
- Each first-echelon route starts from and ends at the depot;
- Each second-echelon route starts from and ends at the same satellite;
- The capacities of the first- and second-echelon vehicles cannot be violated;
- The total pickup load and delivery load assigned to a satellite cannot exceed its capacity;

Figure 1 illustrates an instance of 2E-VRPGS with three satellites and five groups of customers enclosed by ellipses. A tuple next to each customer contains the values of its delivery demand and pickup demand in turn (this also applies to the later tuples in Figures 2 and 3). The loads of vehicles when traversing the arcs are represented by the numbers beside the arrows. In Figure 1, there are two first-echelon distribution routes (dotted arrows), two first-echelon collection routes, and five second-echelon routes (solid arrows).

2.1. Path-based (PB) model

This subsection introduces a path-based model in which the first- and second-echelon routes are decomposed in the same manner as the formulation for the classical 2E-VRP in Baldacci et al. (2013). The set of first-echelon routes is defined as $\Phi = \Phi^1 \cup \Phi^2$, where Φ^1 is the set of distribution routes and Φ^2 is the set of collection routes. Let Φ_S be the set of routes in Φ visiting at least one satellite in S, where $S \subseteq V_S$ is a subset of satellites. Let Φ_S^1 and Φ_S^2 be the sets of distribution and collection routes in Φ_S, respectively, i.e., $\Phi_S = \Phi_S^1 \cup \Phi_S^2$. Let S_ϕ be the set of satellites visited by a first-echelon route $\phi \in \Phi$. The cost of a first-echelon route $\phi \in \Phi$ is denoted by $c_\phi = \sum_{(i,j)\in A_1} \beta_{ij\phi}^1 c_{ij}$, where $\beta_{ij\phi}^1$ is the number of times the arc $(i,j) \in A_1$ is traversed by the first-echelon route $\phi \in \Phi$. All the first-echelon distribution and collection routes are exhaustively enumerated, which is plausible when the number of satellites is not large. For each non-empty subset $S \subseteq V_S$, a first-echelon collection route and a first-echelon distribution route can be constructed by solving a traveling salesman problem on the set $S \cup \{0\}$. Therefore, a total of $2^{|V_S|-1}$ first-echelon distribution (collection) routes will be generated. Let $R = \bigcup_{s\in V_S} R_s$ be the set of feasible second-echelon routes, where R_s is the set of feasible second-echelon routes departing from satellite $s \in V_S$. For each second-echelon route $r \in R$, let α_{ir} denote the number of times the customer i is visited by route r. The cost of a second-echelon route $r \in R$ is represented by $c_r = \sum_{(i,j)\in A_2} \beta_{ijr}^2 c_{ij}$, where β_{ijr}^2 is the number of times the arc $(i,j) \in A_2$ is traversed by the second-echelon route $r \in R$.

To formulate the 2E-VRPGS upon the first- and second-echelon routes, the following decision variables are used:

- x_ϕ: a binary variable indicating whether ($x_\phi = 1$) or not ($x_\phi = 0$) a first-echelon route $\phi \in \Phi$ is selected;
- θ_r: a binary variable indicating whether ($\theta_r = 1$) or not ($\theta_r = 0$) a second-echelon route $r \in R$ is selected;
- w_ϕ^s: the amount of cargo distributed to satellite $s \in S_\phi$ from the depot by route $\phi \in \Phi^1$;
- g_ϕ^s: the amount of cargo collected from satellite $s \in S_\phi$ to the depot by route $\phi \in \Phi^2$;
- z_{ks}: a binary variable indicating whether ($z_{ks} = 1$) or not ($z_{ks} = 0$) a group $C_k, k \in L$ is assigned to satellite $s \in V_S$.

With the aforementioned notation, the PB model for 2E-VRPGS is presented as follows:

$$\min \sum_{\phi\in\Phi} c_\phi x_\phi + \sum_{r\in R} c_r \theta_r + \sum_{\phi\in\Phi^1} \sum_{s\in S_\phi} h_s w_\phi^s + \sum_{\phi\in\Phi^2} \sum_{s\in S_\phi} h_s g_\phi^s \tag{1}$$

$$\text{s.t.} \sum_{\phi\in\Phi^1} x_\phi \le |K_1^d| \tag{2}$$

$$\sum_{\phi \in \Phi^2} x_\phi \leq |K_1^p| \tag{3}$$

$$\sum_{s \in S_\phi} w_\phi^s \leq Q_1 x_\phi, \forall \phi \in \Phi^1 \tag{4}$$

$$\sum_{s \in S_\phi} g_\phi^s \leq Q_1 x_\phi, \forall \phi \in \Phi^2 \tag{5}$$

$$\sum_{\phi \in \Phi^1_{\{s\}}} w_\phi^s \leq Q^s, \forall s \in V_S \tag{6}$$

$$\sum_{\phi \in \Phi^2_{\{s\}}} g_\phi^s \leq Q^s, \forall s \in V_S \tag{7}$$

$$\sum_{r \in R} \theta_r \leq |K_2| \tag{8}$$

$$\sum_{r \in R_s} \theta_r \leq |T_s|, \forall s \in V_S \tag{9}$$

$$\sum_{r \in R_s} \sum_{i \in V_C} d_i \alpha_{ir} \theta_r \leq \sum_{\phi \in \Phi^1_{\{s\}}} w_\phi^s, \forall s \in V_S \tag{10}$$

$$\sum_{r \in R_s} \sum_{i \in V_C} p_i \alpha_{ir} \theta_r \leq \sum_{\phi \in \Phi^2_{\{s\}}} g_\phi^s, \forall s \in V_S \tag{11}$$

$$\sum_{r \in R_s} \alpha_{ir} \theta_r \geq z_{ks}, \forall i \in C_k, k \in L, s \in V_S \tag{12}$$

$$\sum_{s \in V_S} z_{ks} = 1, \forall k \in L \tag{13}$$

$$x_\phi \in \mathbb{N}, \forall \phi \in \Phi \tag{14}$$

$$\theta_r \in \{0, 1\}, \forall r \in R \tag{15}$$

$$w_\phi^s \geq 0, \forall \phi \in \Phi^1, s \in S_\phi \tag{16}$$

$$g_\phi^s \geq 0, \forall \phi \in \Phi^2, s \in S_\phi \tag{17}$$

$$z_{ks} \in \{0, 1\}, \forall k \in L, s \in V_S \tag{18}$$

The objective function (1) minimizes the sum of the total travel cost induced by the first- and second-echelon routes and the total handling cost at satellites. Constraints (2)-(3) ensure that the numbers of the first-echelon distribution and collection routes do not exceed the pre-specified limits. Constraints (4)-(7) are the capacity constraints of each first-echelon vehicle and each satellite. Constraints (8)-(9) impose the fleet size limitations for the second echelon and each satellite. Constraints (10) guarantee that the total amount of distribution to customers from a satellite does not exceed the amount received by this satellite. Constraints (11) ensure that the total amount of collection from customers to a satellite does not exceed the amount shipped out from this satellite. Constraints (12)-(13) are the grouping constraints and also ensure that each customer

is visited exactly once. Constraints (14)-(18) describe the feasible domains of decision variables. Since all the pickup and delivery demands of customers must be served, any feasible solution should have at least $\lceil \sum_{i \in V_C} d_i/Q_1 \rceil$ and $\lceil \sum_{i \in V_C} p_i/Q_1 \rceil$ first-echelon distribution and collection routes, respectively, and at least $\lceil \max\{\sum_{i \in V_C} d_i, \sum_{i \in V_C} p_i\}/Q_2 \rceil$ second-echelon routes. The following three inequalities are valid to strengthen the PB model: (i) $\sum_{\phi \in \Phi^1} x_\phi \geq \lceil \sum_{i \in V_C} d_i/Q_1 \rceil$; (ii) $\sum_{\phi \in \Phi^2} x_\phi \geq \lceil \sum_{i \in V_C} p_i/Q_1 \rceil$; and (iii) $\sum_{r \in R} \theta_r \geq \lceil \max\{\sum_{i \in V_C} d_i, \sum_{i \in V_C} p_i\}/Q_2 \rceil$. In addition, a satellite must be visited by at least one second-echelon vehicle if some groups are assigned to it. Hence, a set of valid inequalities is also obtained to enhance the model: $z_{ks} \leq \sum_{\phi \in \Phi^1_{\{s\}}} x_\phi + \sum_{\phi \in \Phi^2_{\{s\}}} x_\phi, \forall k \in L, s \in S$.

2.2. Models for 2E-VRPG and 2E-VRPS

This subsection discusses how to modify the proposed PB model to formulate two related variants, i.e., 2E-VRPG and 2E-VRPS.

Compared with 2E-VRPGS, the 2E-VRPG overlooks the pickup requests of customers. Without constraints (3), (5), (7), (11), (17), and the last term of the objective function (1), the PB model is capable of formulating the 2E-VRPG. As aforementioned, the 2E-VRPS allows simultaneous pickup and delivery in both the first and second echelons but lacks the grouping constraints. The following steps can be done to formulate the 2E-VRPS based on the PB model:

(i) Construct $|V_C|$ groups such that each group corresponds to a customer;

(ii) Remove the set K_1^p and Φ^2; redefine K_1^d and Φ^1 as the sets of the first-echelon vehicles and routes for simultaneous pickup and delivery, respectively, and g_ϕ^s as the amount of cargo collected from satellite s by route $\phi \in \Phi^1$;

(iii) Replace the set Φ^2 by Φ^1 in the constraints (5), (7), (11), (17), and the last term of the objective function (1); remove constraints (3);

(iv) Add the constraints $\sum_{s' \in S_\phi^a(s)} w_\phi^{s'} + g_\phi^s + \sum_{s' \in S_\phi^b(s)} g_\phi^{s'} \leq Q_1 x_\phi, \forall s \in S_\phi, \phi \in \Phi^1$, where $S_\phi^a(s)$ and $S_\phi^b(s)$ are the sets of satellites visited by route $\phi \in \Phi^1$ after and before visiting the satellite $s \in S_\phi$, respectively.

Recently, Liu et al. (2018) and Belgin et al. (2018) have also proposed arc-flow-based (AFB) models based on the vehicle flows on the arcs for the problems 2E-VRPG and 2E-VRPS, respectively. Unlike the AFB models, the PB model has a high-quality lower bound obtained by its LP-relaxation, but a huge number of second-echelon routes that are difficult to be explicitly formulated. Therefore, both the PB model and its LP-relaxation cannot be directly solved by the MIP solver. However, they can be handled by the well-developed BCP algorithm in Section 5 and the well-designed column generation approach in Section 3, respectively.

To make it easier for the reader to understand the problems and models, we report all the

notation introduced above in Appendix A.

3. Column generation

This section develops a column generation to solve the LP-relaxation (referred to as the linear master problem, LMP) of the PB model. For an overview of column generation, we refer the reader to Desaulniers et al. (2006). Column generation is an iterative procedure where a *restricted linear master problem* (RLMP) and a *pricing problem* are solved alternately. The RLMP is obtainned from the LMP by replacing the set R_s by the small-sized subset $\bar{R}_s \subseteq R_s$ for each satellite $s \in V_S$. The pricing problem formulated by the solution to the dual problem of the RLMP is to identify the columns (second-echelon routes) that have negative reduced costs. If the set $\bar{R} = \bigcup_{s \in V_S} \bar{R}_s$ satisfies the property that none of the columns in $R \backslash \bar{R}$ has negative reduced cost, the optimal solution to the RLMP will also be optimal for the LMP. The column generation iteration will stop if no columns with negative reduced cost are found, which implies that the optimal solution to the LMP is obtained. Otherwise, we will add one or more such columns (at most 50 columns in our implementation) into the RLMP and start the next column generation iteration.

3.1. Pricing problem

Let $(\boldsymbol{\rho}, \boldsymbol{\tau}, \boldsymbol{\mu})$ be a solution to the dual problem of the LMP, where $\rho_0 \leq 0$, $\rho_s \leq 0$ $(s \in V_S)$, $\tau_s^d \leq 0$ $(s \in V_S)$, and $\tau_s^p \leq 0$ $(s \in V_S)$, and $\mu_{is} \geq 0$ $(i \in V_C, s \in V_S)$ are the dual values of constraints (8), (9), (10), (11), and (12), respectively. Let Λ_{is} be the contribution to the reduced cost if customer i is allocated to a satellite $s \in V_S$, which is calculated by $\Lambda_{is} = d_i \tau_s^d + p_i \tau_s^p + \mu_{is}$. The pricing problem to determine a feasible second-echelon route with the minimum reduced cost will thus be formulated as follows:

$$\min_{s \in V_S, r \in R_s} \bar{c}_r = \sum_{(i,j) \in A_2} \beta_{ijr}^2 c_{ij} - \sum_{i \in V_C} \alpha_{ir} \Lambda_{is} - \rho_0 - \rho_s \tag{19}$$

where the second-echelon routes in R_s are elementary and feasible in terms of vehicle capacity. It is a variant of the elementary shortest path problem with resource constraints (ESPPRC), which is konwn to be NP-hard (Dror, 1994). The ESPPRC arising in the column generation approach is typically solved using a dynamic programming technique called a labeling algorithm. The partial paths are first built along the forward direction and backward direction and are then connected together to obtain the complete routes. The existing partial paths are extended to construct other new paths following certain extending conditions and updating functions. To accelerate the algorithm, the non-potential partial paths identified by several sufficient rules can be abandoned during the search. In the rest of Section 3, the components mentioned above in the labeling algorithm will be described in detail.

3.2. Forward labeling

Each forward partial path is represented by a forward label ℓ_f coded as $\big[\eta(\ell_f), \bar{c}(\ell_f), s(\ell_f), \sigma(\ell_f), \pi(\ell_f), N(\ell_f), V(\ell_f)\big]$ with the following elements:

- $\eta(\ell_f)$: the last vertex visited by the partial path;
- $\bar{c}(\ell_f)$: the cumulative reduced cost of the partial path;
- $s(\ell_f)$: the satellite from which the partial path starts;
- $\sigma(\ell_f)$: the amount of cargo that the second-echelon vehicle can deliver after visiting $\eta(\ell_f)$;
- $\pi(\ell_f)$: the amount of cargo that the second-echelon vehicle can collect after visiting $\eta(\ell_f)$;
- $N(\ell_f)$: the set of customers that have already been visited along the partial path;
- $V(\ell_f)$: the set of reachable customers that the partial path can visit afterward (a customer is reachable if it has not yet been visited and the vehicle capacity constraints will still be respected when the label extends to it).

The forward labeling procedure starts with a set of initial labels at satellites, that is, $\big\{[s, -\rho_0 - \rho_s, s, Q_2, Q_2, \varnothing, V_C]\big\}_{s \in V_S}$. Given a label $\ell_f' = \big[\eta(\ell_f'), \bar{c}(\ell_f'), s(\ell_f'), \sigma(\ell_f'), \pi(\ell_f'), N(\ell_f'), V(\ell_f')\big]$ starting from a satellite $s = s(\ell_f')$ and ending at a vertex $i = \eta(\ell_f')$, its extension along an arc (i, j) to the vertex $j \in V(\ell_f')$ is feasible only if the following extending conditions are satisfied: $\sigma(\ell_f') \geq d_j$ and $\pi(\ell_f') \geq p_j$. That is, a label ℓ_f' can be extended to a vertex j when it has not been visited before and the remaining vehicle capacity is enough to deal with delivery demand d_j and pickup demand p_j. If the extension is feasible, then a new label ℓ_f will be created by the following updating functions:

$$\eta(\ell_f) = j \tag{20}$$

$$\bar{c}(\ell_f) = \bar{c}(\ell_f') + c_{ij} - \Lambda_{js} \tag{21}$$

$$s(\ell_f) = s(\ell_f') \tag{22}$$

$$\sigma(\ell_f) = \min\big\{\sigma(\ell_f') - d_j, \pi(\ell_f') - p_j\big\} \tag{23}$$

$$\pi(\ell_f) = \pi(\ell_f') - p_j \tag{24}$$

$$N(\ell_f) = N(\ell_f') \cup \{j\} \tag{25}$$

$$V(\ell_f) = V(\ell_f') \backslash \Big(\{j\} \cup \big\{k \in V(\ell_f') | \sigma(\ell_f) < d_k\big\} \cup \big\{k \in V(\ell_f') | \pi(\ell_f) < p_k\big\}\Big) \tag{26}$$

Figure 2 illustrates the updates of resources σ and π when $Q_2 = 12$. Suppose that a path $(0, 1, 2)$ is going to extend to customer 3. For ease of presentation, let σ_i' and π_i' be the amount of cargo that the vehicle can deliver and collect after vertex i, respectively, and let f_i' be the load of the vehicle after vertex i. We have

$f_0' = d_1 + d_2 = 6$; $f_1' = f_0' - d_1 + p_1 = 8$; $f_2' = f_1' - d_2 + p_2 = 7$;

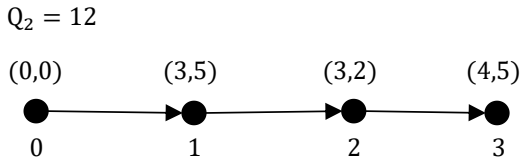

Figure 2. The updates of resources σ and π

$\sigma_2' = Q_2 - \max\{f_0', f_1', f_2'\} = 4$ $\pi_2' = Q_2 - f_2' = 5$;

After the extension to customer 3, let f_i, σ_i, and π_i be the new parameters having the same meanings as f_i', σ_i', and π_i', respectively. We have

$f_0 = f_0' + d_3 = 10$; $f_1 = f_1' + d_3 = 12$; $f_2 = f_2' + d_3 = 11$; $f_3 = f_2 - d_3 + p_3 = f_2' + p_3 = 12$;

$$\begin{aligned}
\sigma_3 &= Q_2 - \max\{f_0, f_1, f_2, f_3\} \\
&= Q_2 - \max\{f_0' + d_3, f_1' + d_3, f_2' + d_3, f_2' + p_3\} \\
&= \min\left\{Q_2 - \max\{f_0', f_1', f_2'\} - d_3, (Q_2 - f_2') - p_3\right\} \\
&= \min\{\sigma_2' - d_3, \pi_2' - p_3\} \\
&= 0
\end{aligned}$$

$\pi_3 = Q_2 - f_3 = Q_2 - f_2' - p_3 = \pi_2' - p_3 = 0$;

Therefore, we have $\sigma_3 = \min\{\sigma_2' - d_3, \pi_2' - p_3\}$ and $\pi_3 = \pi_2' - p_3$, which correspond to the updating functions (23) and (24), respectively.

In terms of updating functions, all possible extensions will be performed and stored as labels. The algorithm in principle enumerates all feasible routes by starting with the initial labels, which can be computationally expensive. However, many *dominated* labels are capable of being removed during the search. Let ℓ_f^1 and ℓ_f^2 denote the two labels with $\eta(\ell_f^1) = \eta(\ell_f^2)$; and let q_f^1 and q_f^2 be their corresponding forward partial paths, respectively. Given a forward partial path q_f, let $\mathscr{P}(q_f)$ be the set of all the backward partial paths that can be used to complete it. Let $q_f \oplus q_b$ represent the complete path obtained by joining the forward partial path q_f and the backward partial path $q_b \in \mathscr{P}(q_f)$, and $\bar{c}(q_f \oplus q_b)$ be its reduced cost. Label ℓ_f^1 dominates label ℓ_f^2 if the following two *standard conditions* hold: (i) all the backward partial paths that are feasible to complete path q_f^1 can also be used to complete q_f^2, i.e., $q_b \in \mathscr{P}(q_f^1), \forall q_b \in \mathscr{P}(q_f^2)$; and (ii) the reduced cost of $q_f^1 \oplus q_b$ is less than or equal to that of $q_f^2 \oplus q_b$, i.e., $\bar{c}(q_f^1 \oplus q_b) \leq \bar{c}(q_f^2 \oplus q_b), \forall q_b \in \mathscr{P}(q_f^2)$. If $s(\ell_f^1) \neq s(\ell_f^2)$, any backward partial path in $\mathscr{P}(q_f^2)$ cannot be used to complete the partial path q_f^1. Therefore, according to the standard conditions, any two labels with different satellites are forbidden from mutually dominating according to the standard conditions. We discover the following less-restrictive dominance rule that enables the algorithm to compare labels originating from different satellites, and its proof is given in Appendix B.

Dominance 1. *Let ℓ_f^1 and ℓ_f^2 denote the two labels with $\eta(\ell_f^1) = \eta(\ell_f^2) = i$ such that*

$$V(\ell_f^1) \supseteq V(\ell_f^2) \tag{27}$$

$$\bar{c}(\ell_f^1) + \max_{k \in V(\ell_f^2)} \left\{ c_{ks_1} - c_{ks_2} \right\} \leq \bar{c}(\ell_f^2) + \sum_{k \in V(\ell_f^2)} \min \left\{ 0, \Lambda_{ks_2} - \Lambda_{ks_1} \right\} \tag{28}$$

$$\sigma(\ell_f^1) \geq \sigma(\ell_f^2) \tag{29}$$

$$\pi(\ell_f^1) \geq \pi(\ell_f^2) \tag{30}$$

where $s_1 = s(\ell_f^1)$ and $s_2 = s(\ell_f^2)$ for the sake of presentation, then label ℓ_f^1 dominates label ℓ_f^2.

3.3. Backward labeling

The backward labeling procedure follows the opposite direction and generates labels from the destination satellite. Similar to the forward labeling, each backward label ℓ_b is coded as $\left[\eta(\ell_b), \bar{c}(\ell_b), s(\ell_b), \sigma(\ell_b), \pi(\ell_b), N(\ell_b), V(\ell_b)\right]$. The meanings of elements are the same as that of the forward labels. The backward labeling procedure starts with a set of initial labels $\left\{[s, 0, s, Q_2, Q_2, \varnothing, V_C]\right\}_{s \in V_S}$. A backward label $\ell_b^{'}$ starting from $s = s(\ell_b^{'})$ and ending at vertex $i = \eta(\ell_b^{'})$ can be extended along an arc (j, i) to create a new label ℓ_b if the same extending conditions as with the forward case are met. The elements in label ℓ_b are updated by

$$\eta(\ell_b) = j \tag{31}$$

$$\bar{c}(\ell_b) = \bar{c}(\ell_b^{'}) + c_{ji} - \Lambda_{js} \tag{32}$$

$$s(\ell_b) = s(\ell_b^{'}) \tag{33}$$

$$\sigma(\ell_b) = \sigma(\ell_b^{'}) - d_j \tag{34}$$

$$\pi(\ell_b) = \min \left\{ \sigma(\ell_b^{'}) - d_j, \pi(\ell_b^{'}) - p_j \right\} \tag{35}$$

$$N(\ell_b) = N(\ell_b^{'}) \cup \{j\} \tag{36}$$

$$V(\ell_b) = V(\ell_b^{'}) \backslash \Big(\{j\} \cup \left\{ k \in V(\ell_b^{'}) | \sigma(\ell_b) < d_k \right\} \cup \left\{ k \in V(\ell_b^{'}) | \pi(\ell_b) < p_k \right\} \Big) \tag{37}$$

Since the forward and backward labeling procedures are symmetrical, the dominance test in the backward labeling procedure is conducted in a similar way to that of the forward one, based on the following rule:

Dominance 2. *Let ℓ_b^1 and ℓ_b^2 be the two labels with $\eta(\ell_b^1) = \eta(\ell_b^2) = i$ such that*

$$V(\ell_b^1) \supseteq V(\ell_b^2) \tag{38}$$

$$\bar{c}(\ell_b^1) + \max_{k \in V(\ell_b^2)} \left\{ c_{s_1 k} - c_{s_2 k} \right\} \leq \bar{c}(\ell_b^2) + \sum_{k \in V(\ell_b^2)} \min \left\{ 0, \Lambda_{ks_2} - \Lambda_{ks_1} \right\} \tag{39}$$

$$\sigma(\ell_b^1) \geq \sigma(\ell_b^2) \tag{40}$$

$$\pi(\ell_b^1) \geq \pi(\ell_b^2) \tag{41}$$

where $s_1 = s(\ell_b^1)$ *and* $s_2 = s(\ell_b^2)$ *for the sake of presentation, then the label* ℓ_b^1 *dominates label* ℓ_b^2.

3.4. Label joining

In our implementation, a new resource $\lambda(\ell)$ is defined to represent the number of customers that label ℓ has visited, i.e., $\lambda(\ell) = |N(\ell)|$. An upper bound on the number of customers visited by a feasible route can be computed by solving a two-constraint knapsack problem $\Big\{\bar{\lambda} = \max \sum_{i \in V_C} e_i | \sum_{i \in V_C} d_i e_i \leq Q_2,\ \sum_{i \in V_C} p_i e_i \leq Q_2,\ e_i \in \{0,1\}, \forall i \in V_C\Big\}$. Instead of solving this problem to optimality, we seek for a high-quality bound by separately solving two knapsack problems that consider only one capacity constraint, and taking the minimum of the two optimal values. As the profit from each customer is identical, these two problems can be solved by a polynomial-time algorithm as in Balas and Zemel (1980). The forward and backward label extensions will be terminated when $\lambda(\ell_f) \geq \lceil \bar{\lambda}/2 \rceil$ and $\lambda(\ell_b) \geq \lceil \bar{\lambda}/2 \rceil$.

A forward label ℓ_f ending at vertex i and a backward label ℓ_b ending at vertex j can be joined together to form a complete feasible second-echelon route r if the following conditions hold: $s(\ell_f) = s(\ell_b)$, $V(\ell_f) \supseteq N(\ell_b)$, $V(\ell_b) \supseteq N(\ell_f)$, $\sigma(\ell_f) + \sigma(\ell_b) \geq Q_2$, $\pi(\ell_f) + \pi(\ell_b) \geq Q_2$. The last two conditions impose that both the amounts of consumed resources σ and π cannot exceed the vehicle capacity Q_2, i.e., $\big(Q_2 - \sigma(\ell_f)\big) + \big(Q_2 - \sigma(\ell_f)\big) \leq Q_2$ and $\big(Q_2 - \pi(\ell_f)\big) + \big(Q_2 - \pi(\ell_f)\big) \leq Q_2$. Eventually, the cost of the resulting route will be calculated by $\bar{c}(\ell_f) + \bar{c}(\ell_b) + c_{ij}$.

4. Improving the lower bounds by valid inequalities

This section proposes four families of valid inequalities to lift the lower bounds provided by the LMP based on four aspects: vehicle capacity, satellite capacity, grouping constraints, and simultaneous pickup and delivery. The proposed valid inequalities are separated and added into the LMP using an iterative procedure called the cutting-plane approach, where column generation and the procedure of identifying valid inequalities are iteratively called in turn until no violated inequality can be found. At each cutting-plane iteration, after the LMP is solved by column generation, all families of valid inequalities will be separated in the order of presentation. In the rest of this section, we describe these families of valid inequalities and their separation approaches in detail, followed by how to incorporate them into column generation.

4.1. k-path inequalities on customers

A family of well-known k-path inequalities was introduced by Kohl et al. (1999) for the vehicle routing problem with time windows, which generalizes the capacity inequalities proposed by

Laporte and Nobert (1983). Each of them imposes a minimum number of vehicles leaving from a subset of customers. Let $A^+(N) = \{(i,j) \in A_2 \mid i \in N, j \in (V_S \cup V_C) \backslash N\}$ denote the set of arcs leaving from the customer set $N \subseteq V_C$. The k-path inequalities (KPIs) are represented as follows:

$$\sum_{(i,j)\in A^+(N)} \sum_{r\in R} \beta_{ijr}^2 \theta_r \geq \Gamma(N), \forall N \subseteq V_C \tag{42}$$

where $\Gamma(N)$ is the minimum number of vehicles leaving from the customer set $N \subseteq V_C$. Both the vehicle capacity and time windows were considered by Kohl et al. (1999) to obtain the value of $\Gamma(N)$. For the 2E-VRPGS, a value of $\Gamma(N)$ can be calculated by combining vehicle capacity and grouping constraints. Let $\Gamma^C(N)$ and $\Gamma^G(N)$ be the minimum numbers of vehicles required to serve all customers in the set $N \subseteq V_C$ when considering the vehicle capacity and the grouping constraints, respectively. Then we have $\Gamma(N) = \max\{\Gamma^C(N), \Gamma^G(N)\}$.

First of all, $\Gamma^C(N)$ is set to $\lceil \max\{\sum_{i\in N} d_i, \sum_{i\in N} p_i\}/Q_2 \rceil$. Let $L(N)$ be the index set of groups having at least one customer in customer set $N \subseteq V_C$. Then $\Gamma^G(N)$ can be set to the minimum number of satellites required to serve the customers in $\bigcup_{k\in L(N)} C_k$. To be more specific, we create a virtual item for each group $C_k, k \in L(N)$ and treat satellites as bins. The bin related with the satellite $s \in V_S$ has two independent capacities, which are set to be Q^s. The item related with the group $C_k, k \in L(N)$ has two resource requirements that are equal to delivery demand $\sum_{i\in C_k} d_i$ and pickup demand $\sum_{i\in C_k} p_i$, respectively. To determine the minimum number of bins to pack all the items, the following problem is formulated:

$$\Gamma^G(N) = \min \sum_{s\in V_S} e_s \tag{43}$$

$$s.t. \sum_{k\in L(N)} \sum_{i\in C_k} d_i z_{ks} \leq Q^s e_s, \forall s \in V_S \tag{44}$$

$$\sum_{k\in L(N)} \sum_{i\in C_k} p_i z_{ks} \leq Q^s e_s, \forall s \in V_S \tag{45}$$

$$\sum_{s\in V_S} z_{ks} = 1, \forall k \in L(N) \tag{46}$$

$$z_{ks} \in \{0,1\}, \forall k \in L(N), s \in V_S \tag{47}$$

$$e_s \in \{0,1\}, \forall s \in V_S \tag{48}$$

which is a special two-dimensional case of the vector bin packing with heterogeneous bins introduced by Gabay and Zaourar (2016). Even if the numbers of satellites and groups are small in practical applications, the problem needs to be solved many times within the BCP algorithm. Therefore, we only employ the MIP solver to optimally solve the problem within the first ϖ

times at each cutting-plane iteration, and we set $\Gamma^G(N)$ to 2 at other times when the inequality $\max\left\{\sum_{k\in L(N)}\sum_{i\in C_k} d_i, \sum_{k\in L(N)}\sum_{i\in C_k} p_i\right\} > \max_{s\in V_S}\min\left\{Q^s, |T_s|Q_2\right\}$ holds.

To separate the KPIs, we take advantage of in turn the extended shrinking heuristic and route-based heuristic used by Li et al. (2020b). We refer the reader to it for more details. Only when the extended shrinking heuristic fails to identify any violated KPIs will the route-based one be conducted.

4.2. Rounded capacity inequalities on satellites

Rounded capacity inequalities on satellites (CISs) have been proposed by Liu et al. (2018) and can be generalized by considering simultaneous pickup and delivery. Given a subset of satellites $S \subseteq V_S$, let $D(S)$ denote the minimum amount of cargo to be delivered to satellites in S, and let $P(S)$ represent the minimum amount of cargo to be collected from satellites in S. We have $D(S) = \sum_{i\in V_C} d_i - \sum_{s\in V_S\setminus S}\min\left\{Q^s, |T_s|Q_2\right\}$ and $P(S) = \sum_{i\in V_C} p_i - \sum_{s\in V_S\setminus S}\min\left\{Q^s, |T_s|Q_2\right\}$. As a result, the rounded capacity inequalities on satellites are given by

$$\sum_{(i,j)\in A^+(S)}\sum_{\phi\in\Phi^1}\beta^1_{ij\phi}x_\phi \geq \left\lceil\frac{D(S)}{Q_1}\right\rceil, \forall S\subseteq V_S \tag{49}$$

$$\sum_{(i,j)\in A^+(S)}\sum_{\phi\in\Phi^2}\beta^1_{ij\phi}x_\phi \geq \left\lceil\frac{P(S)}{Q_1}\right\rceil, \forall S\subseteq V_S \tag{50}$$

where $A^+(S) = \{(i,j)\in A_1 \mid i\in S, j\in (V_S\cup\{0\})\setminus S\}$ denotes the set of arcs leaving from the satellite set $S\subseteq V_S$. On account of the small number of groups and satellites, these inequalities are identified by a simple full enumeration.

4.3. Two-dimensional extended cover inequalities

The extended cover inequalities have been developed by Nauss and Robert (2003) for the classical knapsack problem. They defined a cover as a set of items with the total resource requirement exceeding the knapsack capacity. Moreover, a cover is minimal if it will no longer be a cover when one of the items in it is removed. In the 2E-VRPGS, treated as items, the customer groups have two kinds of resource requirements corresponding to the delivery and pickup demands, respectively. Hence, we generalize the classical extended cover inequalities to a family of two-dimensional ones (TDECIs). A group-index subset $L'\subseteq L$ is a two-dimensional cover for the satellite $s\in V_s$ if

$$\max\left\{\sum_{k\in L'}\sum_{i\in C_k} d_i, \sum_{k\in L'}\sum_{i\in C_k} p_i\right\} \geq \min\left\{Q^s, |T_s|Q_2\right\} \tag{51}$$

Furthermore, a two-dimensional cover L' is minimal if the following conditions are satisfied:

$$\max\Big\{\sum_{k'\in L'\backslash\{k\}}\sum_{i\in C_{k'}} d_i, \sum_{k'\in L'\backslash\{k\}}\sum_{i\in C_{k'}} p_i\Big\} < \min\Big\{Q^s, |T_s|Q_2\Big\}, \forall k \in L' \tag{52}$$

The two-dimensional minimal cover inequality for the minimal cover L' and satellite s is given by

$$\sum_{k\in L'} z_{ks} \leq |L'| - 1 \tag{53}$$

Let $ET_1(L')$ and $ET_2(L')$ denote two extended covers of the minimal cover L' defined as

$$ET_1(L') = \begin{cases} \varnothing, & \text{if } \sum_{k\in L'}\sum_{i\in C_k} d_i \leq \min\Big\{Q^s, |T_s|Q_2\Big\} \\ \Big\{k \in L\backslash L' \Big| \sum_{i\in C_k} d_i \geq \sum_{i\in C_{k'}} d_i, \forall k' \in L'\Big\}, & \text{otherwise} \end{cases} \tag{54}$$

$$ET_2(L') = \begin{cases} \varnothing, & \text{if } \sum_{k\in L'}\sum_{i\in C_k} p_i \leq \min\Big\{Q^s, |T_s|Q_2\Big\} \\ \Big\{k \in L\backslash L' \Big| \sum_{i\in C_k} p_i \geq \sum_{i\in C_{k'}} p_i, \forall k' \in L'\Big\}, & \text{otherwise} \end{cases} \tag{55}$$

Then, we have the following lifted two-dimensional extended cover inequality:

$$\sum_{k\in ET(L')} z_{ks} \leq |L'| - 1 \tag{56}$$

where $ET(L') := L' \cup ET_1(L') \cup ET_2(L')$.

The heuristic approach introduced by Kaparis and Letchford (2010) will be used to identify the violated inequalities (56), in which the covers and minimal covers are tested by inequalities (51) and (52), respectively.

4.4. Homogeneous simultaneous pickup-delivery (partial) multistar inequalities

Multistar and partial multistar inequalities were first proposed by Araque et al. (1990) for the capacitated vehicle routing problem with unit demand. Later, Letchford et al. (2002) generalized them to the capacitated vehicle routing problem with general demand. Here, we further extend them to a family of customized valid inequalities for the 2E-VRPGS, referred to as the homogeneous simultaneous pickup-delivery multistar and partial multistar inequalities (PDMIs and PDPMIs).

A partial multistar is a subgraph of G with three non-empty sets of vertices: a set of *nuclei* $M_N \subseteq V_C$, a set of *satellites* $M_S \subseteq V_C\backslash M_N$, and a set of *connectors* $M_C \subseteq M_N$. Note that the '*satellites*' in the context of PDMIs and PDPMIs are different from the physical satellites in the problem description. A multistar can be treated as a special partial multistar with $M_C = M_N$. In other words, a PDMI is essentially a special case of a PDPMI. In what follows, we elaborate on

the PDPMIs only. Let u_l and u_r be two coefficients depending on M_N, M_S, and M_C; then the PDPMIs are given by

$$u_l \sum_{(i,j)\in E(M_C:M_S)} \sum_{r\in R} (\beta_{ijr}^2 + \beta_{jir}^2)\theta_r + \sum_{(i,j)\in E(M_N)} \sum_{r\in R} \beta_{ijr}^2 \theta_r \le u_r,$$
$$\forall M_N \subsetneq V_C, M_S \subseteq V_C \backslash M_N, M_C \subseteq M_N \tag{57}$$

where $E(M_N) = \{(i,j) \in A_2 \mid i \in M_N, j \in M_N, i \neq j\}$ is the set of arcs with both end-vertices in M_N, and $E(M_C : M_S) = \{(i,j) \in A_2 \mid i \in M_C, j \in M_S\}$ represents the set of arcs from M_C to M_S. The sets M_N, M_S and M_C can be generated by a greedy heuristic provided by Letchford et al. (2002). The mathematical expressions of classical partial multistar inequalities and PDPMIs are identical, but they have different sets of valid parameter pairs (u_l, u_r). Given the sets M_N, M_S, and M_C, Letchford et al. (2002) have designed an effective polygon procedure to separate the valid parameter pairs (u_l, u_r) for the classical partial multistar inequalities. Here, we further extend this procedure for PDPMIs based on simultaneous pickup and delivery. Let $\gamma = \sum_{(i,j)\in E(M_C:M_S)} \sum_{r\in R} (\beta_{ijr}^2 + \beta_{jir}^2)\theta_r$ and $\delta = \sum_{(i,j)\in E(M_N)} \sum_{r\in R} \beta_{ijr}^2 \theta_r$ be two auxiliary variables. The polygon procedure consists of the following three steps:

(i) Compute an upper bound UB_γ on variable γ by Lemma 1;

(ii) Given each value of γ in $\{0, 1, ..., UB_\gamma\}$, compute the upper bound $UB_\delta(\gamma)$ on variable δ by Lemmas 2-4;

(iii) In the (γ, δ)-space, construct a polygon bounded by γ axe, δ axe, and the set of coordinates $\{(\gamma, UB_\delta(\gamma))\}_{\gamma\in\{0,1,...,UB_\gamma\}}$. The slope a and intercept b of each boundary line of this polygon lead to a valid pair (u_l, u_r) where $u_l = -a$ and $u_r = b$.

The above lemmas are presented as follows, and their proofs can be found in Appendix B.

Lemma 1. *For any feasible solution to 2E-VPRGS, an upper bound for variable γ can be calculated by*

$$UB_\gamma = \min\left\{2|M_C|, 2|M_S|, |M_C| + |M_S| - \left\lceil \max\left\{ \sum_{i\in M_C\cup M_S} d_i, \sum_{i\in M_C\cup M_S} p_i \right\}/Q_2 \right\rceil \right\} \tag{58}$$

Lemma 2. *For any feasible solution to 2E-VPRGS, we have*

$$\delta \le |M_N| - \left\lceil \frac{\gamma}{2} \right\rceil \tag{59}$$

Lemma 3. *Let $\{v_1, ..., v_{|M_S|}\}$ and $\{v'_1, ..., v'_{|M_S|}\}$ be two ordered lists for vertices in M_S sorted by non-decreasing delivery demand and pickup demand, respectively; then the following inequality*

holds:

$$\delta \leq \begin{cases} |M_N| - \left\lceil \max\left\{ \sum\limits_{i\in M_N} d_i, \sum\limits_{i\in M_N} p_i \right\}/Q_2 \right\rceil, \textit{if} \;\; \gamma = 0 \\ |M_N| - \left\lceil \max\left\{ \sum\limits_{i\in M_N} d_i + \sum\limits_{j=1}^{\gamma} d_{v_j}, \sum\limits_{i\in M_N} p_i + \sum\limits_{j=1}^{\gamma} p_{v'_j} \right\}/Q_2 \right\rceil, \textit{if} \;\; 1 \leq \gamma \leq |M_S| \\ |M_N \cup M_S| - \left\lceil \max\left\{ \sum\limits_{i\in M_N\cup M_S} d_i, \sum\limits_{i\in M_N\cup M_S} p_i \right\}/Q_2 \right\rceil, \textit{if} \;\; \gamma > |M_S| \end{cases} \tag{60}$$

Lemma 4. *Let* $\{v_1, ..., v_{|M_S|}\}$ *and* $\{v'_1, ..., v'_{|M_S|}\}$ *be two ordered lists for vertices in* M_S *sorted by non-decreasing delivery demand and pickup demand, respectively. Analogously, let* $\{o_1, ..., o_{|M_C|}\}$ *and* $\{o'_1, ..., o'_{|M_C|}\}$ *be two ordered lists for vertices in* M_C *sorted by non-decreasing delivery demand and pickup demand, respectively. The following inequality is valid if* $|M_C| \leq \gamma \leq 2|M_C|$*:*

$$\begin{aligned} \delta \leq \;\; & |M_N| - \gamma + |M_C| \\ & - \left\lceil \max\left\{ \sum_{i\in M_N\setminus M_C} d_i + \sum_{j=1}^{2|M_C|-\gamma} (d_{v_j} + d_{o_j}), \sum_{i\in M_N\setminus M_C} p_i + \sum_{j=1}^{2|M_C|-\gamma} (p_{v'_j} + p_{o'_j}) \right\}/Q_2 \right\rceil \end{aligned} \tag{61}$$

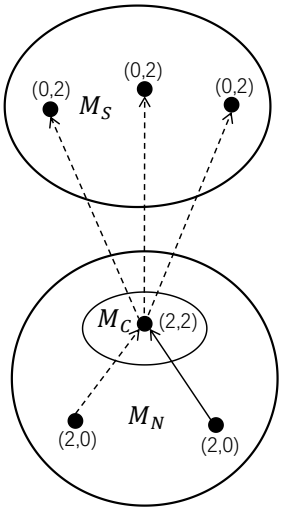

Figure 3. A fractional solution violating PDPMI

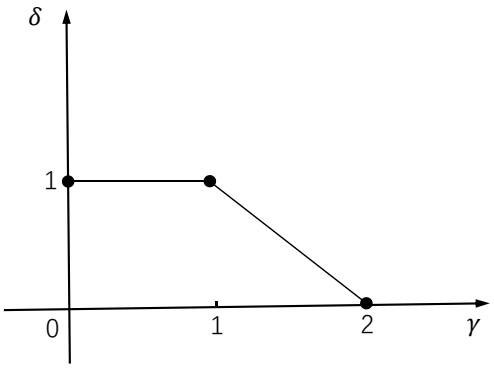

Figure 4. An example of a polyhedron

In terms of these lemmas, for each value of γ in $\{0, 1, ..., UB_\gamma\}$, the corresponding upper bound of δ, i.e., $UB_\delta(\gamma)$, is given by $\min\left\{UB^1_\delta(\gamma), UB^2_\delta(\gamma), UB^3_\delta(\gamma)\right\}$ where $UB^1_\delta(\gamma)$, $UB^2_\delta(\gamma)$, and $UB^3_\delta(\gamma)$ are the right-hand-side values of inequalities (59), (60), and (61), respectively. Figure 3 shows a fractional solution for an instance with $Q_2 = 4$ and a partial multistar with the sets M_N, M_S, and M_C enclosed by ellipses. The solid and dotted arrows indicate the arc variables with values of 1 and 0.5, respectively, and the arcs incident on the vertices outside of this multistar have been omitted. According to Lemma 1, we have $UB_\gamma = 2$. For each value $\gamma \in \{0, 1, 2\}$, we can sequentially calculate $UB_\delta(\gamma) = \{1, 1, 0\}$ based on Lemmas 2-4. Then, the set of coordinates $\{(0, 1), (1, 1), (2, 0)\}$, γ axe, and δ axe constitute a planar polygon illustrated in Figure 4, which has two boundary lines $\delta = 1$

and $\gamma + \delta = 2$. As a result, there are two PDPMIs found, i.e., $\sum_{(i,j)\in E(M_N)} \sum_{r\in R} \beta_{ijr}^2 \theta_r \leq 1$ and $\sum_{(i,j)\in E(M_C:M_S)} \sum_{r\in R} (\beta_{ijr}^2 + \beta_{jir}^2)\theta_r + \sum_{(i,j)\in E(M_N)} \sum_{r\in R} \beta_{ijr}^2 \theta_r \leq 2$, that are violated by the fractional solution in Figure 3.

4.5. Incorporating valid inequalities in column generation

The CISs (49)-(50) and TDECIs (56) will not affect the labeling algorithm, whereas the KPIs (42) and PDPMIs (57) will influence the contributions of arcs to the reduced cost. Let I^{KP} and I^{PM} denote the sets of identified KPIs and PDPMIs, respectively, and $\left\{\xi_t^1 : t \in I^{KP}\right\}$ and $\left\{\xi_t^2 : t \in I^{PM}\right\}$ be the optimal values of the dual variables associated with them. The contribution of arc $(i,j) \in A_2$ to the reduced cost should be updated by $c'_{ij} = c_{ij} - \sum_{t\in I^{KP}} \varphi_{ij}^t \xi_t^1 - \sum_{t\in I^{PM}} \varphi_{ij}^t \xi_t^2$, where φ_{ij}^t denotes the integer coefficient of a variable $y_{ij} = \sum_{r\in R} \beta_{ijr}^2 \theta_r$ associated with arc (i,j) in an inequality $t \in I^{KP} \cup I^{PM}$. The cost c_{ij} of each arc $(i,j) \in A_2$ is replaced by c'_{ij} in pricing problem (19). Since KPIs and PDPMIs do not influence the structure of the pricing problem, the column generation can still work.

5. Branch-and-cut-and-price algorithm

The BCP algorithm is one of the effective solution approaches for many large-scale integer programming problems (e.g., Arslan et al., 2021). In our BCP algorithm, the initial set of second-echelon routes are constructed by a two-phase primal heuristic. Over the course of the branch-and-bound (B&B) framework, the LMP of each B&B node is optimally solved by the column generation in Section 3, and its optimal value is treated as a lower bound. For each node that cannot be pruned, the four families of valid inequalities in Section 4 are dynamically identified and added into the LMP to improve the lower bound. Furthermore, all valid inequalities that have been separated will be retained in the subsequent B&B nodes. To initialize the RLMP at the root node, a set of initial second-echelon routes are generated by a two-phase primal heuristic (detailed in Appendix C). In rest of this section, we describe two techniques for accelerating the labeling algorithm and four branching rules to guide the exploration of the B&B tree, followed by the adjustments to the BCP algorithm for solving the problems 2E-VRPG and 2E-VRPS.

5.1. Accelerating Techniques

This subsection presents two helpful techniques for enhancing the performance of the algorithm.

5.1.1. Heuristic column generator by tabu search

Solving the pricing problem by a labeling algorithm in every iteration could be computationally expensive. Actually, many promising routes with negative reduced costs can be easily found by a heuristic algorithm. Therefore, we will first seek potential routes using a tabu search. If the

heuristic fails to do so, to ensure the exactness of the overall method, the labeling algorithm will then be involved. Specifically, let $N(r)$ and $\bar{N}(r)$ denote the sets of visited customers and unvisited customers for a second-echelon route $r \in R$, respectively. The following three operators will be implemented in the tabu search: (i) *Insertion*: Insert an unvisited customer from $\bar{N}(r)$ into route r; (ii) *Removal*: Remove a visited customer from $N(r)$ out of route r; and (iii) *Swap*: Replace a visited customer in $N(r)$ with an unvisited customer in $\bar{N}(r)$. A neighbor solution can be derived from a given solution using one of the above operators. The heuristic finds the best neighbor solution of the current solution at each iteration and replaces the current solution with the best neighbor. To avoid cycling, a tabu list is also defined to record the arcs removed in the latest iterations. These arcs are simply forbidden from being inserted over 10 iterations.

5.1.2. Ng-route relaxation

This paper takes advantage of an iterative approach provided by Martinelli et al. (2014) to accelerate the labeling algorithm. For each customer $i \in V_C$, an *ng-set* denoted by B_i is a set of the closest customers to customer i in terms of the distance. It is obvious that each customer i belongs to its own *ng-set*, i.e., $i \in B_i$. Given a path $q = (v_1, v_2, ..., v_m)$, we call it an *ng-cycle* if $v_1 = v_m$ and $v_m \in \bigcap_{k=1,...,m-1} B_{v_k}$. A route is an *ng-route* if it does not contain any *ng-cycles*. Ng-route relaxation is to relax the pricing problem by allowing the labeling algorithm to search for *ng-routes* that are essentially non-elementary. With the increase of the size of *ng-sets*, a better lower bound will be found, but more time is needed to solve the pricing problem. If $B_i := \{i\}, i \in V_C$, the pricing problem is equivalent to a non-elementary shortest path problem with resource constraints (SPPRC), while if $B_i = V_C$, the pricing problem is equivalent to the ESPPRC described in Subsection 3.1. For the algorithmic implementation, we will first initialize a small *ng-set* for each customer and call the labeling algorithm to seek an optimal *ng-route*. If the optimal *ng-route* does not contain any *ng-cycles*, it is proved to be elementary, and the iteration stops. Otherwise, we will enlarge the *ng-set* of each customer to forbid more *ng-cycles* and the labeling algorithm is invoked again. This process will be repeated until an optimal elementary route can be found. To incorporate ng-route relaxation, when extending the forward label ℓ'_f to a new forward label ℓ_f along arc (i, j), the component $V(\ell_f)$ will be updated to be

$$V(\ell_f) = \big(V(\ell'_f) \cup \bar{B}_j\big) \backslash \big(\{j\} \cup \{k \in V(\ell'_f) | \sigma(\ell'_f) < d_k\} \cup \{k \in V(\ell'_f) | \pi(\ell'_f) < p_k\}\big) \tag{62}$$

where $\bar{B}_j = V_C \backslash B_j$. Similarly, when extending the backward label ℓ'_b to a new backward label ℓ_b along arc (j, i), the component $V(\ell_b)$ will be updated to be

$$V(\ell_b) = \big(V(\ell'_b) \cup \bar{B}_j\big) \backslash \big(\{j\} \cup \{k \in V(\ell'_f) | \sigma(\ell'_b) < d_k\} \cup \{k \in V(\ell'_f) | \pi(\ell'_b) < p_k\}\big) \tag{63}$$

5.2. Branching rules

Let $\{\hat{x}_\phi : \phi \in \Phi\}$, $\{\hat{\theta}_r : r \in R\}$, $\{\hat{w}^s_\phi : \phi \in \Phi^1, s \in V_S\}$, $\{\hat{g}^s_\phi : \phi \in \Phi^2, s \in V_S\}$, and $\{\hat{z}_{ks} : s \in V_S, k \in L\}$ denote the solution to the strengthened LMP after the last iteration of the cutting procedure. A B&B node will be branched into two new nodes if at least one of the values in $\{\hat{x}_\phi : \phi \in \Phi^1 \cup \Phi^2\}$, $\{\hat{\theta}_r : r \in R\}$, and $\{\hat{z}_{ks} : s \in V_S, k \in L\}$ is fractional. In what follows, we introduce four branching rules in the order of their execution. A new branching rule can be used only if the preceding branching rules fail.

- The first branching rule concerns the values of $\hat{z}_{ks}, \forall s \in V_S, k \in L$. Specifically, given the solution, if no values of $\hat{z}_{ks}$ are fractional, this branching rule fails; otherwise, we look for a satellite s^* and a customer group C_{k^*} such that $\hat{z}_{k^*s^*}$ is closest to 0.5. We then impose two branches: (i) $z_{k^*s^*} \leq \lfloor \hat{z}_{k^*s^*} \rfloor$ and (ii) $z_{k^*s^*} \geq \lceil \hat{z}_{k^*s^*} \rceil$. For the former branch, vehicles departing from satellite s^* are forbidden from visiting any customer in group C_{k^*}, while the later branch forces all the customers in group C_{k^*} to be visited by vehicles from satellite s^*.
- The second branching rule concerns the values of $\hat{x}_\phi, \forall \phi \in \Phi$. Similar to the previous rule, for a given solution, if no values of $\hat{x}_\phi$ are fractional, this branching rule fails; otherwise, we seek for a first-echelon route $\phi^* \in \Phi$ such that $\hat{x}_{\phi^*}$ is closest to 0.5. We impose $x_{\phi^*} \leq \lfloor \hat{x}_{\phi^*} \rfloor$ on one branch and $x_{\phi^*} \geq \lceil \hat{x}_{\phi^*} \rceil$ on the other branch.
- The third branching rule concerns the values of $\hat{n}_s = \sum_{r \in R_s} \hat{\theta}_r, \forall s \in V_S$, i.e., the total number of vehicles used from satellite s in the second echelon. Analogously, if no values of $\hat{n}_s$ are fractional, this branching rule fails; otherwise, we find a satellite $s^* \in V_S$ such that the fractional part of $\hat{n}_{s^*}$ is closest to 0.5. We impose $\sum_{r \in R_{s^*}} \theta_r \leq \lfloor \hat{n}_{s^*} \rfloor$ on one branch and $\sum_{r \in R_{s^*}} \theta_r \geq \lceil \hat{n}_{s^*} \rceil$ on the other branch.
- The fourth branching rule concerns the values of vehicle flows on arcs in A_2. Specifically, given a solution to the LMP, we will calculate the total number of vehicles traversing each arc $(i, j) \in A_2$, i.e., $\hat{y}_{ij} = \sum_{r \in R} \beta^2_{ijr} \hat{\theta}_r$. We search for an arc $(i^*, j^*) \in A_2$ in the second echelon such that $\hat{y}_{i^*j^*} + \hat{y}_{j^*i^*}$ is closest to 0.5, and impose two branches: (i) $y_{i^*j^*} + y_{j^*i^*} \leq \lfloor \hat{y}_{i^*j^*} + \hat{y}_{j^*i^*} \rfloor$ and (ii) $y_{i^*j^*} + y_{j^*i^*} \geq \lceil \hat{y}_{i^*j^*} + \hat{y}_{j^*i^*} \rceil$. For the former branch, the arc (i^*, j^*) will be removed from the graph when solving the pricing problem, while the later branch removes all the arcs in $\{(i^*, j) \in A_2 \mid j \neq j^*\}$ from the graph.

5.3. Adjustments for 2E-VRPG and 2E-VRPS

In the BCP algorithm, the first-echelon routes are explicit in the PB model, whereas the second-echelon routes require being dynamically generated. Neither of the modifications of the PB model for the 2E-VRPG and 2E-VRPS influences the structure of the pricing problem in the second echelon (see Subsection 2.2); therefore the BCP algorithm is still applicable. For the 2E-

VRPG, the modifications in the second echelon are: (i) all the pickup demands are set to zero ($p_i = 0, \forall i \in V_C$), and (ii) the constraints (11) are removed. All the components such as labeling algorithm, all families of valid inequalities, and the branch rules can be kept intact, except that the dual variables associated with constraints (11) are set to zero, i.e., $\tau_i^p = 0, \forall i \in V_C$. As for the 2E-VRPS, all the changes are made in the first echelon except for the elimination of grouping constraints. Therefore, the set Φ^2 will not be applicable and should be replaced by Φ^1 in the inequalities (50), and the inequalities (56) are removed from the BCP algorithm.

6. Numerical experiments

Numerical experiments have been carried out to demonstrate the effectiveness of the proposed algorithm. The algorithm is coded in Java, using Eclipse SDK version 4.2.0 calling ILOG CPLEX 12.6.3 on a personal computer with an Intel i7-9750HF 2.60Ghz CPU, 32GB RAM, and Windows 10 operating system. All the computing times reported are expressed in seconds. A limit of 7200 seconds has been imposed on each run. When separating k-path inequalities on customers, we set parameter $\varpi = 10$ over all of the experiments.

6.1. Benchmark instances

For the numerical experiments, we consider three types of test instances denoted by IT_A, IT_B, and IT_C. The instance IT_A, IT_B, and IT_C are used for the three problems, i.e., 2E-VRPG, 2E-VRPS, and 2E-VRPGS, respectively.

The instances IT_A are taken from Liu et al. (2018) and are generated as follows. Two subtypes of instances are considered in IT_A, i.e., scattered instances and clustered instances. The customers in scattered instances have scattered geographical locations, while the ones in clustered instances have clustered geographical locations. Both scattered and clustered instances are divided into four classes based on the number of satellites, ranging from 2 to 5. Five randomly-generated instances will be included in each class of instances. For ease of differentiation, each instance is named as <s/c>-< $|V_S| - 1$ >-<a/b/c/d/e> where 's' or 'c' in the first entry denotes the scattered and clustered instances, and 'a', 'b', 'c', 'd' or 'e' in the last entry indicates each of the five instances in the same class.

For each instance, the number of customers and the number of disjoint groups are set to be 10 times and 2 times the number of satellites, respectively, i.e., $|V_C| = 10|V_S|$ and $l = 2|V_S|$. The delivery demand of each customer is randomly generated from the interval [1,7]. The capacities of vehicles and the fleet sizes in the first echelon and second echelon are set as follows: $Q_1 = 30$, $Q_2 = 15$, $|K_1^d| = |K_1^p| = \lceil \frac{1.5|V_C|D_{avg}}{Q_1} \rceil$, and $|K_2| = \lceil \frac{1.5|V_C|D_{avg}}{Q_2} \rceil$, where D_{avg} is the average delivery demand of all customers. For each satellite $s \in V_S$, the number of available second-echelon vehicles

$|T_s|$ and the satellite capacity Q^s are set to be $\lceil \frac{|K_2|}{|V_S|} \rceil$ and $\lceil \frac{1.2|V_c|D_{avg}}{|V_S|} \rceil$, respectively. The unit handling cost h_s for each satellite $s \in V_S$ is randomly generated from $[1, 2]$. As for the layout of facilities and customers, a 100 km by 100 km grid with a centrally-located depot is considered, i.e., at (50,50). The locations of satellites depend on the number of satellites in the following ways: (i) if $|V_S| = 2$, the locations of the two satellites are set to $(25, 50)$ and $(75, 50)$, respectively; (ii) if $|V_S| = 3$, the locations of the three satellites are set to $(50, 25)$, $(25, 75)$, and $(75, 75)$, respectively; (iii) if $|V_S| = 4$, the locations of the four satellites are set to $(25, 25)$, $(25, 75)$, $(75, 25)$, and $(75, 75)$, respectively; and (iv) if $|V_S| = 5$, the locations of the five satellites are set to $(25, 25)$, $(25, 75)$, $(50, 50)$, $(75, 25)$, and $(75, 75)$, respectively. For the scattered instances, the locations of all customers are uniformly generated in the square, while for the clustered instances, the locations of every 10 customers are uniformly generated within ([-15, 15],[-15, 15]) with respect to the location of each satellite. Given the location information, the square will be further equally divided into m sub-areas by a horizontal line in the center and $(|V_S| - 1)$ uniformly-distributed vertical lines. The customers in the same sub-area are assigned to the same group.

To test the algorithm on larger-scale instances, we expand instances IT_A up to 10 satellites and 100 customers by using above method except that locations of satellites are randomly generated. Both scattered and clustered instances will be increased to 9 classes based on the number of satellites, ranging from 2 to 10. Each class still has five instances and is named as <S/C><1/2/.../9>, where 'S' and 'C' are used to indicate scattered and clustered instances and the integer values 1-9 represent the parameter $(|V_S| - 1)$. For example, class S1 contains five scattered instances with two satellites, i.e., s-1-a, s-1-b, s-1-c, s-1-d, and s-1-e. The instances IT_C are derived from type IT_A by randomly generating pickup demands of customers from interval $[1, 3]$, whereas the instances IT_B are generated from instances IT_C by removing the customer groups. Therefore, a total of 270 instances are used.

6.2. Results on the 2E-VRPG, 2E-VRPS, and 2E-VRPGS

This subsection summarizes the results obtained by the BCP algorithm on the problems 2E-VRPG, 2E-VRPS, and 2E-VRPGS. The instance-level results are reported in Appendix D. Figure 5 depicts an overview of the percentages of the number of instances that are optimally solved by the BCP algorithm for a specific number of customers (horizontal axis).

Figure 5 shows that the 2E-VRPG is much easier for the proposed BCP algorithm than the 2E-VRPS and 2E-VRPGS as more instances are solved to optimality for 2E-VRPG. More specifically, the BCP algorithm has solved to optimality 48/90, 25/90, and 29/90 instances in total for the problems 2E-VRPG, 2E-VRPS, and 2E-VRPGS to optimality, respectively. This can also be seen from the scale of the problem solved, compared with the 2E-VRPS and 2E-VRPGS, larger-scale

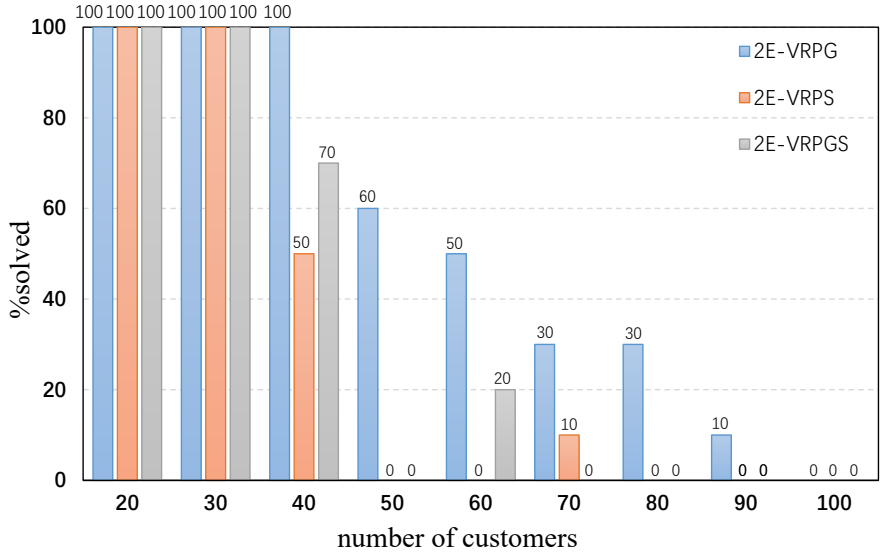

Figure 5. Percentages of the instances solved to optimality on the three problems

Table 2. Summary of lower and upper bounds of BCP algorithm on three problems

	2E-VRPG				2E-VRPS				2E-VRPGS			
Class	$\%gap$	ub	opt	t_{opt}	%gap	ub	opt	t_{opt}	$\%gap$	ub	opt	t_{opt}
S1	4.4	1026.1	5	1.8	2.2	975.7	5	5.7	5.8	1199.1	5	3.3
S2	10.0	1335.5	5	24.9	6.8	1264.1	5	3546.6	12.2	1578.4	5	229.0
S3	8.1	1645.5	5	1034.9	17.6	1837.2	2	4764.1	13.0	2098.3	3	3221.8
S4	13.0	1863.5	2	6110.1	20.8	2046.9	0	-	22.9	2495.5	0	-
S5	14.1	3116.8	2	5817.4	14.5	2865.2	0	-	13.4	3434.4	0	-
S6	4.8	3966.1	2	4527.5	8.2	4053.6	0	-	7.7	4659.9	0	-
S7	3.9	5185.6	1	1233.8	8.1	5354.1	0	-	8.9	5686.6	0	-
S8	4.7	6063.7	0	-	6.4	6224.7	0	-	8.3	6783.6	0	-
S9	5.1	6670.6	0	-	6.3	7230.3	0	-	9.8	8582.4	0	-
avg	7.6	3430.4	-	-	10.1	3539.1	-	-	11.3	4057.6	-	-
C1	2.2	541.6	5	2.4	1.6	589.1	5	3.5	1.5	689.9	5	4.2
C2	10.5	862.8	5	38.5	8.0	884.9	5	353.0	12.6	1102.3	5	402.2
C3	11.3	1411.7	5	566.2	11.2	1397.3	3	1056.3	12.3	1793.2	4	3599.5
C4	10.6	1239.6	4	2224.8	18.8	1441.3	0	-	18.6	1736.3	0	-
C5	16.2	2322	3	4944.1	17.2	2366.8	0	-	13.5	2680.9	2	5770.6
C6	15.2	3680.7	1	1438.9	7.8	3461.3	1	4321.5	9.8	3857.7	0	-
C7	5.6	3818.6	2	4721.2	7.4	4538.1	0	-	8.1	4767.0	0	-
C8	6.1	4699.8	1	6757.6	6.1	5245.2	0	-	8.5	5728.5	0	-
C9	7.1	5871.8	0	-	7.5	6594.3	0	-	10.1	6732.9	0	-
avg	9.4	2716.5	-	-	9.5	2946.5	-	-	10.6	3232.1	-	-

instances on the 2E-VRPG were solved to optimality by the BCP algorithm. The BCP algorithm has optimally solved the instances with up to 90 customers for the 2E-VRPG, while no any instance with more than 70 customers can be solved to optimality for the 2E-VRPS and 2E-VRPGS.

Table 2 reports the lower and upper bounds computed by the BCP algorithm for the three problems. The table summarizes the following average results for each problem and each class of instances: the final upper bound computed within the time limit ('ub'), the percentage gap between the final upper bound and the lower bound computed at the root node ('$\%gap$'), the number of instances solved to optimality (opt), and the computing time to obtain the optimal solution ('t_{opt}'). In the row 'avg' the table also reports the average percentage gap between upper bound and lower bound over all scattered and clustered instances. Table 2 shows that the BCP algorithm computes the tightest lower bounds and the highest-quality upper bounds (evaluated by percentage gaps) on average for the problem 2E-VRPG (see row 'avg'). The average gaps obtained are 7.6% and 9.4% for the scattered and clustered instances, respectively. Looking at class S1, S2, C1, and C2 in which all the instances can be solved to optimality for the three problems, the BCP algorithm takes the shortest average times to optimally deal with them for the 2E-VRPG (see column 't_{opt}'). All in all, the comprehensive comparisons show that the BCP algorithm works best on the 2E-VRPG. Comparing instance-level results in Appendix D, we can see that the optimal value of each instance for the 2E-VRPGS is generally larger than that for the 2E-VRPS. This is because the 2E-VRPS admits simultaneous pickup and delivery in both echelons and ignores grouping constraints, which are necessary for the 2E-VRPGS.

6.3. Comparison with the existing algorithm

To evaluate the effectiveness of the BCP algorithm, this subsection makes a comparison with the algorithm proposed in Liu et al. (2018) for the 2E-VRPG. The results of Liu et al. (2018) were calculated on a PC equipped with a Dell personal computer with an Intel i7-4790 3.60 Ghz CPU, which has a similar performance to the machine used for our experiments. A time limit of 7200 seconds for each run was also imposed in their experiments. In the following, we denote by L18 the algorithm of Liu et al. (2018).

The comparison between the results in Liu et al. (2018) and the results in Tables C1 and C2 shows that all instances solved to optimality by the L18 algorithm can also be optimally solved by the BCP algorithm. It is worth mentioning that the BCP algorithm discovered better solutions on some instances being claimed to be optimally solved by Liu et al. (2018). Table 3 summarizes the exact results obtained by the L18 algorithm and BCP algorithm for eight classes of instances on the 2E-VRPG. For each algorithm, the table reports the number of instances solved to optimality ('opt', for the L18 algorithm the actual numbers solved to optimality are summarized in parentheses) and

Table 3. Summary of the exact results between the L18 and BCP algorithms

				L18		BCP	
Class	ns	ng	nc	opt	t_{opt}	opt	t_{opt}
S1	2	4	20	5(5)	3.6	5	1.8
S2	3	6	30	5(5)	596.3	5	24.9
S3	4	8	40	4(1)	1683.6	5	631.2
S4	5	10	50	0(0)	-	2	-
C1	2	4	20	5(5)	4.2	5	2.4
C2	3	6	30	5(5)	916.2	5	38.5
C3	4	8	40	5(4)	656.0	5	310.1
C4	5	10	50	3(1)	6461.6	4	2574.8

the average computing times over the instances actually solved to optimality by the L18 algorithm ('t_{opt}'). Figure 6 provides an overview of the percentages of the number of instances that are optimally solved by L18 algorithm and BCP algorithm for instances with a different number of customers (horizontal axis).

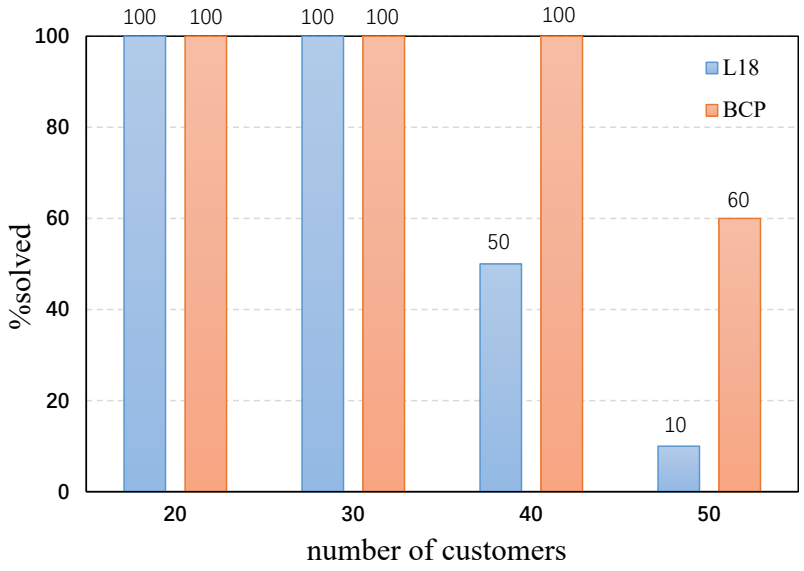

Figure 6. Percentages of instances solved to optimality by the L18 and BCP algorithms

The table shows that 17 (11) scattered instances and 19 (15) clustered instances are solved by the BCP (L18) algorithm within the time limit. The L18 algorithm fails to optimally solve any scattered instance with five satellites and 50 customers, while our BCP algorithm solves both the scattered and clustered instances to optimality with up to five satellites and 50 customers. Compared with L18 algorithm, for each class of instances the BCP algorithm solves the same instances to optimality within significantly less computing time on average and thus it is more competitive (see column 't_{opt}'). Figure 6 clearly shows that the BCP algorithm is capable of solving larger-sized instances than the L18 algorithm. In particular, the BCP algorithm solves all

the instances with 40 customers and 60% of instances with 50 customers to optimality, while the L18 algorithm only optimally solves 50% of instances with 40 customers and 10% of instances with 50 customers.

Table 4. Summary of the lower and upper bounds obtained by L18 and BCP algorithms

	L18				BCP			
Class	lb_{rc}	ub	t_{ub}	$node$	lb_{rc}	ub	t_{ub}	$node$
S1	945.2	1026.1	3.6	400	980.9	1026.1	1.8	31
S2	1151.5	1335.5	596.3	8,479	1202.0	1335.5	24.9	85
S3	1420.1	1671.7	2786.9	12,695	1511.7	1645.5	982.9	2,679
S4	1504.2	1848.3	7200.0	8,337	1621.6	1863.5	6764.0	7,188
C1	523.2	541.6	4.2	858	529.5	541.6	2.4	39
C2	736.8	862.8	916.2	7,328	772.3	862.8	38.5	87
C3	1157.9	1422.9	1079.1	4,001	1251.5	1411.7	566.2	285
C4	1050.6	1255.0	5027.0	10,926	1107.8	1239.6	3219.8	6,317

Table 4 summarizes the lower and upper bounds obtained by the L18 and BCP algorithms for eight classes of instances. For each algorithm and each class of instances, the table reports the following average results: the final lower bounds at the root node ('lb_{rc}'), the final upper bounds obtained within the time limit ('ub'), the total computing times for obtaining the upper bounds ('t_{ub}'), and the total number of B&B nodes generated. Compared with the L18 algorithm, the BCP algorithm obtained higher-quality lower bounds and enumerated fewer B&B nodes on average. This is within our prior expectation, since the PB model in this paper is tighter than the AFB model proposed in Liu et al. (2018). For each class of instances, on average the BCP algorithm attains the better upper bounds within much less computing time than the L18 algorithm (see columns 'ub' and 't_{ub}').

In summary, the BCP algorithm in this paper is considerably more efficient than the L18 algorithm on the problem 2E-VRPG.

6.4. Algorithm analysis

This subsection evaluates the performances of proposed valid inequalities and dominance rule. Table 5 summarizes the performance of valid inequalities on eight classes of instances. We computed the following average results of solving LP-relaxation of PB model over different classes of instances: the lower bound obtained after adding all cuts ('ALL'), the lower bound without adding any cut ('NO'), and the lower bounds after adding all cuts except for one family of the following valid inequalities: k-path inequalities on customers ('KPIs'), rounded capacity inequalities on satellites ('CISs'), two-dimensional extended cover inequalities ('TDECIs'), and homogeneous simultaneous pickup-delivery partial multistar inequalities ('PDPMIs'). Besides, we also report the average lower

bounds ('avg') and the average percentage reductions with respect to the lower bounds after adding all cuts ('%gap') over all scattered and clustered instances.

Table 5. Summary of the performance of valid inequalities

Class	NO	KPIs	CISs	TDECIs	PDPMIs	ALL
S1	1019.0	1096.1	1096.2	1084.7	1116.8	1130.1
S2	1300.6	1360.5	1379.4	1344.6	1373.0	1385.0
S3	1702.5	1797.9	1800.7	1770.0	1796.5	1826.3
S4	1818.1	1886.3	1920.0	1868.7	1912.7	1923.5
avg	1460.1	1535.2	1549.1	1517.0	1549.8	1566.2
%gap	-6.8	-2.0	-1.1	-3.1	-1.1	-
C1	639.9	660.6	659.6	679.6	671.5	679.6
C2	936.0	945.5	962.7	956.3	958.8	963.3
C3	1417.4	1546.4	1566.7	1454.0	1545.8	1572.9
C4	1302.2	1382.1	1389.9	1362.3	1381.6	1414.2
avg	1073.9	1133.6	1144.7	1113.0	1139.4	1157.5
%gap	-7.2	-2.1	-1.1	-3.8	-1.6	-

As can be seen from the average results in Table 5, all the families of valid inequalities have a positive impact on the lower bounds and their synthesis lifts lower bound by 6.8% and 7.2% on average for scattered and clustered instances, respectively. Specifically, the TDECIs, followed by KPIs, offer the most significant improvement on both scattered and clustered instances. If this family of inequalities is missing, the lower bound will reduce by approximately 3.1% and 3.8% on average for scattered and clustered instances, respectively. Although the CISs and PDPMIs have smaller impacts than the other two families, the lower bound will reduce by more than 1% on average if either of them is missing for both scattered and clustered instances.

For assessing the proposed dominance rule, we use column generation called CGV (CGNV) to solve LP-relaxation of PB model without adding any cut, in which the proposed dominance is verified (or not verified) between labels from different satellites. Table 6 summarizes the performance of alternative column generations with different dominance rules on eight classes of instances. For each column generation, we computed the following average results over the different classes of instances: the time spent in solving LP-relaxation ('t_r'), the number of forward labels generated ('fw'), and the number of backward labels generated ('bw'). Finally, the average results ('avg') over all scattered and clustered instances are also reported. Columns 'fw' and 'bw' show that over the scattered and clustered instances the number of labels generated by CGV is greatly less than that of CGNV. Furthermore, column 't_r' shows that the CGV is faster than the CGNV on both scattered (4.5 vs. 9.8 on average) and clustered instances (4.4 vs. 9.3 on average), which indicates that the dominance being verified between labels from different satellites can effectively accelerate the algorithm. This is because that the stronger dominance rule will contribute to a

Table 6. Summary of the performance of alternative column generations

	CGV			CGNV		
Class	t_r	fw	bw	t_r	fw	bw
S1	0.2	2,451	1,258	0.3	3,996	2,302
S2	0.5	14,977	5,217	0.9	32,210	18,720
S3	1.1	49,919	12,468	2.5	126,346	69,100
S4	2.8	113,335	28,353	6.1	378,826	218,276
avg	4.5	180,683	47,297	9.8	541,378	308,398
C1	0.2	2,857	1,077	0.3	7,639	5,117
C2	0.5	13,449	3,991	1.1	42,484	25,313
C3	1.0	35,985	8,312	2.1	99,648	63,047
C4	2.7	91,428	16,823	5.8	322,798	237,323
avg	4.4	143,719	30,204	9.3	472,569	330,801

smaller number of generated labels and thus improve computational efficiency.

6.5. Sensitivity analysis

The PB model aims at minimizing the total operating cost, including traveling costs of vehicles and handling costs at satellites. As described in Section 1, grouping constraints will bring many management benefits to the new system but will also largely influence the routing plans and incur additional operating cost. As PB model treats grouping constraints as hard constraints, benefits of grouping constraints on system have to be considered implicitly. This subsection analyses the negative impact of grouping constraints on operating cost, which is sensitive to customer distribution pattern, number of selected groups, and satellite capacity.

Considering a specific instance, we analyse the change of routes and cost in the solution under different scenarios featured by the number of groups and satellite capacity $< \bar{l}, \bar{Q}^s >$, where $\bar{l}$ and $\bar{Q}^s$ represent the number of selected groups and specified satellite capacity, respectively. Under each scenario $< \bar{l}, \bar{Q}^s >$, we randomly choose $\bar{l}$ groups from the original groups of the instance and relax the grouping constraints on the other groups.

Considering s-1-b and c-1-b with different customer distribution patterns as example, we first analyse 5 scenarios with $\bar{l} = 0, 1, 2, 3, 4$ and $\bar{Q}^s = 48$. Figure 7 shows the variations of operating costs with the increase of $\bar{l}$. It can be seen that with the increase of $\bar{l}$, the operating cost becomes higher. This is because the increased grouping constraints will make some more cost-effective routes infeasible in terms of grouping constraints. To illustrate the difference of the operating costs under different numbers of groups, Figures 8 and 9 present the detailed routing plans of the optimal solutions under two scenarios $< 0, 48 >$ and $< 4, 48 >$ in instance s-1-b. From these figures, we can see that customers in the routing plan of the scenario $< 0, 48 >$ are always served by vehicles from the satellites nearby, while customers in the routing plan of the scenario $< 4, 48 >$ may have to be

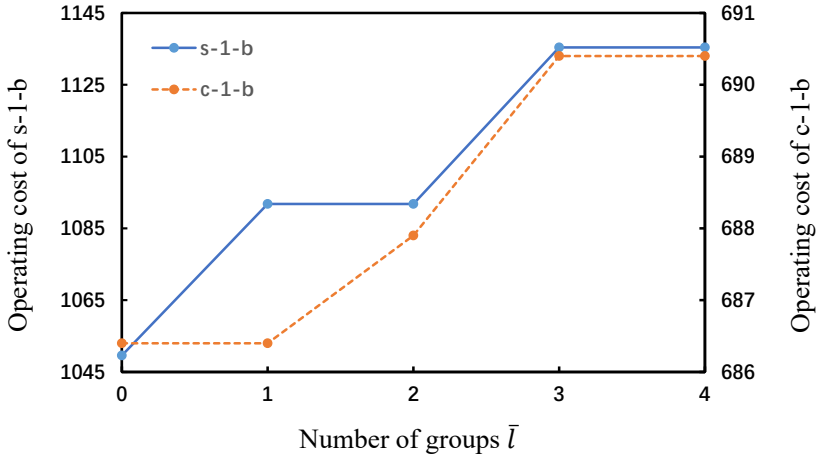

Figure 7. Variations of operating costs of s-1-b and c-1-b with the increase of $\bar{l}$

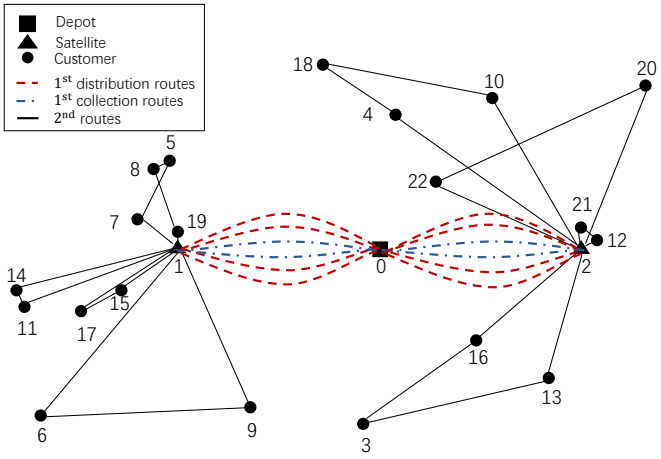

Figure 8. The routing plan of the optimal solution of instance s-1-b under scenario $< 0, 48 >$

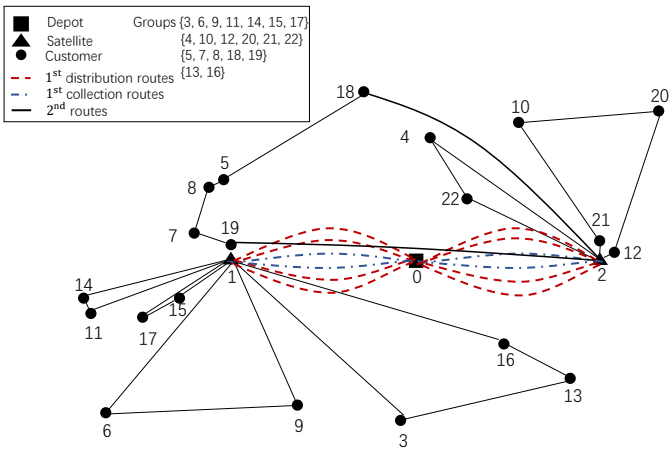

Figure 9. The routing plan of the optimal solution of instance s-1-b under scenario $< 4, 48 >$

served by vehicles from the satellites far from them due to the four grouping constraints added, resulting in long vehicle routes and high traveling cost.

It is interesting to find that the increment rate of the operating cost on instance s-1-b is much larger than that of instance c-1-b. For example, as $\bar{l}$ rises from 0 to 4, the operating cost of c-1-b goes up by 4 (increment percentage is 0.6%), whereas that of instance s-1-b increases by 85.8 (increment percentage is 8.2%). This can be explained by the fact that, for instance c-1-b most customers in the same group uniformly managed by an old satellite have already clustered beside the newly-established satellites. Hence, the increase of $\bar{l}$ has a small impact on instance c-1-b. The result suggests that logistics companies had better establish satellites near the clustered centers of customers as far as possible, so as to control the operating cost incurred by grouping management when expanding the transportation business.

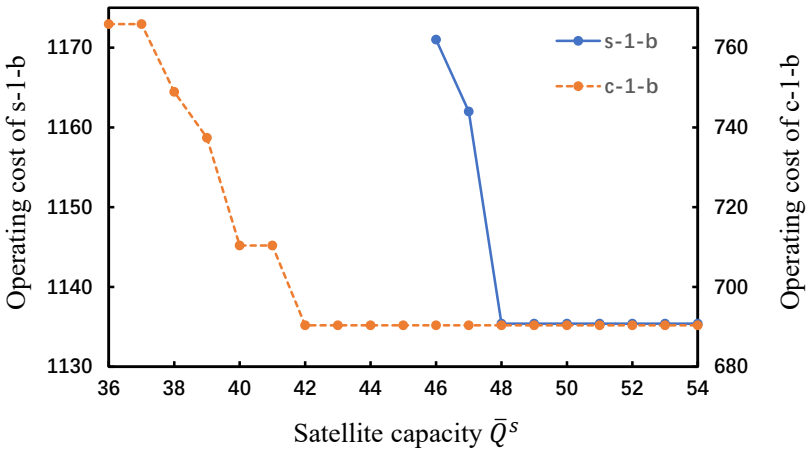

Figure 10. Variations of operating costs of s-1-b and c-1-b with the increase of $\bar{Q}^s$

Furthermore, we also analyse 19 scenarios with $\bar{l} = 3$ and $\bar{Q}^s = 36, 37, 38, \ldots, 54$. The variations of operating costs against $\bar{Q}^s$ are visualized in Figure 10. It can be seen that the operating cost keeps decreasing with the increase of $\bar{Q}^s$, when $\bar{Q}^s$ is not larger than 48 and 42 for scattered and clustered instance, respectively. This is within our expectation as large capacitated satellites allow us to allocate customers to the most cost-saving satellites. Since there are no feasible solution to s-1-b under the scenarios with $\bar{Q}^s = 36, \ldots, 44$, the curve doesn't depict the results in that range. In addition, both the operating costs for s-1-b and c-1-b are stabilized to a certain value when the satellite capacity exceeds a critical point, i.e., 48 for s-1-b and 42 for c-1-b. This implies that the capacities of satellites will no longer be a binding constraint that restricts the optimal customer assignment to satellites and vehicle routes. As a matter of fact, larger satellite capacity will induce a higher establishment cost of satellites for logistics companies. The logistics companies are not suggested to blindly increase the satellite capacity aiming for operating cost reduction, especially when it has reached a critical value.

7. Conclusions

This study is the first one to introduce a new variant of the two-echelon vehicle routing problem considering both the grouping constraints and simultaneous pickup and delivery (2E-VRPGS). More specifically, customers are divided into different groups based on administrative areas, and customers from the same group can only be served by second-echelon vehicles from the same satellite. The objective is to minimize the total operating cost by determining the assignment of customer groups to satellites and the optimal routes of vehicles in the two echelons.

In this paper, a path-based (PB) model is proposed to formulate the 2E-VRPGS. Modifications are also presented to model the 2E-VRPG and the 2E-VRPS introduced by other researchers. To solve the three problems to optimality, a branch-and-cut-and-price (BCP) algorithm is developed, where both a novel dominance rule in the labeling algorithm and several families of valid inequalities are proposed. To the best of our knowledge, this paper is the first attempt at developing an exact algorithm to solve both the 2E-VRPS and 2EVRPGS.

A number of numerical experiments are conducted on three types of instances. This paper reports the final results of the experiments for the problems 2E-VRPG, 2E-VRPS, and 2E-VRPGS, and analyzes the performance of the algorithm for solving them. Subsequently, the impacts of proposed valid inequalities and dominance rule on BCP algorithm are systematically evaluated. Computational results show that all the valid inequalities have positive effects on strengthening the PB model and our dominance rule can significantly reduce the number of generated labels. Comparisons between our algorithm and the branch-and-cut algorithm in Liu et al. (2018) illustrate that our algorithm is more effective than the existing algorithm. Some new findings and managerial insights are provided by conducting sensitivity analysis on some sensitive factors.

The proposed exact algorithm can only solve small- and medium-sized instances in a reasonable computing time. Further work could focus on the development of effective and efficient meta-heuristic algorithms that can solve large-sized instances of the proposed problem. In addition, it makes great sense to study some extensions of this problem in the future that are derived from relaxing some existing restrictive assumptions or adding some practical constraints. For example, we can consider that an unlimited fleet of vehicles are used, multiple types of vehicles are employed, and time windows of customers. Last but not least, extending the models and solution method to many other similar transportation and logistics systems, such as the emerging two-level crowd-shipping services, would be a promising direction for future research.

Acknowledgement

The work described in this paper was supported by a grant from National Natural Science Foundation of China (No. 71901189) and a grant from the Research Grants Council of the Hong Kong Special Administrative Region, China (Project No. PolyU 15210620). This research was also partially supported by National Natural Science Foundation of China (Grant nos. 71971090, 71821001, 7210010522). The authors would like to thank the anonymous reviewers and associate editor for their helpful suggestions and very thorough review of the paper. The authors also thank Tian Liu for providing the instances and the corresponding details used in Liu et al. (2018).

References

Araque, J. R., Hall, L. A., and Magnanti, T. L. (1990). *Capacitated trees, capacitated routing, and associated polyhedra*. Massachusetts Institute of Technology, Operations Research Center.

Arslan, O., Kumcu, G. Ç., Kara, B. Y., and Laporte, G. (2021). The location and location-routing problem for the refugee camp network design. *Transportation Research Part B: Methodological*, 143:201–220.

Avci, M. and Topaloglu, S. (2015). An adaptive local search algorithm for vehicle routing problem with simultaneous and mixed pickups and deliveries. *Computers & Industrial Engineering*, 83:15–29.

Balas, E. and Zemel, E. (1980). An algorithm for large zero-one knapsack problems. *Operations Research*, 28(5):1130–1154.

Baldacci, R., Mingozzi, A., Roberti, R., and Calvo, R. W. (2013). An exact algorithm for the two-echelon capacitated vehicle routing problem. *Operations Research*, 61(2):298–314.

Belgin, O., Karaoglan, I., and Altiparmak, F. (2018). Two-echelon vehicle routing problem with simultaneous pickup and delivery: Mathematical model and heuristic approach. *Computers & Industrial Engineering*, 115:1–16.

Breunig, U., Baldacci, R., Hartl, R. F., and Vidal, T. (2019). The electric two-echelon vehicle routing problem. *Computers & Operations Research*, 103:198–210.

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

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

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

Dantzig, G. B. and Ramser, J. H. (1959). The truck dispatching problem. *Management Science*, 6(1):80–91.

Dellaert, N., Dashty Saridarq, F., Van Woensel, T., and Crainic, T. G. (2019). Branch-and-price–based algorithms for the two-echelon vehicle routing problem with time windows. *Transportation Science*, 53(2):463–479.

Desaulniers, G., Desrosiers, J., and Solomon, M. M. (2006). *Column Generation*, volume 5. Springer Science & Business Media.

Dror, M. (1994). Note on the complexity of the shortest path models for column generation in vrptw. *Operations Research*, 42(5):977–978.

Fazi, S., Fransoo, J. C., Van Woensel, T., and Dong, J.-X. (2020). A variant of the split vehicle routing problem with simultaneous deliveries and pickups for inland container shipping in dry-port based systems. *Transportation Research Part E: Logistics and Transportation Review*, 142:102057.

Franceschetti, A., Honhon, D., Laporte, G., Van Woensel, T., and Fransoo, J. C. (2017). Strategic fleet planning for city logistics. *Transportation Research Part B: Methodological*, 95:19–40.

Gabay, M. and Zaourar, S. (2016). Vector bin packing with heterogeneous bins: application to the machine reassignment problem. *Annals of Operations Research*, 242(1):161–194.

Gonzalez-Feliu, J. (2008). *Models and methods for the city logistics: The two-echelon capacitated vehicle routing problem*. PhD thesis, Politecnico di Torino.

Groër, C., Golden, B., and Wasil, E. (2009). The consistent vehicle routing problem. *Manufacturing & Service Operations Management*, 11(4):630–643.

Hu, Z.-H., Sheu, J.-B., Zhao, L., and Lu, C.-C. (2015). A dynamic closed-loop vehicle routing problem with uncertainty and incompatible goods. *Transportation Research Part C: Emerging Technologies*, 55:273–297.

Jepsen, M., Spoorendonk, S., and Ropke, S. (2013). A branch-and-cut algorithm for the symmetric two-echelon

capacitated vehicle routing problem. *Transportation Science*, 47(1):23–37.

Kaparis, K. and Letchford, A. N. (2010). Separation algorithms for 0-1 knapsack polytopes. *Mathematical Programming*, 124(s1-2):69–91.

Kergosien, Y., Lenté, C., Billaut, J.-C., and Perrin, S. (2013). Metaheuristic algorithms for solving two interconnected vehicle routing problems in a hospital complex. *Computers & Operations Research*, 40(10):2508–2518.

Koç, Ç., Laporte, G., and Tükenmez, İ. (2020). A review on vehicle routing with simultaneous pickup and delivery. *Computers & Operations Research*, page 104987.

Kohl, N., Desrosiers, J., Madsen, O. B., Solomon, M. M., and Soumis, F. (1999). 2-path cuts for the vehicle routing problem with time windows. *Transportation Science*, 33(1):101–116.

Laporte, G. and Nobert, Y. (1983). A branch and bound algorithm for the capacitated vehicle routing problem. *Operations-Research-Spektrum*, 5(2):77–85.

Letchford, A. N., Eglese, R. W., and Lysgaard, J. (2002). Multistars, partial multistars and the capacitated vehicle routing problem. *Mathematical Programming*, 94(1):21–40.

Li, H., Liu, Y., Jian, X., and Lu, Y. (2018). The two-echelon distribution system considering the real-time transshipment capacity varying. *Transportation Research Part B: Methodological*, 110:239–260.

Li, H., Wang, H., Chen, J., and Bai, M. (2020a). Two-echelon vehicle routing problem with time windows and mobile satellites. *Transportation Research Part B: Methodological*, 138:179–201.

Li, H., Zhang, L., Lv, T., and Chang, X. (2016). The two-echelon time-constrained vehicle routing problem in linehaul-delivery systems. *Transportation Research Part B: Methodological*, 94:169–188.

Li, J., Qin, H., Baldacci, R., and Zhu, W. (2020b). Branch-and-price-and-cut for the synchronized vehicle routing problem with split delivery, proportional service time and multiple time windows. *Transportation Research Part E: Logistics and Transportation Review*, 140:101955.

Liu, T., Luo, Z., Qin, H., and 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.

Marques, G., Sadykov, R., Deschamps, J.-C., and Dupas, R. (2020). An improved branch-cut-and-price algorithm for the two-echelon capacitated vehicle routing problem. *Computers & Operations Research*, 114:104833.

Marques, G., Sadykov, R., Dupas, R., and Deschamps, J.-C. (2022). A branch-cut-and-price approach for the single-trip and multi-trip two-echelon vehicle routing problem with time windows. *Transportation Science*, ISSN 1526-5447 (online).

Martinelli, R., Pecin, D., and Poggi, M. (2014). Efficient elementary and restricted non-elementary route pricing. *European Journal of Operational Research*, 239(1):102–111.

Mhamedi, T., Andersson, H., Cherkesly, M., and Desaulniers, G. (2022). A branch-price-and-cut algorithm for the two-echelon vehicle routing problem with time windows. *Transportation Science*, 56(1):245–264.

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

Nadizadeh, A. and Kafash, B. (2019). Fuzzy capacitated location-routing problem with simultaneous pickup and delivery demands. *Transportation Letters*, 11(1):1–19.

Nauss and Robert, M. (2003). Solving the generalized assignment problem: An optimizing and heuristic approach. *Informs Journal on Computing*, 15(3):249–266.

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

Qu, Y. and Bard, J. F. (2013). The heterogeneous pickup and delivery problem with configurable vehicle capacity. *Transportation Research Part C: Emerging Technologies*, 32:1–20.

Qu, Y. and Bard, J. F. (2015). A branch-and-price-and-cut algorithm for heterogeneous pickup and delivery problems with configurable vehicle capacity. *Transportation Science*, 49(2):254–270.

Santos, F. A., da Cunha, A. S., and Mateus, G. R. (2013). Branch-and-price algorithms for the two-echelon capacitated vehicle routing problem. *Optimization Letters*, 7(7):1537–1547.

Santos, F. A., Mateus, G. R., and da Cunha, A. S. (2015). A branch-and-cut-and-price algorithm for the two-echelon capacitated vehicle routing problem. *Transportation Science*, 49(2):355–368.

Subramanian, A., Uchoa, E., Pessoa, A. A., and Ochi, L. S. (2011). Branch-and-cut with lazy separation for the vehicle routing problem with simultaneous pickup and delivery. *Operations Research Letters*, 39(5):338–341.

Subramanian, A., Uchoa, E., Pessoa, A. A., and Ochi, L. S. (2013). Branch-cut-and-price for the vehicle routing problem with simultaneous pickup and delivery. *Optimization Letters*, 7(7):1569–1581.

Toth, P. and Vigo, D. (2014). *Vehicle routing: problems, methods, and applications*, volume 18. SIAM.

Wang, Y., Ma, X., Lao, Y., Wang, Y., and Mao, H. (2013). Vehicle routing problem: simultaneous deliveries and pickups with split loads and time windows. *Transportation Research Record*, 2378(1):120–128.

Yao, Y., Woensel, T. V., Veelenturf, L. P., and Mo, P. (2021). The consistent vehicle routing problem considering path consistency in a road network. *Transportation Research Part B: Methodological*, 153.

Zeng, Z., Xu, W., Xu, Z., and Shao, W. (2014). A hybrid GRASP + VND heuristic for the two-echelon vehicle routing problem arising in city logistics. *Mathematical Problems in Engineering*, 2014.

Zhou, H., Qin, H., Zhang, Z., and Li, J. (2022). Two-echelon vehicle routing problem with time windows and simultaneous pickup and delivery. *Soft Computing*, pages 1–16.

A. Notation

Table A1. Notation for the definition of the 2E-VRPGS

Notation	Descriptions
0	depot
V_C	set of customers
V_S	set of satellites
V	set of vertices defined as $\{0\} \cup V_C \cup V_S$
Q^s	capacity of the satellite $s \in V_S$
h_s	unit handling cost at the satellite $s \in V_S$
$d_i(p_i)$	delivery (pickup) demand of the customer $i \in V_C$
l	number of customer groups
L	index set of customer groups
A_1	set of the 1^{st}-echelon arcs
A_2	set of the 2^{nd}-echelon arcs
A	set of arcs defined as $A = A_1 \cup A_2$
c_{ij}	traveling cost of arc $(i, j) \in A$
$Q_1(Q_2)$	capacity of the 1^{st}-echelon (2^{nd}-echelon) vehicles
$K_1^d(K_1^p)$	set of the available 1^{st}-echelon vehicles for distribustion (collection)
K_2	set of the available 2^{nd}-echelon vehicles
T_s	set of the available vehicles departing from satellite $s \in V_S$
G	directed graph defined as $G = (V, A)$
Φ	set of the 1^{st}-echelon routes
Φ^1	set of the 1^{st}-echelon distribution routes
Φ^2	set of the 1^{st}-echelon collection routes
Φ_S	set of 1^{st}-echelon routes in Φ visiting at least one satellite in $S \subseteq V_S$
Φ_S^1	set of 1^{st}-echelon distribution routes in Φ_S
Φ_S^2	set of 1^{st}-echelon collection routes in Φ_S
S_Φ	set of satellites visited by a 1^{st}-echelon route $\phi \in \Phi$
c_ϕ	cost of the 1^{st}-echelon route $\phi \in \Phi$
$\beta_{ij\phi}^1$	number of times that arc $(i, j) \in A_1$ is traversed by the 1^{st}-echelon route $\phi \in \Phi$
R	set of the 2^{nd}-echelon routes
R_s	set of the 2^{nd}-echelon routes departing from satellite $s \in V_S$
α_{ir}	number of times that customer $i \in V_C$ is visited by the 2^{nd}-echelon route $r \in R$
c_r	cost of the 2^{nd}-echelon route $r \in R$
$\beta^2 ijr$	number of times that arc $(i, j) \in A_2$ is traversed by the 2^{nd}-echelon route $r \in R$
$S_\phi^a(s)$	set of satellites visited by the 1^{st}-echelon route $\phi \in \Phi$ after visiting the satellite $s \in V_S$
$S_\phi^b(s)$	set of satellites visited by the 1^{st}-echelon route $\phi \in \Phi$ before visiting the satellite $s \in V_S$
x_ϕ	binary variable indicating whether the 1^{st}-echelon route $\phi \in \Phi$ is selected
θ_r	binary variable indicating whether the 2^{nd}-echelon route $r \in R$ is selected
w_ϕ^s	decision variable indicating the amount distributed to satellite s by route $\phi \in \Phi^1$
g_ϕ^s	decision variable indicating the amount collected from satellite s by route $\phi \in \Phi^2$
z_{ks}	binary variable indicating whether the group $C_k, k \in L$ are assigned to satellite $s \in V_s$

B. Proofs

B.1. Proof of Dominance 1

Let q_f^1 and q_f^2 represent the forward partial paths for the labels ℓ_f^1 and ℓ_f^2, respectively. Given a forward (backward) partial path $q = (v_1, \ldots, v_{|q|})$ where $|q|$ is the number of vertices visited and v_t is the t^{th} vertex, let $f_t(q)$ be the load of vehicle after (before) visiting the vertex v_t, and let $\pi(q)$, $\sigma(q)$, and $N(q)$ have the same meanings as the resources π, σ, and N in the label definition, respectively.

In Dominance 1, we have $\eta(\ell_f^1) = \eta(\ell_f^2) = i$, $s(\ell_f^1) = s_1$, and $s(\ell_f^2) = s_2$. Given a backward partial path $q_b^2 \in \mathscr{P}(q_f^2)$, let j and h be the first and last customers in path q_b^2, respectively. A new backward partial path q_b^1 can be constructed from q_b^2 by replacing its satellite s_2 by s_1. Therefore, we have $\pi(q_b^1) = \pi(q_b^2)$, $\sigma(q_b^1) = \sigma(q_b^2)$, and $N(q_b^1) = N(q_b^2)$.

Firstly, we prove that $q_b^1 \in \mathscr{P}(q_f^1)$. Because $q_b^2 \in \mathscr{P}(q_f^2)$, we have

$$f_{|q_b^2|}(q_b^2) + \max_{t=1}^{|q_f^2|} \left\{ f_t(q_f^2) \right\} \leq Q_2 \tag{64}$$

$$f_{|q_f^2|}(q_f^2) + \max_{t=1}^{|q_b^2|} \left\{ f_t(q_b^2) \right\} \leq Q_2 \tag{65}$$

where $f_{|q_b^2|}(q_b^2)$ is actually the amount of cargo that should be delivered to the customers in q_b^2 and $f_{|q_f^2|}(q_f^2)$ is actually the amount of cargo that should be collected from the customers in q_f^2. We have

$$\pi(\ell_f^2) = \pi(q_f^2) = Q_2 - f_{|q_f^2|}(q_f^2),\ \sigma(\ell_f^2) = \sigma(q_f^2) = Q_2 - \max_{t=1}^{|q_f^2|} \left\{ f_t(q_f^2) \right\}$$

$$\pi(q_b^2) = Q_2 - \max_{t=1}^{|q_b^2|} \left\{ f_t(q_b^2) \right\},\ \sigma(q_b^2) = Q_2 - f_{|q_b^2|}(q_b^2)$$

Combining the above equalities and inequalities (64)-(65), we can obtain the following conditions of the relationship $q_b^2 \in \mathscr{P}(q_f^2)$:

$$N(q_b^2) \subseteq V(\ell_f^2), \pi(\ell_f^2) + \pi(q_b^2) \geq Q_2, \sigma(\ell_f^2) + \sigma(q_b^2) \geq Q_2$$

In terms of conditions (27),(29), and (30), we also have

$$N(q_b^1) \subseteq V(\ell_f^1), \pi(\ell_f^1) + \pi(q_b^1) \geq Q_2, \sigma(\ell_f^1) + \sigma(q_b^1) \geq Q_2$$

and therefore $q_b^1 \in \mathscr{P}(q_f^1)$.

Next, we prove that $\bar{c}(q_f^2 \oplus q_b^2)$ is not less than $\bar{c}(q_f^1 \oplus q_b^1)$. The difference between $\bar{c}(q_f^1 \oplus q_b^1)$ and $\bar{c}(q_f^2 \oplus q_b^2)$ is calculated by

$$\begin{aligned}
&\bar{c}(q_f^2 \oplus q_b^2) - \bar{c}(q_f^1 \oplus q_b^1) \\
&= \bar{c}(\ell_f^2) + c_{ij} + \bar{c}(q_b^2) - \left(\bar{c}(\ell_f^1) + c_{ij} + \bar{c}(q_b^1)\right) \\
&= \bar{c}(\ell_f^2) - \bar{c}(\ell_f^1) + \bar{c}(q_b^2) - \bar{c}(q_b^1)
\end{aligned}$$

$$
\begin{aligned}
&= \bar{c}(\ell_f^2) - \bar{c}(\ell_f^1) + \sum_{k \in N(q_b^2)} \left(\Lambda_{ks_2} - \Lambda_{ks_1}\right) + c_{hs_2} - c_{hs_1} \\
&\geq \max_{k \in V(\ell_f^2)} \left\{c_{ks_1} - c_{ks_2}\right\} - \sum_{k \in V(\ell_f^2)} \min\left\{0, \Lambda_{ks_2} - \Lambda_{ks_1}\right\} + \sum_{k \in N(q_b^2)} \left(\Lambda_{ks_2} - \Lambda_{ks_1}\right) + c_{hs_2} - c_{hs_1} \\
&\geq \max_{k \in V(\ell_f^2)} \left\{c_{ks_1} - c_{ks_2}\right\} + \sum_{k \in N(q_b^2)} \left(\Lambda_{ks_2} - \Lambda_{ks_1} - \min\left\{0, \Lambda_{ks_2} - \Lambda_{ks_1}\right\}\right) + c_{hs_2} - c_{hs_1} \\
&\geq \max_{k \in V(\ell_f^2)} \left\{c_{ks_1} - c_{ks_2}\right\} + c_{hs_2} - c_{hs_1} \\
&\geq 0
\end{aligned}
$$

and the first symbol "$\geq$" holds according to inequality (28).

B.2. Proof of Lemma 1

Firstly, $\sum_{(i,j) \in E(M_C:M_S)} \sum_{r \in R} (\beta_{ijr}^2 + \beta_{jir}^2)\theta_r \leq min\{2|M_C|, 2|M_S|\}$ is obviously valid. Moreover, the inequality $\sum_{(i,j) \in A^+(M_C \cup M_S)} \sum_{r \in R} \beta_{ijr}^2 \theta_r \leq \left\lceil \max\left\{\sum_{i \in M_C \cup M_S} d_i, \sum_{i \in M_C \cup M_S} p_i\right\}/Q_2 \right\rceil$ holds according to inequalities (42). Then we have $\sum_{(i,j) \in E(M_C \cup M_S)} \sum_{r \in R} \beta_{ijr}^2 \theta_r \leq |M_C| + |M_S| - \left\lceil \max\left\{\sum_{i \in M_C \cup M_S} d_i, \sum_{i \in M_C \cup M_S} p_i\right\}/Q_2 \right\rceil$ since each customer should be visited exactly once by vehicles. As $\sum_{(i,j) \in E(M_C:M_S)} \sum_{r \in R} (\beta_{ijr}^2 + \beta_{jir}^2)\theta_r \leq \sum_{(i,j) \in E(M_C \cup M_S)} \sum_{r \in R} \beta_{ijr}^2 \theta_r$ holds, we have

$$
\begin{aligned}
\gamma &= \sum_{(i,j) \in E(M_C:M_S)} \sum_{r \in R} (\beta_{ijr}^2 + \beta_{jir}^2)\theta_r \\
&\leq min\left\{2|M_C|, 2|M_S|, |M_C| + |M_S| - \left\lceil \max\left\{\sum_{i \in M_C \cup M_S} d_i, \sum_{i \in M_C \cup M_S} p_i\right\}/Q_2 \right\rceil\right\}.
\end{aligned}
$$

and therefore Lemma 1 holds.

B.3. Proof of Lemma 2

The inequality $\sum_{(i,j) \in A^+(M_N) \cup A^-(M_N)} \sum_{r \in R} \beta_{ijr}^2 \theta_r \geq \sum_{(i,j) \in E(M_C:M_S)} \sum_{r \in R} (\beta_{ijr}^2 + \beta_{jir}^2)\theta_r = \gamma$ obviously holds, where $A^-(M_N) = \left\{(i,j) \mid (i,j) \in A_2, i \in (V_S \cup V_C) \backslash M_N, j \in M_N\right\}$ represents the set of arcs entering into M_N. Note that $\sum_{(i,j) \in A^+(M_N)} \sum_{r \in R} \beta_{ijr}^2 \theta_r = \sum_{(i,j) \in A^-(M_N)} \sum_{r \in R} \beta_{ijr}^2 \theta_r$. As a result, $\sum_{(i,j) \in A^+(M_N)} \sum_{r \in R} \beta_{ijr}^2 \theta_r \geq \lceil \gamma/2 \rceil$ and then $\delta = \sum_{(i,j) \in E(M_N)} \sum_{r \in R} \beta_{ijr}^2 \theta_r \leq |M_N| - \lceil \gamma/2 \rceil$ since each customer should be served exactly once.

B.4. Proof of Lemma 3

If $\gamma = 0$ or $\gamma > |M_S|$, the corresponding two inequalities can be easily deduced by two k-path inequalities defined on sets M_N and $(M_N \cup M_S)$, respectively (see Subsection 4.1). We mainly prove the second case, i.e., $1 \leq \gamma \leq |M_S|$. Let H^1 and H^2 be two subsets of M_S in which each customer has a monodirectional and bidirectional link with M_C, respectively, i.e.,

$H^e = \{k \in M_S | \sum_{(i,j)\in E(M_C:\{k\})} \sum_{r\in R}(\beta^2_{ijr} + \beta^2_{jir})\theta_r = e\}$ for $e \in \{1, 2\}$. The equality $\gamma = \sum_{(i,j)\in E(M_C:M_S)} \sum_{r\in R}(\beta^2_{ijr} + \beta^2_{jir})\theta_r = |H^1| + 2|H^2|$ obviously holds. We then obtain

$$\begin{aligned}
\sum_{(i,j)\in E(M_N)} \sum_{r\in R} \beta^2_{ijr}\theta_r &= \sum_{(i,j)\in E(M_N)\cup H^1\cup H^2} \sum_{r\in R} \beta^2_{ijr}\theta_r - |H^1| - 2|H^2| \\
&\le |M_N \cup H^1 \cup H^2| - \Big\lceil \sum_{i\in M_N\cup H^1\cup H^2} d_i/Q_2 \Big\rceil - |H^1| - 2|H^2|, \\
&= |M_N| - \Big\lceil \sum_{i\in M_N\cup H^1\cup H^2} d_i/Q_2 \Big\rceil - |H^2|, \\
&\le |M_N| - \Big\lceil \sum_{i\in M_N\cup\{v_1,\dots,v_{|H^1|+2|H^2|}\}} d_i/Q_2 \Big\rceil, \\
&\le |M_N| - \Big\lceil \sum_{i\in M_N\cup\{v_1,\dots,v_\gamma\}} d_i/Q_2 \Big\rceil, \\
&= |M_N| - \Big\lceil \Big(\sum_{i\in M_N} d_i + \sum_{j=1}^{\gamma} d_{v_j} \Big)/Q_2 \Big\rceil
\end{aligned}$$

$$\begin{aligned}
\sum_{(i,j)\in E(M_N)} \sum_{r\in R} \beta^2_{ijr}\theta_r &= \sum_{(i,j)\in E(M_N)\cup H^1\cup H^2} \sum_{r\in R} \beta^2_{ijr}\theta_r - |H^1| - 2|H^2| \\
&\le |M_N \cup H^1 \cup H^2| - \Big\lceil \sum_{i\in M_N\cup H^1\cup H^2} p_i/Q_2 \Big\rceil - |H^1| - 2|H^2|, \\
&= |M_N| - \Big\lceil \sum_{i\in M_N\cup H^1\cup H^2} p_i/Q_2 \Big\rceil - |H^2|, \\
&\le |M_N| - \Big\lceil \sum_{i\in M_N\cup\{v'_1,\dots,v'_{|H^1|+2|H^2|}\}} p_i/Q_2 \Big\rceil, \\
&\le |M_N| - \Big\lceil \sum_{i\in M_N\cup\{v'_1,\dots,v'_\gamma\}} p_i/Q_2 \Big\rceil, \\
&= |M_N| - \Big\lceil \Big(\sum_{i\in M_N} p_i + \sum_{j=1}^{\gamma} p_{v'_j} \Big)/Q_2 \Big\rceil
\end{aligned}$$

Therefore, we have $\delta = \sum_{(i,j)\in E(M_N)} \sum_{r\in R} \beta^2_{ijr}\theta_r \le |M_N| - \lceil \max\{\sum_{i\in M_N} d_i + \sum_{j=1}^{\gamma} d_{v_j}, \sum_{i\in M_N} p_i + \sum_{j=1}^{\gamma} p_{v'_j}\}/Q_2 \rceil$.

B.5. Proof of Lemma 4

We divide inequalities (61) into the following two parts:

$$\delta \le |M_N| - \gamma + |M_C| - \Big\lceil \Big(\sum_{i\in M_N\setminus M_C} d_i + \sum_{j=1}^{2|M_C|-\gamma} d_{v_j} + \sum_{j=1}^{2|M_C|-\gamma} d_{o_j} \Big)/Q_2 \Big\rceil \tag{66}$$

$$\delta \le |M_N| - \gamma + |M_C| - \Big\lceil \Big(\sum_{i\in M_N\setminus M_C} p_i + \sum_{j=1}^{2|M_C|-\gamma} p_{v'_j} + \sum_{j=1}^{2|M_C|-\gamma} p_{o'_j} \Big)/Q_2 \Big\rceil \tag{67}$$

We firstly give the proof for the first part (66). Note that $|M_C| \le \gamma \le 2|M_C|$, hence there are at least $\gamma - |M_C|$ customers in M_C with a bidirectional link with M_S. Let $F = \{k \in M_C | \sum_{(i,j)\in E(\{k\}:M_S)} \sum_{r\in R}(\beta^2_{ijr} + \beta^2_{jir})\theta_r = 2\}$ and $F' \subseteq F$ be the one of subset of F such that $|F'| = \gamma - |M_C|$. Define two new subsets $M'_C = M_C \setminus F'$, $M'_N = M_N \setminus F'$, and a new value $\gamma' = |M'_C|$.

In terms of capacity inequality on M'_N, we have:

$$\begin{aligned}\sum_{(i,j)\in E(M'_N)} \beta^2_{ijr}\theta_r &\leq |M'_N| - \Big\lceil \sum_{i\in M'_N} d_i/Q_2 \Big\rceil \\ &\leq |M'_N| - \Big\lceil \Big(\sum_{i\in M'_N} d_i + \sum_{j=1}^{\gamma'} d_{v_j} \Big)/Q_2 \Big\rceil \\ &= |M'_N| - \gamma' + |M'_C| \\ &\quad - \Big\lceil \Big(\sum_{i\in M'_N\setminus M'_C} d_i + \sum_{j=1}^{2|M'_C|-\gamma'} d_{v_j} + \sum_{j=1}^{2|M'_C|-\gamma'} d_{o_j} \Big)/Q_2 \Big\rceil \end{aligned} \tag{68}$$

If $\gamma = |M_C|$, we then have $F' = \varnothing$, $M'_C = M_C$, $M'_N = M_N$ and $\gamma = \gamma'$, the inequality (68) becomes inequality (66), hence inequality (66) is valid.

If $\gamma > |M_C|$, we then obtain $F' \neq \varnothing$. Starting with the inequality (68) on $M'_N = M_N \setminus F'$, we move the customers from F' into M'_N and M'_C one by one and prove the validity of the new resulting inequality (68). Let k be one of the customers in F', the new form of inequality (68) can be obtained by setting $M'_N := M'_N \cup \{k\}$ and $M'_C := M'_C \cup \{k\}$. Subsequently, the value of γ' will go up by 2, while the components $|M'_N| - \gamma' + |M'_C|$, $2|M'_C| - \gamma'$, $M'_N \setminus M'_C$ and $\sum_{j=1}^{2|M'_C|-\gamma'} d_{v_j}$ will not change. Note that the sum of delivery demands $\sum_{j=1}^{2|M'_C|-\gamma'} d_{o_j}$ may decrease. Therefore, the new inequality is still valid after adding customer k into M'_N and M'_C. Eventually, we can repeat this procedure until $M'_N = M_N$, $M'_C = M_C$, and $\gamma' = \gamma$. As a result, the first part (66) holds. The second part (67) can also be proved by the same procedure as the first part. As a result, Lemma 4 holds.

C. Two-phase primal heuristic

The pseudo-code of two-phase primal heuristic to generate a set of initial second-echelon routes $\bigcup_{s\in V_S} \bar{R}_s$ is outlined in Algorithm 1.

In the first phase, the customer groups will be assigned to satellites subject to the satellite-capacity constraints. We create a center point for each group with the coordinate being the average values of geographical coordinates of all customers in it. The profit for assigning a group to a satellite is set to the opposite value of the distance from the center point to the satellite. We reformulate the bin packing problem (43)-(48) for the customer set V_C but with a different objective function to determine a set of maximum-profit packing patterns, which will be solved by a MIP solver as it needs to be addressed only once in the primal heuristic (Line 1, Algorithm 1). By solving the problem, we confirm the assignment of customers to satellites (Line 5, Algorithm 1). The second phase is to generate the second-echelon routes by a greedy insertion heuristic. At the beginning, a total of $|T_s|$ empty routes for each satellite $s \in V_S$ are created. Subsequently, the heuristic iteratively inserts the unvisited customer with the largest comprehensive demand

Algorithm 1 Two-phase primal heuristic

1: Build and solve the previously mentioned bin packing problem by a MIP solver such as CPLEX;
2: Let U be the set of unvisited customers, $\bar{R}_s$ be the set of routes starting from and ending at satellite s, and $\hat{s}_i$ be the satellite to which the customer i is assigned;
3: Initialize the set of unvisited customers: $U \leftarrow V_C$;
4: For each satellite $s \in V_S$, $\bar{R}_s \leftarrow |T_s|$ empty routes without visiting any customers;
5: Initialize $\hat{s}_i, \forall i \in V_C$ according to the optimal solution to the bin packing problem;
6: **while** $U \neq \varnothing$ **do**
7: Select a customer $i \in U$ with the largest comprehensive demand;
8: Initialize minimum insertion cost $\hat{c}_{min} \leftarrow +\infty$;
9: **for all** $r \in \bar{R}_{\hat{s}_i}$ **do**
10: **for all** insertion positions in route r **do**
11: **if** the position $\hat{p}$ is feasible for inserting customer i **then**
12: Calculate the incremental cost $\hat{c}$ after insertion;
13: **if** $\hat{c} < \hat{c}_{min}$ **then**
14: $\hat{c}_{min} \leftarrow \hat{c}, r^* \leftarrow r, \hat{p}^* \leftarrow \hat{p}$;
15: **end if**
16: **end if**
17: **end for**
18: **end for**
19: Insert customer i into position $\hat{p}^*$ of route r^* and remove i from U;
20: **end while**
21: Remove the empty routes from $\bigcup_{s \in V_S} \bar{R}_s$;
22: **return** $\bigcup_{s \in V_S} \bar{R}_s$.

(the sum of delivery and pickup demand) into the best position of the best route in terms of the increasing cost. The insertion is checked feasible if the second-echelon vehicle capacity Q_2 is not violated (Line 11, Algorithm 1). If no customer can be inserted into the any non-empty route, an empty one will be used. The insertion process will stop once all customers are inserted and the resulting non-empty routes are then used as the initial second-echelon routes.

D. Instance-level results

Tables C1-C6 report in turn the instance-level results for the problems 2E-VRPG, 2E-VRPS and 2E-VRPGS. In these tables, the following results at the root node are provided: the lower bounds obtained without adding any valid inequalities ('lb_r'), the computing time for obtaining the lower bounds in column 'lb_r' ('t_r'), the lower bounds obtained after adding all families of valid inequalities ('lb_{rc}'), the computing time for obtaining the lower bounds in column 'lb_{rc}' ('t_{rc}'), and the number of valid inequalities identified at the root node ('cut_r'). In addition, the following results over the whole algorithm are given: the total computing times ('t'), the best upper bounds within time limit ('ub'), the total number of B&B nodes enumerated ('$node$'), and the total number of valid inequalities identified ('cut'). The entries in column 'ub' will be bolded if the corresponding instances were solved to optimality. Besides this, six entries in column 'Name' are also bolded

because the proposed BCP algorithm figured out better solutions to the corresponding instances, which have been claimed to be solved to optimality by Liu et al. (2018).

Table C1. Solutions to the 2E-VRPG on scattered instances

Name	ns	ng	nc	lb_r	t_r	lb_{rc}	t_{rc}	cut_r	t	ub	$node$	cut
s-1-a	2	4	20	872.4	0.0	1004.4	0.1	32	0.5	**1029.8**	9	48
s-1-b	2	4	20	891.2	0.0	945.2	0.1	37	1.2	**988.0**	30	79
s-1-c	2	4	20	897.4	0.1	973.7	0.2	26	1.1	**1024.6**	19	72
s-1-d	2	4	20	876.8	0.1	1021.2	0.3	42	4.6	**1068.7**	87	109
s-1-e	2	4	20	825.3	0.1	960.2	0.3	30	1.4	**1019.3**	12	63
s-2-a	3	6	30	1221.4	0.1	1261.0	0.6	44	21.5	**1522.4**	56	174
s-2-b	3	6	30	1145.9	0.2	1253.5	0.7	48	32.3	**1402.0**	129	440
s-2-c	3	6	30	1096.9	0.3	1171.9	1.1	53	22.5	**1240.8**	57	196
s-2-d	3	6	30	1129.4	0.2	1248.8	1.1	47	33.3	**1368.9**	159	401
s-2-e	3	6	30	942.6	0.2	1075.0	1.3	55	15.1	**1143.2**	23	143
s-3-a	4	8	40	1480.1	0.4	1701.1	2.0	74	103.7	**1881.0**	236	473
s-3-b	4	8	40	1483.7	0.3	1675.1	1.7	69	3752.5	**1868.9**	12,459	2,426
s-3-c	4	8	40	1292.9	0.7	1342.7	4.4	62	542.3	**1455.4**	265	850
s-3-d	4	8	40	1263.0	0.5	1355.4	2.5	64	631.2	**1448.2**	350	953
s-3-e	4	8	40	1362.0	0.8	1484.2	3.2	73	144.8	**1573.9**	85	286
s-4-a	5	10	50	1410.8	1.7	1595.9	6.4	76	6884.8	**1785.7**	2,869	314
s-4-b	5	10	50	1449.5	1.6	1517.9	11.4	70	7200.0	1776.2	8,994	443
s-4-c	5	10	50	1530.3	1.1	1595.5	8.8	73	7200.0	1806.7	14,702	780
s-4-d	5	10	50	1688.9	1.5	1761.3	4.3	65	5335.4	**1948.4**	1,317	506
s-4-e	5	10	50	1496.7	1.6	1637.2	5.9	67	7200.0	2000.6	8,056	399
s-5-a	6	12	60	2220.0	3.5	2267.4	33.3	95	7200.0	2860.9	6456	261
s-5-b	6	12	60	3072.4	1.9	3123.9	22.5	76	7200.0	3661.0	2716	163
s-5-c	6	12	60	2496.1	2.0	2556.2	11.7	86	6289.4	**2716.6**	4147	2309
s-5-d	6	12	60	2787.5	1.0	2818.4	8.3	67	7200.0	3521.8	5552	1577
s-5-e	6	12	60	2564.9	0.7	2618.8	7.6	90	5345.4	**2823.5**	6027	294
s-6-a	7	14	70	3561.0	1.4	3615.6	15.1	57	4224.0	**3846.0**	1190	433
s-6-b	7	14	70	3729.3	0.7	3792.6	4.0	52	7200.0	3955.9	3941	517
s-6-c	7	14	70	3573.9	0.4	3635.0	7.0	58	4831.0	**3752.2**	1806	758
s-6-d	7	14	70	4002.3	0.8	4067.9	15.1	53	7200.0	4342.2	573	363
s-6-e	7	14	70	3656.0	0.8	3776.0	11.0	33	7200.0	3934.3	994	474
s-7-a	8	16	80	5788.5	0.6	5799.5	2.7	44	7200.0	6088.1	5753	694
s-7-b	8	16	80	4826.4	1.0	4844.7	5.4	71	7200.0	5141.9	4602	1112
s-7-c	8	16	80	4542.9	0.5	4567.9	3.8	67	1233.8	**4620.9**	663	457
s-7-d	8	16	80	4574.7	1.0	4593.0	3.4	55	7200.0	4712.7	3542	927
s-7-e	8	16	80	5090.3	0.9	5107.7	3.0	66	7200.0	5364.6	10072	2316
s-8-a	9	18	90	4920.3	6.7	4957.1	24.8	51	7200.0	5272.1	514	567
s-8-b	9	18	90	5226.3	1.6	5242.4	3.2	74	7200.0	5483.9	4075	1078
s-8-c	9	18	90	6710.2	0.9	6739.9	9.4	54	7200.0	6964.3	1695	678
s-8-d	9	18	90	6333.1	0.8	6395.2	8.2	51	7200.0	6699.3	2796	1236
s-8-e	9	18	90	5564.0	0.6	5566.5	1.1	42	7200.0	5899.1	11216	2110
s-9-a	10	20	100	6143.3	0.9	6216.2	3.3	44	7200.0	6626.7	4003	1068
s-9-b	10	20	100	5372.5	1.0	5396.0	3.1	46	7200.0	5785.7	2136	679
s-9-c	10	20	100	5878.2	3.3	5878.2	3.3	59	7200.0	6075.2	2190	578
s-9-d	10	20	100	6978.0	1.7	7003.6	6.4	49	7200.0	7405.0	3028	836
s-9-e	10	20	100	7159.4	0.6	7163.4	1.0	52	7200.0	7460.3	9284	1592

Table C2. Solutions to the 2E-VRPG on clustered instances

Name	ns	ng	nc	lb_r	t_r	lb_{rc}	t_{rc}	cut_r	t	ub	$node$	cut
c-1-a	2	4	20	509	0.1	588.9	0.5	34	1.0	**588.9**	10	34
c-1-b	2	4	20	463.4	0.1	486.8	0.3	16	1.4	**536.8**	18	32
c-1-c	2	4	20	603	0.0	625.2	0.3	28	0.5	**625.7**	3	35
c-1-d	2	4	20	392.7	0.1	402.4	0.2	12	0.5	**411.4**	3	12
c-1-e	2	4	20	480	0.1	544.4	0.3	25	8.6	**545.4**	161	53
c-2-a	3	6	30	761.3	0.2	803.1	0.9	29	21.6	**935.1**	99	177
c-2-b	3	6	30	730.4	0.3	752.6	1.2	34	14.8	**827.0**	30	93
c-2-c	3	6	30	711.5	0.2	748.3	1.3	35	110.2	**856.7**	213	313
c-2-d	3	6	30	784.3	0.2	804.9	1.4	49	43.9	**932.8**	93	142
c-2-e	3	6	30	736.6	0.3	752.5	0.9	27	2.0	**762.5**	2	27
c-3-a	4	8	40	960	1.1	978.9	4.1	44	68.4	**1049.4**	87	50
c-3-b	4	8	40	1036.8	0.6	1082.1	6.1	60	1590.5	**1182.1**	446	1,073
c-3-c	4	8	40	1119	0.6	1235.7	4.2	64	831.2	**1385.6**	296	326
c-3-d	4	8	40	1116.2	0.5	1357.8	2.5	67	318.5	**1653.7**	555	2,066
c-3-e	4	8	40	1250.7	0.4	1602.9	2.1	68	22.4	**1787.6**	43	423
c-4-a	5	10	50	888.5	1.9	923.8	14.6	74	1404.0	**996.6**	580	559
c-4-b	5	10	50	1010.6	0.8	1071.6	9.4	79	2574.8	**1244.1**	1,215	572
c-4-c	5	10	50	1024.8	1.1	1140.1	6.6	59	4426.7	**1278.6**	8,372	8,518
c-4-d	5	10	50	1202.1	1.1	1481.0	4.1	78	493.6	**1666.8**	18,646	4,422
c-4-e	5	10	50	878.9	2.1	922.3	16.8	59	7200.0	1012.1	2,771	317
c-5-a	6	12	60	1869.7	3.8	1901.4	33.0	57	6531.7	**2144.7**	6888	983
c-5-b	6	12	60	1552.9	3.5	1577.6	25.8	66	5762.7	**1730.6**	5065	3490
c-5-c	6	12	60	1592.4	2.6	1611.8	20.6	53	7200.0	2181.4	834	3904
c-5-d	6	12	60	2292.3	2.2	2310.5	17.5	40	2538.0	**2389.8**	208	299
c-5-e	6	12	60	2160.0	1.3	2213.0	18.4	100	7200.0	3163.3	2279	3117
c-6-a	7	14	70	3268.2	0.8	3307.1	6.3	39	7200.0	3872.3	1225	1015
c-6-b	7	14	70	2188.3	0.5	2218.0	10.4	61	7200.0	3142.1	2531	781
c-6-c	7	14	70	3619.8	0.6	3672.5	3.2	57	7200.0	3965.1	4582	1126
c-6-d	7	14	70	3302.5	0.8	3368.3	18.3	60	7200.0	4622.4	451	644
c-6-e	7	14	70	2622.5	3.3	2678.8	66.6	115	1438.9	**2801.4**	70	193
c-7-a	8	16	80	3763.6	0.4	3784.0	19.3	63	7200.0	4114.2	1488	1969
c-7-b	8	16	80	2828.9	1.4	2842.4	10.8	41	7200.0	3017.0	353	846
c-7-c	8	16	80	3599.3	2.6	3623.5	20.4	72	7200.0	3878.9	2176	274
c-7-d	8	16	80	2630.2	1.5	2669.4	13.7	46	5974.8	**2848.8**	5437	3149
c-7-e	8	16	80	5000.6	0.9	5096.8	17.6	52	3467.5	**5234.0**	738	804
c-8-a	9	18	90	4685.4	6.4	4701.1	50.3	58	7200.0	5113.4	1781	1124
c-8-b	9	18	90	3727.6	1.7	3765.1	15.9	27	7200.0	4091.0	3133	2104
c-8-c	9	18	90	4230.0	1.3	4315.7	13.5	42	7200.0	4612.0	1249	1252
c-8-d	9	18	90	4683.8	0.6	4711.8	11.8	47	6757.6	**4868.5**	5648	3564
c-8-e	9	18	90	4462.9	0.9	4565.7	16.0	67	7200.0	4814.2	1610	1231
c-9-a	10	20	100	4829.9	4.0	4883.7	30.9	78	7200.0	5297.6	2278	2369
c-9-b	10	20	100	5205.8	0.9	5213.2	2.7	67	7200.0	5381.0	7166	4214
c-9-c	10	20	100	5591.0	1.3	5647.4	20.3	64	7200.0	6540.7	666	711
c-9-d	10	20	100	5339.1	5.4	5420.6	44.9	54	7200.0	5720.1	1734	1054
c-9-e	10	20	100	6086.6	8.6	6121.4	135.4	58	7200.0	6419.4	370	365

Table C3. Solutions to the 2E-VRPS on scattered instances

Name	ns	ng	nc	lb_r	t_r	lb_{rc}	t_{rc}	cut_r	t	ub	$node$	cut
s-1-a	2	4	20	871.8	0.0	963.2	0.2	39	1.9	**971.9**	26	76
s-1-b	2	4	20	901.3	0.1	935.3	0.2	32	2.5	**949.6**	53	63
s-1-c	2	4	20	865.9	0.1	950.3	0.2	32	2.2	**974.5**	31	113
s-1-d	2	4	20	901.0	0.1	995.0	0.3	32	15.6	**1021.4**	306	246
s-1-e	2	4	20	866.5	0.0	927.3	0.3	43	6.4	**961.2**	106	192
s-2-a	3	6	30	1186.7	0.1	1225.7	0.6	54	3363.1	**1288.4**	626	6,921
s-2-b	3	6	30	1181.5	0.1	1227.7	0.6	42	58.5	**1270.3**	513	1,918
s-2-c	3	6	30	1160.8	0.2	1191.2	0.6	28	1765.0	**1267.9**	1,023	3,207
s-2-d	3	6	30	1113.7	0.2	1175.2	1.0	65	5885.2	**1333.5**	4,625	9,559
s-2-e	3	6	30	1013.4	0.2	1068.3	0.9	86	6661.1	**1160.6**	4,320	6,378
s-3-a	4	8	40	1536.3	0.3	1673.7	1.2	69	5324.7	**1875.4**	2,468	6,343
s-3-b	4	8	40	1583.1	0.4	1641.3	1.5	73	7200.0	2022.9	12,466	4,536
s-3-c	4	8	40	1363.2	0.3	1408.2	1.4	53	7200.0	1801.4	14,666	13,208
s-3-d	4	8	40	1347.0	0.4	1376.8	1.9	36	7200.0	1743.8	9,358	2,023
s-3-e	4	8	40	1443.4	0.3	1468.7	1.6	68	4203.5	**1742.3**	3,844	6,665
s-4-a	5	10	50	1484.9	0.6	1543.4	3.1	89	7200.0	1896.7	6,433	11,607
s-4-b	5	10	50	1509.0	0.6	1553.1	3.1	74	7200.0	1989.6	4,927	13,540
s-4-c	5	10	50	1593.9	0.5	1637.7	2.8	84	7200.0	2007.3	1,093	8,201
s-4-d	5	10	50	1729.2	0.5	1765.9	1.6	65	7200.0	2231.7	1,386	6,843
s-4-e	5	10	50	1536.0	0.5	1601.1	2.2	66	7200.0	2109.3	11,179	12,953
s-5-a	6	12	60	2444.7	1.1	2475.7	10.7	75	7200.0	2899.5	1974	3697
s-5-b	6	12	60	2626.6	0.6	2663.7	7.3	74	7200.0	3151.5	9243	2097
s-5-c	6	12	60	2325.5	0.7	2375.7	30.2	158	7200.0	2793.1	4779	320
s-5-d	6	12	60	2443.0	0.4	2480.1	3.3	64	7200.0	3034.7	4818	4719
s-5-e	6	12	60	2198.6	0.3	2248.2	4.8	131	7200.0	2447.0	4216	2799
s-6-a	7	14	70	3128.5	0.8	3171.5	7.1	85	7200.0	3405.7	4699	816
s-6-b	7	14	70	3564.2	0.5	3614.3	9.4	105	7200.0	4010.7	1174	287
s-6-c	7	14	70	4023.1	0.4	4074.5	4.0	76	7200.0	4483.5	6890	461
s-6-d	7	14	70	3869.2	0.6	3897.1	3.3	50	7200.0	4309.3	7796	2346
s-6-e	7	14	70	3819.8	1.0	3854.8	4.4	59	7200.0	4058.7	3710	14005
s-7-a	8	16	80	4545.3	0.8	4576.1	5.1	72	7200.0	5332.2	4862	285
s-7-b	8	16	80	5119.2	0.6	5161.6	6.3	42	7200.0	5518.1	2588	4719
s-7-c	8	16	80	5063.4	0.4	5093.5	3.0	29	7200.0	5439.8	7910	893
s-7-d	8	16	80	4723.6	0.5	4743.9	2.7	54	7200.0	5115.7	5174	3954
s-7-e	8	16	80	4973.8	0.8	5015.6	6.8	76	7200.0	5364.7	1885	705
s-8-a	9	18	90	5241.1	1.2	5271.7	4.2	39	7200.0	5853.1	3077	2146
s-8-b	9	18	90	5343.6	0.8	5380.6	5.0	49	7200.0	5926.2	2860	1194
s-8-c	9	18	90	6645.4	0.4	6678.4	2.6	24	7200.0	6958.0	8213	269
s-8-d	9	18	90	6231.9	0.3	6269.5	2.8	38	7200.0	6502.3	4994	5643
s-8-e	9	18	90	5485.9	0.5	5523.0	3.6	75	7200.0	5884.1	3723	474
s-9-a	10	20	100	6981.0	0.4	7018.0	2.2	25	7200.0	7381.1	6132	416
s-9-b	10	20	100	5723.6	0.8	5760.4	4.5	52	7200.0	6052.2	1947	1143
s-9-c	10	20	100	6313.7	1.3	6335.2	7.7	45	7200.0	6995.4	1343	232
s-9-d	10	20	100	7110.1	0.7	7129.2	3.9	46	7200.0	7640.0	5454	253
s-9-e	10	20	100	7613.3	0.5	7649.1	4.2	45	7200.0	8082.7	5283	2152

Table C4. Solutions to the 2E-VRPS on clustered instances

Name	ns	ng	nc	lb_r	t_r	lb_{rc}	t_{rc}	cut_r	t	ub	$node$	cut
c-1-a	2	4	20	565.4	0.1	644.9	0.3	33	1.0	**644.9**	11	36
c-1-b	2	4	20	517.4	0.1	540.4	0.2	25	8.8	**586.4**	154	352
c-1-c	2	4	20	654.2	0.0	676.4	0.2	28	0.5	**676.9**	3	36
c-1-d	2	4	20	437.7	0.1	447.2	0.2	9	0.5	**447.2**	3	9
c-1-e	2	4	20	524.9	0.1	589.5	0.2	27	6.8	**590.2**	169	55
c-2-a	3	6	30	806.3	0.1	818.8	0.4	19	112.3	**918.5**	804	4,017
c-2-b	3	6	30	760.2	0.2	794.6	0.8	44	50.0	**798.4**	361	162
c-2-c	3	6	30	782.9	0.2	815.8	0.6	44	775.1	**905.8**	4,658	1,534
c-2-d	3	6	30	856.1	0.1	876.0	0.7	49	808.9	**979.6**	5,775	1,745
c-2-e	3	6	30	743.4	0.3	766.3	0.7	20	18.8	**822.0**	150	527
c-3-a	4	8	40	1068.1	0.3	1085.0	0.9	36	20.2	**1135.5**	94	105
c-3-b	4	8	40	1138.2	0.3	1180.1	1.4	63	2509.2	**1255.5**	3,364	3,021
c-3-c	4	8	40	1222.7	0.2	1265.1	0.8	36	7200.0	1498.4	10,592	10,623
c-3-d	4	8	40	1212.8	0.2	1240.5	0.8	51	7200.0	1600.1	1,715	3,954
c-3-e	4	8	40	1370.5	0.2	1429.9	0.8	47	639.4	**1496.8**	3,208	7,708
c-4-a	5	10	50	1014.2	0.6	1048.3	7.2	82	7200.0	1174.0	1,095	12,126
c-4-b	5	10	50	1130.3	0.5	1178.5	2.5	83	7200.0	1446.7	8,417	2,895
c-4-c	5	10	50	1150.3	0.4	1178.0	1.8	52	7200.0	1717.2	11,499	2,259
c-4-d	5	10	50	1296.3	0.5	1427.1	1.7	68	7200.0	1576.6	2,467	11,743
c-4-e	5	10	50	973.8	0.7	1020.0	3.0	74	7200.0	1291.8	7,960	13,085
c-5-a	6	12	60	2022.4	0.7	2061.3	7.1	76	7200.0	2477.0	4838	777
c-5-b	6	12	60	2181.5	0.8	2206.1	4.1	41	7200.0	2641.8	4568	9183
c-5-c	6	12	60	1681.3	0.7	1701.3	4.1	78	7200.0	2078.2	5586	3140
c-5-d	6	12	60	2147.0	0.5	2169.6	3.9	57	7200.0	2513.2	3590	8200
c-5-e	6	12	60	1637.6	0.6	1660.3	4.4	76	7200.0	2123.9	3477	1579
c-6-a	7	14	70	2974.6	0.4	2993.5	2.3	61	7200.0	3386.0	6352	4271
c-6-b	7	14	70	2686.8	0.5	2708.9	9.6	112	7200.0	3004.3	1446	296
c-6-c	7	14	70	3112.9	0.3	3125.5	2.1	31	7200.0	3289.3	13589	8031
c-6-d	7	14	70	3403.6	0.3	3416.4	1.8	37	4321.5	**3558.8**	8325	24803
c-6-e	7	14	70	3665.7	0.7	3705.0	5.4	75	7200.0	4068.3	4162	2625
c-7-a	8	16	80	5081.4	0.5	5107.0	2.4	28	7200.0	5506.0	7393	4281
c-7-b	8	16	80	4184.5	0.5	4204.9	1.2	25	7200.0	4436.6	9767	5857
c-7-c	8	16	80	3666.1	0.3	3692.8	2.8	50	7200.0	4094.5	7685	5191
c-7-d	8	16	80	3547.5	0.6	3554.4	2.2	34	7200.0	3704.2	10910	2282
c-7-e	8	16	80	4427.5	0.4	4460.9	2.5	41	7200.0	4949.0	9155	522
c-8-a	9	18	90	4333.9	1.3	4365.5	8.8	60	7200.0	4630.4	3871	452
c-8-b	9	18	90	4759.8	0.8	4782.8	2.5	52	7200.0	5218.0	5670	7211
c-8-c	9	18	90	5578.9	0.5	5600.8	3.1	44	7200.0	5967.4	6524	1479
c-8-d	9	18	90	5249.9	0.4	5266.4	1.9	65	7200.0	5607.7	10227	1276
c-8-e	9	18	90	4575.9	0.4	4600.4	3.0	41	7200.0	4802.3	7648	3727
c-9-a	10	20	100	4380.9	0.8	4397.0	6.3	68	7200.0	5037.2	2335	208
c-9-b	10	20	100	6024.2	0.9	6035.7	3.1	78	7200.0	6525.2	7231	5249
c-9-c	10	20	100	6422.2	0.5	6450.6	3.4	43	7200.0	6765.7	5985	1274
c-9-d	10	20	100	6963.6	1.0	6987.3	9.9	57	7200.0	7430.9	4561	2491
c-9-e	10	20	100	6621.2	0.7	6632.5	3.8	30	7200.0	7212.7	5204	1114

Table C5. Solutions to the 2E-VRPGS on scattered instances

Name	ns	ng	nc	lb_r	t_r	lb_{rc}	t_{rc}	cut_r	t	ub	$node$	cut
s-1-a	2	4	20	1018.8	0.2	1151.3	0.6	42	2.3	**1176.2**	15	54
s-1-b	2	4	20	1039.6	0.1	1095.6	0.4	41	4.3	**1135.4**	45	76
s-1-c	2	4	20	1037.5	0.1	1115.5	0.4	33	3.0	**1164.6**	29	83
s-1-d	2	4	20	1021.7	0.2	1165.0	0.5	30	3.3	**1340.9**	27	86
s-1-e	2	4	20	977.2	0.2	1123.1	0.6	37	3.8	**1178.4**	19	91
s-2-a	3	6	30	1407.7	0.5	1450.6	1.3	39	110.9	**1737.5**	197	282
s-2-b	3	6	30	1337.4	0.3	1448.8	1.0	48	109.0	**1638.2**	444	461
s-2-c	3	6	30	1297.0	0.5	1374.7	2.7	44	278.2	**1503.7**	340	666
s-2-d	3	6	30	1324.2	0.4	1375.4	1.7	59	493.5	**1630.0**	1,457	832
s-2-e	3	6	30	1136.9	0.6	1275.6	3.5	54	153.6	**1382.4**	98	134
s-3-a	4	8	40	1801.9	0.8	2021.6	4.2	75	1460.0	**2273.2**	2,027	1,130
s-3-b	4	8	40	1815.8	0.9	2003.7	4.3	69	7200.0	2269.5	15,624	3,910
s-3-c	4	8	40	1625.0	1.2	1671.3	7.1	57	7200.0	2097.1	10,345	1,771
s-3-d	4	8	40	1587.0	1.4	1628.4	11.7	61	5524.6	**1887.9**	8,549	2,139
s-3-e	4	8	40	1683.0	1.1	1806.6	4.7	64	2680.9	**1964.0**	845	1,570
s-4-a	5	10	50	1705.3	3.5	1887.8	13.9	76	7200.0	2996.0	1,590	394
s-4-b	5	10	50	1752.4	4.1	1820.0	33.6	99	7200.0	2282.4	1,244	368
s-4-c	5	10	50	1826.0	1.8	1889.9	14.2	68	7200.0	2346.7	3,985	398
s-4-d	5	10	50	2012.1	1.4	2087.1	7.8	73	7200.0	2382.2	4,877	664
s-4-e	5	10	50	1794.7	3.0	1932.5	11.8	63	7200.0	2470.1	8,534	479
s-5-a	6	12	60	2866.7	5.7	2934.4	101.6	96	7200.0	3751.8	1856	423
s-5-b	6	12	60	3424.8	1.3	3465.6	18.1	68	7200.0	3949.5	1881	331
s-5-c	6	12	60	2570.5	4.5	2624.8	26.7	62	7200.0	2810.2	5209	637
s-5-d	6	12	60	3218.4	2.3	3249.9	11.5	50	7200.0	3792.5	2316	1659
s-5-e	6	12	60	2531.1	1.4	2598.0	10.9	79	7200.0	2867.8	4754	160
s-6-a	7	14	70	3906.4	0.9	3933.6	5.0	10	7200.0	4287.5	1952	210
s-6-b	7	14	70	3528.2	1.2	3615.5	16.0	52	7200.0	3911.5	975	566
s-6-c	7	14	70	4906.6	0.6	4958.0	6.2	41	7200.0	5331.5	3697	964
s-6-d	7	14	70	4220.3	1.6	4301.2	22.8	61	7200.0	4788.1	737	478
s-6-e	7	14	70	4634.7	0.8	4689.0	28.9	47	7200.0	4981.0	308	267
s-7-a	8	16	80	5660.2	1.2	5668.7	2.2	61	7200.0	6152.6	3343	566
s-7-b	8	16	80	4137.0	2.1	4168.6	20.0	37	7200.0	4678.9	1598	652
s-7-c	8	16	80	5889.9	0.5	5928.3	2.8	31	7200.0	6337.6	3789	811
s-7-d	8	16	80	4665.9	1.1	4713.2	14.7	29	7200.0	5207.0	1600	359
s-7-e	8	16	80	5352.3	1.9	5427.1	18.8	45	7200.0	6057.1	695	68
s-8-a	9	18	90	5456.2	5.4	5457.6	8.1	52	7200.0	5912.8	1142	1002
s-8-b	9	18	90	6526.0	2.9	6585.6	47.8	41	7200.0	7460.5	560	546
s-8-c	9	18	90	6774.1	1.3	6796.3	7.2	29	7200.0	7414.2	949	629
s-8-d	9	18	90	6281.1	2.2	6341.4	8.8	41	7200.0	6847.8	1043	711
s-8-e	9	18	90	5913.5	1.1	5918.0	1.9	31	7200.0	6282.7	3845	1101
s-9-a	10	20	100	7465.2	1.4	7493.4	2.6	25	7200.0	8232.5	3153	545
s-9-b	10	20	100	6403.3	2.0	6460.2	10.3	25	7200.0	7321.0	686	234
s-9-c	10	20	100	7811.9	4.7	7811.9	4.7	31	7200.0	8424.8	1525	865
s-9-d	10	20	100	8469.8	1.7	8486.1	7.9	53	7200.0	9591.7	1189	613
s-9-e	10	20	100	8461.9	1.2	8466.6	1.7	72	7200.0	9342.0	3998	752

Table C6. Solutions to the 2E-VRPGS on clustered instances

Name	ns	ng	nc	lb_r	t_r	lb_{rc}	t_{rc}	cut_r	t	ub	$node$	cut
c-1-a	2	4	20	665.4	0.2	744.9	0.8	33	3.1	**744.9**	21	33
c-1-b	2	4	20	617.4	0.3	640.4	0.6	18	3.0	**690.4**	20	42
c-1-c	2	4	20	754.2	0.1	776.4	0.6	36	1.2	**776.9**	6	40
c-1-d	2	4	20	537.7	0.3	547.2	0.6	10	1.4	**547.2**	7	10
c-1-e	2	4	20	624.9	0.3	688.9	0.7	22	12.2	**690.2**	131	62
c-2-a	3	6	30	948.0	0.4	993.0	1.9	43	223.5	**1167.0**	560	464
c-2-b	3	6	30	926.7	0.6	949.9	2.4	39	171.4	**1066.4**	170	155
c-2-c	3	6	30	904.2	0.5	939.2	2.3	40	976.3	**1091.4**	613	633
c-2-d	3	6	30	977.5	0.4	998.0	2.0	46	590.6	**1169.6**	505	540
c-2-e	3	6	30	923.6	0.5	936.3	1.4	22	49.1	**1017.0**	55	38
c-3-a	4	8	40	1278.1	1.5	1295.5	8.3	38	1095.4	**1415.5**	641	328
c-3-b	4	8	40	1348.2	1.0	1392.4	10.3	62	7200.0	1652.4	1,996	1,550
c-3-c	4	8	40	1439.7	1.0	1553.9	7.9	61	6832.4	**1725.8**	2,787	1,486
c-3-d	4	8	40	1436.5	0.9	1676.9	5.2	66	2454.7	**2002.4**	6,494	2,904
c-3-e	4	8	40	1584.3	0.4	1945.7	2.5	78	415.1	**2169.9**	743	922
c-4-a	5	10	50	1187.8	4.4	1227.2	33.1	77	7200.0	1433.8	1,874	186
c-4-b	5	10	50	1321.7	0.9	1388.9	14.2	92	7200.0	2001.6	4,581	359
c-4-c	5	10	50	1331.1	2.3	1450.6	13.5	59	7200.0	1769.6	11,170	2,239
c-4-d	5	10	50	1508.7	1.4	1782.6	6.3	77	7200.0	2059.3	8,150	750
c-4-e	5	10	50	1161.9	4.7	1221.5	32.1	70	7200.0	1417.1	6,560	466
c-5-a	6	12	60	2287.5	3.3	2318.4	42.2	92	7200.0	3242.5	7076	5162
c-5-b	6	12	60	2537.0	5.4	2567.0	63.9	78	7200.0	2801.4	3927	357
c-5-c	6	12	60	1922.6	3.2	1941.6	23.3	87	5437.8	**2044.4**	3217	867
c-5-d	6	12	60	2840.4	1.8	2856.3	11.6	29	7200.0	3259.0	942	1530
c-5-e	6	12	60	1867.4	1.4	1908.1	22.6	91	6103.4	**2057.0**	6155	2942
c-6-a	7	14	70	3911.8	0.6	3983.3	15.3	71	7200.0	4279.7	1247	496
c-6-b	7	14	70	3075.7	1.5	3123.1	13.3	48	7200.0	3467.8	3299	1467
c-6-c	7	14	70	4136.8	0.5	4211.6	6.1	59	7200.0	4530.6	2209	674
c-6-d	7	14	70	3196.8	3.2	3260.4	34.5	65	7200.0	3742.7	17819	168
c-6-e	7	14	70	2796.8	3.2	2810.6	35.3	68	7200.0	3267.8	1309	839
c-7-a	8	16	80	5544.0	0.9	5626.0	28.5	70	7200.0	5952.7	618	506
c-7-b	8	16	80	3359.3	2.0	3370.5	29.4	40	7200.0	3772.6	184	362
c-7-c	8	16	80	3521.2	3.2	3556.3	34.9	67	7200.0	3950.0	1970	108
c-7-d	8	16	80	3589.7	3.3	3651.8	37.5	70	7200.0	3923.2	3307	215
c-7-e	8	16	80	5472.4	0.7	5707.1	23.5	70	7200.0	6236.5	1192	274
c-8-a	9	18	90	4426.4	4.4	4477.8	84.7	40	7200.0	4930.3	112	144
c-8-b	9	18	90	4748.8	4.8	4829.1	35.5	35	7200.0	5142.1	4727	1067
c-8-c	9	18	90	5880.5	1.7	5895.9	26.3	44	7200.0	6376.6	3922	897
c-8-d	9	18	90	6153.0	0.7	6263.5	17.0	42	7200.0	6983.2	2723	742
c-8-e	9	18	90	4726.0	1.4	4745.0	14.6	38	7200.0	5210.4	1916	674
c-9-a	10	20	100	4989.6	4.6	5052.8	64.2	72	7200.0	5785.7	899	589
c-9-b	10	20	100	5735.9	1.2	5782.7	8.6	52	7200.0	6496.6	2324	713
c-9-c	10	20	100	6342.2	1.1	6381.2	17.0	42	7200.0	6912.9	4964	1152
c-9-d	10	20	100	6791.9	5.2	6874.6	79.5	57	7200.0	7609.3	1190	570
c-9-e	10	20	100	6104.3	2.5	6156.8	52.0	53	7200.0	6860.0	185	232