Accepted Manuscript

A branch-and-cut algorithm for the two-echelon capacitated vehicle routing problem with grouping constraints

Tian Liu, Zhixing Luo, Hu Qin, Andrew Lim

PII: S0377-2217(17)30930-X
DOI: 10.1016/j.ejor.2017.10.017
Reference: EOR 14741

To appear in: *European Journal of Operational Research*

Received date: 15 March 2017
Revised date: 23 September 2017
Accepted date: 10 October 2017

Highlights

- Addressed a new two-echelon vehicle routing problem
- Formulated the problem into a mathematical programming model
- Proposed six families of valid inequalities to strengthen the model
- Implemented a branch-and-cut algorithm to exactly solve the problem
- Evaluated the algorithm by extensive numerical experiments

A branch-and-cut algorithm for the two-echelon capacitated vehicle routing problem with grouping constraints

Tian Liu[a], Zhixing Luo[b,d,*], Hu Qin[c], Andrew Lim[b,d]

[a] School of Economics and Management, Dongguan University of Technology, Dongguan 523000, PR China
[b] School of Management and Engineering, Nanjing University, Nanjing 210093, PR China
[c] School of Management, Huazhong University of Science and Technology, No. 1037, Luoyu Road, Wuhan, PR China
[d] Department of Industrial Systems Engineering and Management, National University of Singapore, Singapore

Abstract

In this paper, we present a branch-and-cut algorithm for the two-echelon capacitated vehicle routing problem with grouping constraints (2E-CVRPGC), which is a new problem deriving from the classical two-echelon capacitated vehicle routing problem (2E-CVRP) by considering grouping constraints in the second echelon. Customers in the 2E-CVRPGC are divided into several disjoint groups, and the grouping constraints ensure that customers from the same group are served by vehicles from the same satellite. We formulate the problem as a mixed 0-1 linear program and propose five families of valid inequalities to strengthen the model. Based on the model and the inequalities, we implement a branch-and-cut algorithm to solve the problem. The proposed branch-and-cut algorithm was evaluated on two classes of randomly generated instances. The computational results show that the five families of valid inequalities can substantially improve the lower bounds yielded by the LP relaxation of the model, and the branch-and-cut algorithm can solve more instances to optimality than CPLEX. We also conduct additional experiments to analyze the impacts of the grouping constraints on the problem, and illustrate the differences between the 2E-CVRPGC and the 2E-CVRP.

Keywords: cutting; branch and cut; polyhedral combinatorics; two-echelon vehicle routing;

*Corresponding author: School of Management and Engineering, Nanjing University, No. 22, Hankou Road, Nanjing, China. Tel.: +86 13222052602.

Email addresses: liutian@dgut.edu.cn (Tian Liu), luozx.hkphd@gmail.com (Zhixing Luo), tigerqin1980@gmail.com (Hu Qin), alim.china@gmail.com (Andrew Lim)

1. Introduction

The vehicle routing problem (VRP) has been a hot research topic in combinatorial optimization for decades since it was proposed by Dantzig and Ramser (1959). Given a set of customers, each having a demand, and a fleet of identical vehicles, the VRP calls for a set of routes for the vehicles to serve the customers subject to the capacity constraints of the vehicles, while minimizing the total travel distance of the vehicles (Toth and Vigo, 2014). The *two-echelon capacitated vehicle routing problem* (2E-CVRP) is a well-known variant of the classic VRP. It involves determining a set of optimal routes for a two-level freight distribution system, where cargoes are delivered from a depot to a subset of intermediary satellites in the first level, and from the satellites to customers in the second level. To achieve cost effectiveness, delivery tasks in the first level are usually accomplished by large identical trucks, while delivery tasks in the second level are usually accomplished by small identical vehicles. The satellites can be left unvisited or visited by one or multiple vehicles, depending on the amount of cargoes delivered to them, whereas the customers must be served exactly once. Each customer has a demand, and each vehicle and satellite have a capacity. The amount of cargoes loaded in a vehicle and delivered to a satellite cannot exceed the capacities of the vehicle and the satellite, respectively. The number of each type of vehicles is bounded, and additionally, the number of small vehicles associated with each satellite is also bounded. Loading and unloading cargoes at each satellite results in handling costs, which are proportional to the amount of cargoes involved. The objective is to minimize the summation of the total routing cost of the vehicles and the total handling cost of the satellites. Applications of the 2E-CVRP arise in multi-echelon city logistics systems (Crainic et al., 2009; Grangier et al., 2016; Taniguchi et al., 2001; Zhao et al., 2008; Perboli et al., 2011; Jepsen et al., 2013). The 2E-CVRP contains the classic *location routing problem* (LRP) as a special case (Baldacci et al., 2011; Nguyen et al., 2006; Schiffer and Walther, 2017; Nadizadeh and Nasab, 2014). Given a set of possible depot locations and a set of customers, the LRP requires to decide the set of optimal locations to open depots and a set of optimal routes for the vehicles to serve the customers. Baldacci et al. (2013) proposed a method to convert an LRP instance to an equivalent 2E-CVRP instance.

In this paper, we study the *two-echelon capacitated vehicle routing problem with grouping* *constraints* (2E-CVRPGC), an extension of the classic 2E-CVRP where customers are divided into several disjoint groups beforehand and customers from the same group must be served by vehicles from the same satellite. The new constraints in the 2E-CVRPGC are referred to as the *grouping constraints*. The grouping constraints are motivated by a project done by our team for an appliance retailer in China. For some historical reasons, our client organizes their distribution network according to the administrative divisions of the country. For each administrative division, it establishes a branch to manage the stores in it and sets up a warehouse to replenish the inventories of the corresponding stores. There exists a very obvious drawback in this kind of distribution network. That is, there are too many small warehouses, which result in very high inventory and manpower costs. Therefore, our client wants to reorganize its distribution network by merging the warehouses in each administrative division into several large modern warehouses, but hoping that the stores under the same branch are replenished by the same warehouse so as to reduce impacts on the existing management system and the organization structure. Actually, we see many Chinese retailers proposing the same requirement when trying to reorganize their distribution network to reduce logistics costs. The result is that, in the two-echelon distribution system of Chinese city logistic, customers from the same administrative division are required to be served by vehicles from the same satellite, which is equivalent to assigning customers from the same administrative division to a satellite as a whole. According to our knowledge, the grouping constraint has not been studied in the literature of the 2E-CVRP yet. We give an example to illustrate the grouping constraints in Figure 1, where all customers are divided into four groups, and the optimal solution for this instance is shown. We can see that all customers in each group are served by the vehicles from the same satellite.

1.1. Literature Review

As the 2E-CVRP is a relatively new problem, solution approaches for the 2E-CVRP are limited. The exact approaches for the 2E-CVRP are mainly based on branch-and-cut algorithms. Feliu et al. (2008) proposed the first branch-and-cut algorithm for the 2E-CVRP based on a commodity flow model. Perboli et al. (2009a,b, 2011) provided a series of comprehensive studies on the 2E-CVRP and proposed several families of new valid inequalities based on the characteristics

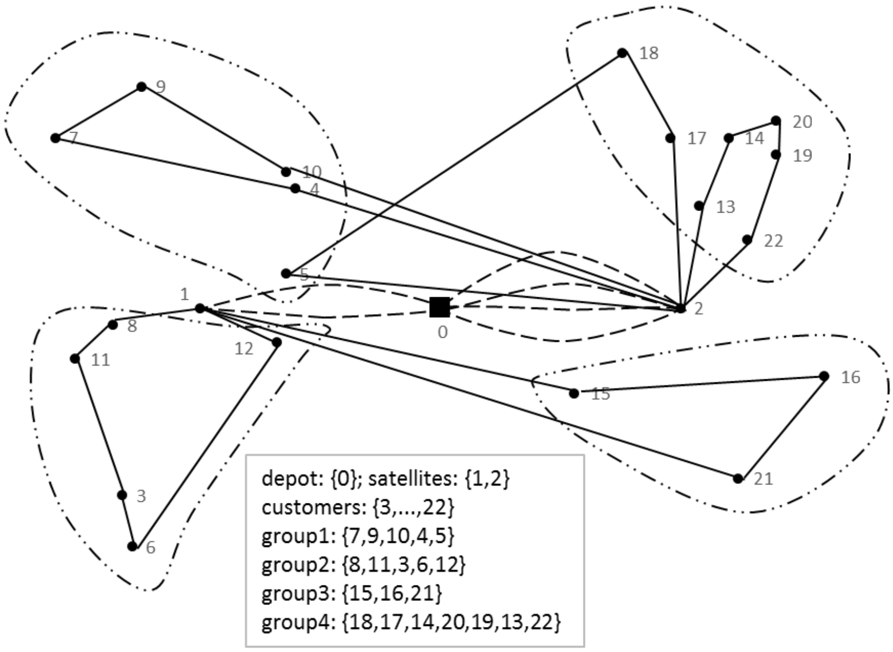

Figure 1: An example for grouping constraints.

of the 2E-CVRP. The new valid inequalities were implemented in the branch-and-cut algorithms to solve the 2E-CVRP and demonstrate their effectiveness. Jepsen et al. (2013) proposed a new branch-and-cut algorithm which was based on a relaxed edge flow model and guaranteed the final optimal feasible solution by a special branching scheme. This branch-and-cut algorithm demonstrated its superiority over the other branch-and-cut algorithms for the 2E-CVRP in the literature through computational experiments. The other exact algorithm for the 2E-CVRP is the hybrid algorithm proposed by Baldacci et al. (2013), which is based on a new mathematical formulation for deriving valid lower bounds, and solves the 2E-CVRP by decomposing the original problem into a limited set of multi-depot capacitated vehicle routing problems with side constraints. According to our knowledge, the exact algorithm proposed by Baldacci et al. (2013) is the best one for the 2E-CVRP to date. Heuristics for the 2E-CVRP include clustering-based heuristics (Crainic et al., 2008), multi-start heuristics (Crainic et al., 2011), two math-based heuristics (Perboli et al., 2011) and an adaptive large neighborhood search (Hemmelmayr et al., 2012). We refer the reader to Cuda et al. (2015) for a survey of the literature on two-echelon routing problems.

1.2. Contributions

In this paper, we introduce the 2E-CVRPGC, a new variant of the 2E-CVRP that originates in warehouse re-organization for Chinese retailers. To solve this problem, we propose a mathematical model which involves three-index variables in the first echelon and two-index variables in the second echelon. The three-index formulation for the first echelon is caused by the split-delivery, and the two-index formulation for the second echelon can result in fewer variables and less computation burden. To further strengthen the model, we propose five families of valid inequalities, which are based on some classic valid inequalities for the other routing problems but consider special structures of the 2E-CVRPGC. For instance, by considering the grouping constraints, we derive the *strengthened capacity inequalities* for the second echelon formulation based on the classic capacity inequalities for the CVRP (Laporte and Nobert, 1983; Fukasawa et al., 2006). According to the computational experiments, the strengthened capacity inequalities perform much better than the capacity inequalities on the 2E-CVRPGC. Based on the model and the inequalities, we implement a branch-and-cut algorithm to solve the 2E-CVRPGC. To evaluate the branch-and-cut algorithm, we generate a class of scattered instances and a class of clustered instances. Because of the good lower bounds from the strengthened model, the branch-and-cut algorithm outperforms ILOG CPLEX on both classes of the instances. In addition, we analyze the impacts brought from the grouping constraints by comparing differences between the optimal solutions of the 2E-CVRP with and without the grouping constraints. Our computational results show that the grouping constraints increase optimal costs by 51.31% and 64.92% on the scattered and the clustered instances, respectively. This result demonstrates the strong influence of the grouping constraints on the structure of the 2E-CVRP.

The remainder of this paper is organized as follows. Section 2 presents a mathematical model for the 2E-CVRPGC. Section 3 introduces five families of valid inequalities to strengthen the mathematical model. The branch-and-cut algorithm is presented in Section 4. Section 5 reports the computational results, and Section 6 concludes the paper with some closing remarks.

2. Mathematical model

The 2E-CVRPGC is defined on an undirected graph $G = (V, A)$, where $V = V_0 \cup V_S \cup V_C$ is the node set. $V_0 = \{0\}$, V_S and V_C represents the depot, the set of satellites and the set of customers, respectively. $A = A_1 \cup A_2$ is the arc set, where $A_1 = \{(i, j) \mid i, j \in V_0 \cup V_S\}$ represents the set of arcs connecting the depot and the satellites, and $A_2 = \{(i, j) \mid i, j \in V_S \cup V_C, i \neq j\} \setminus \{(i, j) \mid i, j \in V_S, i \neq j\}$ represents the set of arcs connecting the satellites and the customers. Set V_C can be further divided into m disjoint groups, i.e., $V_C = C_1 \cup \ldots \cup C_m$ and $C_i \cap C_j = \emptyset$ for $\forall\, i \neq j$. Let $\mathbb{C} = \{C_1, \ldots, C_m\}$. Customers in the same group must be served by vehicles from the same satellite. Each customer $i \in V_C$ has a demand d_i, and each satellite $s \in V_S$ has a capacity Q_s and a handling cost h_s for each unit of freight. Each arc $(i, j) \in A$ has a travel cost $c_{i,j}$ for vehicles. Vehicles in each echelon are identical. The capacities of the vehicles in the first and the second echelons are Q_1 and Q_2, respectively. The numbers of available vehicles in each echelon and each satellite are bounded, with T_1 for the first echelon, T_2 for the second echelon and T_s for each satellite $s \in V_S$. We use K_1 to denote the set of available vehicles in the first echelon. The 2E-CVRPGC requires to determine a set of routes for the vehicles in the first echelon which ship cargoes from the depot to the satellites, and a set of routes for the vehicles in the second echelon which ship cargoes from the satellites to the customers. The objective is to minimize the total travel cost of the vehicles as well as the handling cost at the satellites.

Now we present the mathematical model for the 2E-CVRPGC. The decision variables in the model are as follows.

- $x_{i,j,k}$ ($(i, j) \in A_1, k \in K_1$): a binary decision variable and relative to the first echelon vehicles, which is equal to 1 if arc (i, j) is traveled by vehicle k, and 0 otherwise;

- $w_{s,k}$ ($s \in V_S, k \in K_1$): a decision variable representing the quantity delivered to satellite s by vehicle k;

- $u_{s,k}$ ($s \in V_S, k \in K_1$): a decision variable representing the position of satellite s in the route of vehicle k;

- $f_{i,j}$ $((i, j) \in A_2)$: a decision variable representing the load of the vehicle when it travels through arc (i, j);

- $y_{i,j}$ $((i, j) \in A_2)$: a binary decision variable and relative to the second echelon vehicles, which is equal to 1 if a vehicle travels through arc (i, j), and 0 otherwise;

- $z_{i,s}$ $(i \in V_C, s \in V_S)$: a binary variable which is equal to 1 if customer i is served by a vehicle from satellite s, and 0 otherwise;

- $v_{C_h,s}$ $(h = 1, \ldots, m, s \in V_S)$: a binary variable which is equal to 1 if customer group $C_h \in \mathbb{C}$ is assigned to satellite s, and 0 otherwise.

$$\min \sum_{k \in K_1} \sum_{(i,j) \in A_1} c_{i,j} x_{i,j,k} + \sum_{s \in V_S} h_s \sum_{k \in K_1} w_{s,k} + \sum_{(i,j) \in A_2} c_{i,j} y_{i,j} \tag{1}$$

$$\text{s.t.} \quad \sum_{(i,j) \in A_1} x_{i,j,k} = \sum_{(j,i) \in A_1} x_{j,i,k}, \ \forall\, i \in V_0 \cup V_S, k \in K_1 \tag{2}$$

$$\sum_{(i,j) \in A_1} x_{i,j,k} \leq 1, \ \forall\, i \in V_S, k \in K_1 \tag{3}$$

$$u_{i,k} + 1 \leq u_{j,k} + |V_S|(1 - x_{i,j,k}), \ \forall\, (i, j) \in A_1, k \in K_1 \tag{4}$$

$$w_{s,k} \leq Q_s \sum_{(s,i) \in A_1} x_{s,i,k}, \ \forall\, s \in V_S, k \in K_1 \tag{5}$$

$$\sum_{s \in V_S} w_{s,k} \leq Q_1, \ \forall\, k \in K_1 \tag{6}$$

$$\sum_{k \in K_1} w_{s,k} \leq Q_s, \ \forall\, s \in V_S \tag{7}$$

$$\sum_{(i,j) \in A_2} y_{i,j} = \sum_{(j,i) \in A_2} y_{j,i} = 1, \ \forall\, i \in V_C \tag{8}$$

$$\sum_{(s,i) \in A_2} y_{s,i} \leq T_s, \ \forall\, s \in V_S \tag{9}$$

$$\sum_{s \in V_S} \sum_{(s,i) \in A_2} y_{s,i} \leq T_2, \tag{10}$$

$$\sum_{(j,i) \in A_2} f_{j,i} = \sum_{(i,j) \in A_2} f_{i,j} + d_i, \ \forall\, i \in V_C \tag{11}$$

$$d_j y_{i,j} \leq f_{i,j} \leq (Q_2 - d_i) y_{i,j}, \ \forall \ (i, j) \in A_2 \tag{12}$$

$$z_{j,s} \geq z_{i,s} + y_{i,j} - 1, \ \forall \ (i, j) \in A_2 \tag{13}$$

$$z_{s,s} = 1, \ \forall \ s \in V_S \tag{14}$$

$$z_{i,s} = v_{C_h,s}, \ \forall \ i \in C_h, C_h \in \mathbb{C}, s \in V_S \tag{15}$$

$$\sum_{k \in K_1} w_{s,k} = \sum_{(s,i) \in A_2} f_{s,i}, \ \forall \ s \in V_S \tag{16}$$

$$x_{i,j,k} \in \{0, 1\}, \ \forall \ (i, j) \in A_1, k \in K_1 \tag{17}$$

$$w_{s,k} \geq 0, \ \forall \ s \in V_S, k \in K_1 \tag{18}$$

$$u_{s,k} \geq 0, \ \forall \ s \in V_S, k \in K_1 \tag{19}$$

$$f_{i,j} \geq 0, \ \forall \ (i, j) \in A_2 \tag{20}$$

$$y_{i,j} \in \{0, 1\}, \ \forall \ (i, j) \in A_2 \tag{21}$$

$$z_{i,s} \in \{0, 1\}, \ \forall \ i \in V_C, s \in V_S \tag{22}$$

$$v_{C_h,s} \in \{0, 1\}, \ \forall \ C_h \in \mathbb{C}, s \in V_S \tag{23}$$

The objective function (1) minimizes the total cost, consisting of the routing cost of the vehicles and the handling cost at the satellites. Constraints (2) are the flow conservation constraints for each satellite. Constraints (3) ensure that a vehicle can visit each satellite at most once. Constraints (4) are the subtour elimination constraints for the vehicles in the first echelon (Miller et al., 1960). Constraints (5) guarantee that only if a vehicle visits a satellite, the vehicle can deliver cargoes to that satellite. Constraints (6) and (7) are the capacity constraints of each vehicle in the first echelon and each satellite, respectively. Constraints (8) are the flow conservation constraints for each customer, and at the same time, ensure that each customer must be visited once. Constraints (9) and (10) ensure that the numbers of vehicles used in each satellite and in the second echelon cannot exceed their limits, respectively. Constraints (11) ensure that the demand of each customer is satisfied. Constraints (12) ensure the feasibility of flow on the arcs in the second echelon (Gavish, 1984). Constraints (13) and (14) ensure that two customers successively visited by a vehicle are assigned to the same satellite. Constraints (15) guarantee that customers in the same group are served by vehicles from the same satellite. Constraints (16) ensure that the amount of cargoes delivered to the customers from a satellite is equal to that delivered to this satellite from the depot. Finally, constraints (17)–(23) specify the domains of the decision variables.

3. Valid inequalities

We introduce five families of valid inequalities to strengthen the model. Before presenting them in detail, we first introduce some necessary notations used throughout this section. For a subset of customers $S \in V_C$,

- $A(S) = \{(i, j) \mid i, j \in S, (i, j) \in A_2\}$ denotes the set of arcs whose nodes are in S;
- $\bar{S} = \{i \mid i \in V_C, i \notin S\}$ denotes the set of customer nodes that are not in S;
- $\delta^+(S) = \{(i, j) \mid (i, j) \in A_2, i \in S, j \in \bar{S} \cup V_S\}$ denotes the set of arcs leaving S;
- $\delta^-(S) = \{(i, j) \mid (i, j) \in A_2, i \in \bar{S} \cup V_S, j \in S\}$ denotes the set of arcs entering S.

For two subsets $S_1, S_2 \subset V_C$ and $S_1 \cap S_2 = \emptyset$, let $A(S_1, S_2) = \{(i, j) \mid (i, j) \in A_2, i \in S_1, j \in S_2\}$ be the arcs pointing from the nodes in S_1 to the nodes in S_2. Similarly, for a subset of satellites $S \subset V_S$, let $A(S) = \{(i, j) \mid i, j \in S, (i, j) \in A_1\}$, $\bar{S} = \{i \mid i \in V_S, i \notin S\}$, $\delta^+(S) = \{(i, j) \mid (i, j) \in A_1, i \in S, j \in \bar{S} \cup V_0\}$ and $\delta^-(S) = \{(i, j) \mid (i, j) \in A_1, i \in \bar{S} \cup V_0, j \in S\}$.

3.1. Capacity inequalities for the customers

The *capacity inequalities* (CI) are a family of classic and widely-used inequalities for the VRP (Laporte and Nobert, 1983; Fukasawa et al., 2006). For a subset of customers $S \subset V_C$, the capacity inequality with respect to S is defined as:

$$\sum_{(i,j)\in\delta^+(S)} y_{i,j} \geq \left\lceil \frac{\sum_{i\in S} d_i}{Q_2} \right\rceil. \tag{24}$$

The right-hand-side value of (24) is a valid lower bound for the minimum number of vehicles required to serve all customers in S computed according to the vehicle capacity. Therefore, the capacity inequality (24) simply requires that the number of vehicles serving customers in S is not smaller than this lower bound.

3.2. *Strengthened capacity inequalities for the customers*

We introduce two types of strengthened capacity inequalities derived from the conflicts between different groups of customers. Let $Q_{max} = \max_{s \in V_S} \min\{Q_s, T_s Q_2\}$. For a subset of groups $\mathbb{C}' \subset \mathbb{C}$, let $V(\mathbb{C}') = \cup_{C_t \in \mathbb{C}'} C_t$ be the set of customers in $\mathbb{C}'$. If $\sum_{i \in V(\mathbb{C}')} d_i > Q_{max}$, the customers in $V(\mathbb{C}')$ cannot be assigned to a satellite simultaneously. Therefore, for a group of customers $C_k \in \mathbb{C}'$, there must exist at least another group of customers $C_h \in \mathbb{C}'$ such that vehicles from the customers in C_k cannot travel directly to the customers in C_h. Otherwise all of the customers in $V(\mathbb{C}')$ are served by vehicles from a satellite, which results in the violation of the capacity constraint of the satellite. For a group of customers $C_k \in \mathbb{C}'$, the first type of *strengthened capacity inequality* is defined as:

$$\sum_{(i,j) \in A(C_k, V_S)} y_{i,j} + \sum_{(i,j) \in A(C_k, V_C \setminus V(\mathbb{C}'))} y_{i,j} + \frac{|\mathbb{C}'| - 2}{|\mathbb{C}'| - 1} \sum_{(i,j) \in A(C_k, V(\mathbb{C}) \setminus C_k)} y_{i,j} \geq \left\lceil \frac{\sum_{i \in C_k} d_i}{Q_2} \right\rceil, \tag{25}$$

where $|\mathbb{C}'|$ is the number of groups in $\mathbb{C}'$.

Theorem 1. *The strengthened capacity inequality (25) is valid for the 2E-CVRPGC.*

Proof. Since there must be at least a group of customers in $\mathbb{C}'$ such that vehicles from the customers in C_k cannot travel to, at least one of the following inequalities is valid:

$$\sum_{(i,j) \in A(C_k, V_S)} y_{i,j} + \sum_{(i,j) \in A(C_k, V_C \setminus V(\mathbb{C}'))} y_{i,j} + \sum_{(i,j) \in A(C_k, V(\mathbb{C}') \setminus (C_k \cup C_h))} y_{i,j} \geq \left\lceil \frac{\sum_{i \in C_k} d_i}{Q_2} \right\rceil, \ \forall\, C_h \in \mathbb{C}' \setminus \{C_k\}. \tag{26}$$

By multiplying both sides of each inequality in (26) by $\frac{1}{|\mathbb{C}'|-1}$ and adding them together, we have

$$\sum_{(i,j) \in A(C_k, V_S)} y_{i,j} + \sum_{(i,j) \in A(C_k, V_C \setminus V(\mathbb{C}'))} y_{i,j} + \frac{1}{|\mathbb{C}'| - 1} \sum_{C_h \in \mathbb{C}' \setminus \{C_k\}} \sum_{(i,j) \in A(C_k, V(\mathbb{C}') \setminus (C_k \cup C_h))} y_{i,j} \geq \left\lceil \frac{\sum_{i \in C_k} d_i}{Q_2} \right\rceil, \tag{27}$$

which is equivalent to (25).

For a subset of customers $S \in V(\mathbb{C}')$, if $\left\lceil \frac{\sum_{i \in S} d_i}{Q_2} \right\rceil = 1$ and $S \cap C_k \neq \emptyset$, $\forall C_k \in \mathbb{C}'$, the second type of *strengthened capacity inequality* with respect to S is as follows:

$$\sum_{(i,j) \in \delta^+(S)} y_{i,j} \geq 2. \tag{28}$$

The idea behind (28) is simple. Since there exist conflicts among the groups in $\mathbb{C}'$, a vehicle cannot travel through all groups in $\mathbb{C}'$. Therefore, at least two vehicles are required to serve all customers in S.

3.3. Capacity inequalities for the satellites

Although the satellites in the 2E-CVRPGC are not required to be visited by the vehicles in the first echelon by requirements, some satellites may have to be visited by the vehicles because other satellites do not have enough capacities to satisfy all of the customers. For a subset of satellites $S \subset V_S$, let $D(S)$ be the minimum quantity delivered to the satellites in S in any feasible solution. We replace $D(S)$ with $D'(S) = \sum_{i \in V_C} d_i - \sum_{s \in V_S \setminus S} \min\{Q_s, T_s Q_2\}$. This replacement is valid because $D'(S)$ is a valid lower bound for $D(S)$. Then the capacity inequality with respect to S is defined as follows:

$$\sum_{k \in K_1} \sum_{(i,j) \in \delta^+(S)} x_{i,j,k} \geq \left\lceil \frac{\sum_{i \in V_C} d_i - \sum_{s \in V_S \setminus S} \min\{Q_s, K_s Q_2\}}{Q_1} \right\rceil. \tag{29}$$

3.4. Multistar inequalities

The *multistar and partial multistar inequalities* were firstly introduced by Araque et al. (1990) for the CVRP with unit demands, and then were extended to the CVRP with general demands by Letchford et al. (2002). A multistar is a subgraph of G consisting of two sets of nodes: $N \subseteq V_C$, referred to as *nucleus*, and $S \subseteq \bar{N}$, referred to as *satellites*, and the edges connecting the satellites to the nucleus. A partial multistar is the same as a multistar except that only some of the nucleus nodes (named *connector* nodes, $C \subseteq N$) are connected to the satellites. The general forms of the multistar and the partial multistar inequalities are as follows:

$$\lambda \sum_{(i,j) \in A(N)} y_{i,j} + \sum_{i \in N, j \in S} y_{i,j} \leq \mu \tag{30}$$

and

$$\lambda \sum_{(i,j) \in A(N)} y_{i,j} + \sum_{i \in C, j \in S} y_{i,j} \leq \mu, \tag{31}$$

where λ and μ are constants depending on S, N and C. For the validity proofs of the inequalities and the conditions under which they can induce facets, please see Araque et al. (1990). Note that the word "satellites" has a different meaning in the context of the multistar inequalities. Here, a satellite is a customer in the set S.

3.5. *Extended cover inequalities*

The *minimal cover inequalities* are a family of classic valid inequalities for the knapsack problem (Nauss, 2003). As each satellite and each group of customers in the 2E-CVRPGC can be treated as the knapsack and an item, respectively, we can use these inequalities in the context of the 2E-CVRPGC. For a subset $\mathbb{C}' \subset \mathbb{C}$, if $\sum_{i\in V(\mathbb{C}')} d_i > \max\{Q_s, T_s Q_2\}$, then $\mathbb{C}'$ is a cover for satellite $s \in V_S$. In addition, for each $C \in \mathbb{C}'$, if $\sum_{i\in V(\mathbb{C}')\setminus C} d_i \le \max\{Q_s, T_s Q_2\}$, this cover is minimal. The minimal cover inquality with respect to $\mathbb{C}'$ and s is as follows:

$$\sum_{C\in\mathbb{C}'} v_{C,s} \le |\mathbb{C}'| - 1. \tag{32}$$

According to Balas (1975), the minimal cover inequalities can be lifted. Let $E(\mathbb{C}') := \mathbb{C}' \cup \{C \in \mathbb{C} \setminus \mathbb{C}' : \sum_{i\in C} d_i \ge \sum_{i\in C'} d_i, \forall C' \in \mathbb{C}'\}$. The *extended cover inequality* with respect to $\mathbb{C}'$ is as follows:

$$\sum_{C\in E(\mathbb{C}')} v_{C,s} \le |\mathbb{C}'| - 1. \tag{33}$$

4. Branch-and-cut algorithm

In this section, we describe the branch-and-cut algorithm which relies on the arc-flow model proposed in Section 2 and the valid inequalities introduced in Section 3. The branch-and-cut is a tree search enumeration algorithm in essence, but utilizes a lower bound derived from the LP relaxation of the model to prune nodes that cannot produce better solutions so as to speed up the overall algorithm. In our experiments, we implemented the branch-and-cut algorithm using ILOG CPLEX, which provides callbacks for users to add additional constraints to the model after a node is processed. With the help of the callbacks, the valid inequalities in Section 3 can be added to the arc-flow model dynamically. Note that in our implementation, we use the default CPLEX setting, separate the valid inequalities according to the order in which they were introduced in Section 3 and do not delete the non-binding inequalities. We refer the reader to Padberg and Rinaldi (1987) and Conforti et al. (2014) for more details about branch-and-cut algorithm.

In the following context, we outline the main features of our branch-and-cut algorithm.

4.1. Initial linear relaxation

At the root node of the branch-and-cut tree, we initialize the LP relaxation of the arc-flow model by relaxing the integrality constraints (17), (21), (22) and (23). Hence, the initial LP relaxation consists of the objective function (1), constraints (2)–(16), (18)–(20) and the continuous relaxations of constraints (17), (21), (22) and (23).

4.2. Separation Heuristics

We implemented three heuristics to simultaneously separate the violated capacity inequalities for the customers (24) and the strengthened capacity inequalities for the customers (28). The first two heuristics are the *shrink heuristic* and the *connected component heuristic*, both of which are classical separation heuristics for the capacity inequalities (Ralphs et al., 2003; Lysgaard et al., 2004). The third heuristic is a two-phase tabu search algorithm proposed by (Augerat et al., 1998). Let $\bar{y}$ be the values of variables y in the current optimal LP solution. Before the heuristics are invoked, we first construct a supported graph $G^* = (V^*, A^*)$, where $V^* = V_C$ is the node set and $A^* = \{(i, j) \mid (i, j) \in A_2, i, j \in V_C, \bar{y}_{i,j} > 0\}$ is the arc set. All separation heuristics work on this supported graph.

The shrink heuristic is an iterative greedy approach. In each iteration, two nodes $i, j \in V^*$ with the maximum arc value $\bar{y}_{i,j} + \bar{y}_{j,i}$ are shrank into a super node k. The super node k is then used to replace nodes i and j in graph V^*. After that, the inequalities with respect to the node set represented by super node k is examined. If violated inequalities are found, they are saved and added to a pool of violated inequalities. The iteration process continues until all of the arc values in A^* are smaller than a threshold value of 0.1.

The connected component heuristic is also an iterative greedy approach. The algorithm first computes all of the connected components of graph G^*, and then for each connected component, searches for the violated inequalities iteratively. In each iteration, the inequalities with respect to the connected component are checked. If violated inequalities are found, they are saved and added to a pool of violated inequalities. Then the algorithm removes the node which, if removed, will lead to the least outflow from this component. The iteration process repeats until the outflow from the connected component is smaller than a threshold value of 0.1.

The tabu search heuristic starts with an initial set $S = \{i\}$, where i is from a subset of $|V_C|/2$ randomly selected nodes. Then for each number of vehicles $p = 1, 2, ..., T_2 - 1$, the algorithm searches for the violated inequalities by toggling between two phases, called the *expansion phase* and the *interchange phase*. In the expansion phase, the algorithm iteratively adds a node into set S, until its size satisfies $d(S) > (p + ulimit)Q_2$, where *ulimit* is a given parameter between 0 and 1. At this moment, the algorithm toggles to the interchange phase, where a proportion of nodes in S are interchanged with those from other sets generated by the algorithm before. The number of iterations in the interchange phase is limited by a parameter. After that, the algorithm toggles to the expansion phase again. For each set generated during the iteration process, the inequalities with respect to the set are examined.

For the strengthened capacity inequalities for the customers (25) and the capacity inequalities for the satellites (29), we separate them by full enumeration as the number of the groups and the number of the satellites are limited. For the multistar inequalities (30) and the partial multistar inequalities (31), we used the *polygon procedure* proposed by Letchford et al. (2002) to search for the violated inequalities. For the extended cover inequalities (33), we separate them by the heuristic proposed by Kaparis and Letchford (2010).

5. Computational experiments

This section presents the computational results of the branch-and-cut algorithm on two classes of randomly-generated instances. The algorithm was implemented in Java using callbacks of ILOG CPLEX 12.5.1. All experiments were conducted on a Dell personal computer with an Intel i7-4790 3.60 GHz CPU, 16 GB RAM, and Windows 7 operating system.

5.1. Test instances

The test instances used in the experiments consist of a class of scattered instances and a class of clustered instances. Each class of instances can be further divided into 4 groups and each group has 5 instances. Instances in the same group have the same number of satellites, which can be 2, 3, 4 or 5. Therefore, there are 40 test instances in total. The attributes of each instance are set as follows. The number of customers $|V_C|$ and the number of disjoint groups m are set to $10|V_S|$ and $2|V_S|$, respectively, where $|V_S|$ is the number of satellites in the instance. The capacities of the vehicles in the first echelon and the second echelon, namely Q_1 and Q_2, are set to 30 and 15, respectively. The demand of each customer is randomly generated in $[1, 7]$. The number of available vehicles in the first echelon and the second echelon, namely T_1 and T_2, are set to $\left\lceil \frac{1.5|V_C|d_{avg}}{Q_1} \right\rceil$ and $\left\lceil \frac{1.5|V_C|d_{avg}}{Q_2} \right\rceil$, respectively, where d_{avg} is the average demand of the customers. For each satellite s, the capacity Q_s and the maximum number of available vehicles T_s are set to $\left\lceil \frac{1.2|V_C|d_{avg}}{|V_S|} \right\rceil$ and $\left\lceil \frac{T_2}{|V_S|} \right\rceil$, respectively. The unit handling cost h_s for each satellite s is randomly generated in $[1.0, 2.0]$. Therefore, the satellites in the same instance are identical. The detailed attributes of each instance are summarized in Tables 1 and 2.

Next we describe the procedure to generate the locations of the depot, the satellites and the customers. We limit the coordinates of all the nodes within a square of $[0, 100] \times [0, 100]$. The depot is located in the center of the square, namely $(50, 50)$. The coordinates of the satellites are determined as follows. If $|V_S| = 2$, the coordinates of the two satellites are $(25, 50)$ and $(75, 50)$, respectively; if $|V_S| = 3$, the coordinates of the three satellites are $(50, 25)$, $(25, 75)$ and $(75, 75)$, respectively; if $|V_S| = 4$, the coordinates of the four satellites are $(25, 25)$, $(25, 75)$, $(75, 75)$ and $(75, 25)$, respectively; if $|V_S| = 5$, the coordinates of the five satellites are $(25, 25)$, $(25, 75)$, $(75, 75)$, $(75, 25)$ and $(51, 51)$, respectively. For the class of the scattered instances, the customers are randomly distributed in the square, while for the clustered instances, 10 customers are randomly distributed near each satellite. Figures 2 and 3 illustrate the scatter plot of the depot, the satellites and the customers of instances s-1-b and c-1-b, respectively.

Finally, we divide the area of the square into m parts, and assign the customers in the same part to the same group.

5.2. *Effects of valid inequalities*

We conducted experiments to test the effects of the valid inequalities on the model. Tables 3 and 4 summarize the lower bounds obtained at the root node for the scattered instances and the clustered instances, respectively. Columns *LP* present the lower bound without any default inequality from CPLEX and any user inequality. The next five columns show the lower bound after adding one family of the following valid inequalities: the capacity inequalities for the customers

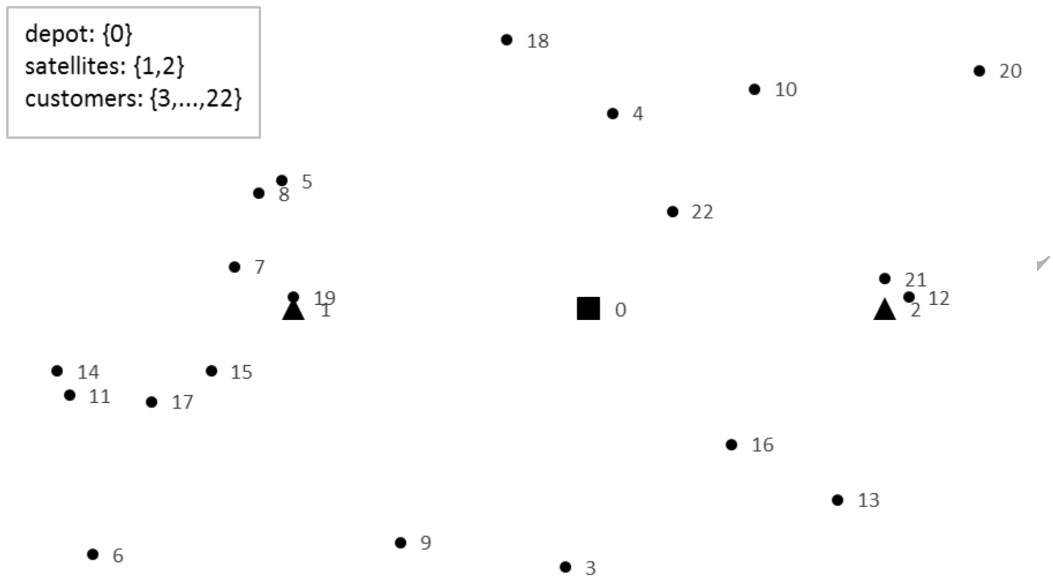

Figure 2: The scatter plot of instance s-1-b.

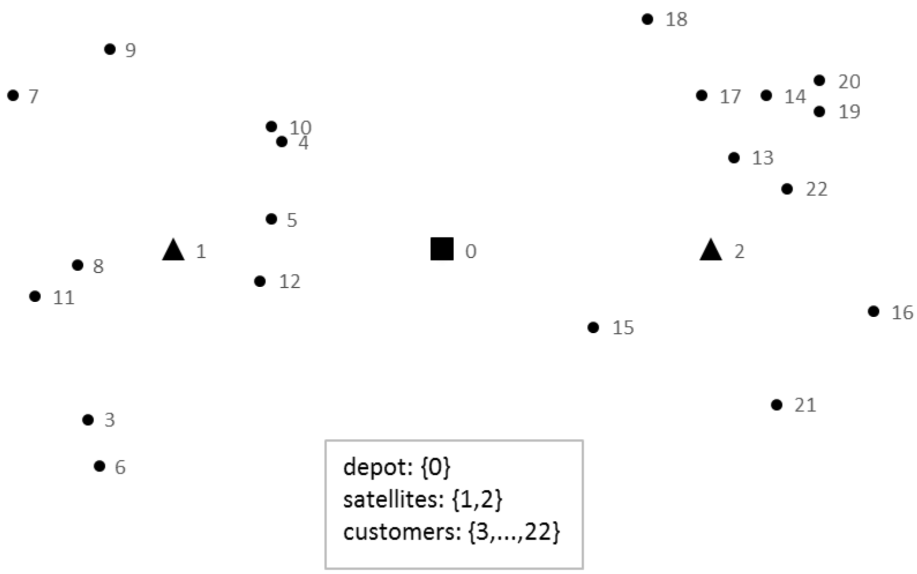

Figure 3: The scatter plot of instance c-1-b.

Table 1: The class of scattered instances.

instance	$\lvert V_S \rvert$	$\lvert V_C \rvert$	m	T_1	T_2	Q_1	Q_2	Q_s	T_s	d_{avg}
s-1-a	2	20	4	4	8	30	15	48	4	4
s-1-b	2	20	4	4	8	30	15	48	4	4
s-1-c	2	20	4	4	8	30	15	48	4	4
s-1-d	2	20	4	4	8	30	15	48	4	4
s-1-e	2	20	4	4	8	30	15	48	4	4
s-2-a	3	30	6	6	12	30	15	48	4	4
s-2-b	3	30	6	6	12	30	15	48	4	4
s-2-c	3	30	6	6	12	30	15	48	4	4
s-2-d	3	30	6	6	12	30	15	48	4	4
s-2-e	3	30	6	6	12	30	15	48	4	4
s-3-a	4	40	8	8	16	30	15	48	4	4
s-3-b	4	40	8	8	16	30	15	48	4	4
s-3-c	4	40	8	8	16	30	15	48	4	4
s-3-d	4	40	8	8	16	30	15	48	4	4
s-3-e	4	40	8	8	16	30	15	48	4	4
s-4-a	5	50	10	10	20	30	15	48	4	4
s-4-b	5	50	10	10	20	30	15	48	4	4
s-4-c	5	50	10	10	20	30	15	48	4	4
s-4-d	5	50	10	10	20	30	15	48	4	4
s-4-e	5	50	10	10	20	30	15	48	4	4

(24) (CIC), the multistar inequalities (30) and (31) (MI), the capacity inequalities for the satellites (29) (CIS), the extended cover inequalities (33) (ECI) and the strengthened capacity inequalities for the customers (25) and (28) ($SCIC$). The lower bounds after adding the five families of inequalities ($CIC, MI, CIS, ECI, SCIC$) are reported in Columns ALL. Columns $CPLEX$ show the lower bounds obtained after adding only the default inequalities from CPLEX. The lower bounds obtained after adding the default inequalities from CPLEX and the five families of inequalities are presented in Columns $FINAL$. The last line "improvement" presents the average percentage improvement of each type of valid inequalities with respect to the lower bound at the root node.

From Tables 3 and 4, we can find that all five families of inequalities can improve the lower bounds at the root node. In terms of average improvement, the strengthened capacity inequalities for the customers achieved the highest improvement on both the scattered instances and the

Table 2: The class of clustered instances.

instance	$\|V_S\|$	$\|V_C\|$	m	T_1	T_2	Q_1	Q_2	Q_s	T_s	d_{avg}
c-1-a	2	20	4	4	8	30	15	48	4	4
c-1-b	2	20	4	4	8	30	15	48	4	4
c-1-c	2	20	4	4	8	30	15	48	4	4
c-1-d	2	20	4	4	8	30	15	48	4	4
c-1-e	2	20	4	4	8	30	15	48	4	4
c-2-a	3	30	6	6	12	30	15	48	4	4
c-2-b	3	30	6	6	12	30	15	48	4	4
c-2-c	3	30	6	6	12	30	15	48	4	4
c-2-d	3	30	6	6	12	30	15	48	4	4
c-2-e	3	30	6	6	12	30	15	48	4	4
c-3-a	4	40	8	8	16	30	15	48	4	4
c-3-b	4	40	8	8	16	30	15	48	4	4
c-3-c	4	40	8	8	16	30	15	48	4	4
c-3-d	4	40	8	8	16	30	15	48	4	4
c-3-e	4	40	8	8	16	30	15	48	4	4
c-4-a	5	50	10	10	20	30	15	48	4	4
c-4-b	5	50	10	10	20	30	15	48	4	4
c-4-c	5	50	10	10	20	30	15	48	4	4
c-4-d	5	50	10	10	20	30	15	48	4	4
c-4-e	5	50	10	10	20	30	15	48	4	4

clustered instances. They improved the lower bounds on average by 8.78% and 6.34% on the two classes of instances, respectively. The second highest improvements were achieved by the capacity inequalities for the customers, and are as much as 8.69% and 6.01% on the two classes of instances, respectively. The capacity inequalities for the satellites on average lift the lower bounds by 1.92% and 3.27% on the scattered instances and the clustered instances, respectively. The average percentage improvement brought by the multistar inequalities on the scattered instances are about 5 times of that on the clustered instances (1.87% vs 0.37%), which suggests that the performance of the multistar inequalities is heavily influenced by the distribution of the customers. Although the average improvements brought by the extended cover inequalities are marginal (0.01% and 1.05% for the scattered instances and the clustered instances, respectively), on certain instances these inequalities can be very effective. For example, on instance c-2-c, the

Table 3: Effects of the valid inequalities on the scattered instances.

instance	LP	CIC	MI	CIS	ECI	SCIC	ALL	CPLEX	FINAL
s-1-a	788.05	904.15	788.05	843.05	788.05	898.15	931.65	892.10	940.07
s-1-b	766.57	846.17	774.67	814.90	766.57	846.17	894.50	866.75	914.27
s-1-c	865.77	981.62	865.77	922.43	865.77	985.83	1014.65	963.06	1014.98
s-1-d	791.87	886.59	801.61	855.20	791.87	885.80	949.13	935.76	965.36
s-1-e	753.77	863.90	758.64	753.77	753.77	858.49	864.52	824.12	891.08
s-2-a	1042.38	1148.84	1117.43	1044.38	1042.38	1150.93	1152.38	1145.06	1212.52
s-2-b	1033.08	1123.48	1066.31	1033.68	1033.10	1130.46	1130.53	1074.75	1139.19
s-2-c	1023.29	1085.10	1062.38	1025.69	1023.29	1085.10	1086.06	1076.91	1120.09
s-2-d	974.49	1088.50	975.81	980.09	974.49	1089.02	1093.48	1132.02	1191.63
s-2-e	899.23	981.66	922.94	904.83	899.31	982.75	988.06	1039.22	1094.12
s-3-a	1374.29	1455.18	1374.52	1523.62	1374.29	1456.70	1606.93	1439.85	1629.31
s-3-b	1390.09	1492.10	1449.02	1390.49	1392.02	1492.70	1492.63	1446.43	1508.24
s-3-c	1192.86	1272.17	1226.15	1196.19	1193.20	1281.37	1286.31	1275.27	1332.51
s-3-d	1203.47	1271.88	1206.76	1208.07	1203.47	1272.97	1277.40	1233.21	1282.33
s-3-e	1251.87	1323.14	1260.82	1255.77	1251.87	1323.45	1328.07	1305.55	1347.93
s-4-a	1318.44	1424.86	1318.44	1330.44	1318.44	1425.30	1436.80	1381.84	1450.60
s-4-b	1298.39	1416.64	1334.05	1304.39	1298.39	1418.07	1423.78	1362.44	1433.10
s-4-c	1406.41	1515.21	1467.91	1409.21	1406.41	1517.64	1520.81	1448.19	1528.29
s-4-d	1492.04	1593.54	1504.54	1492.15	1492.04	1593.75	1593.95	1529.12	1610.65
s-4-e	1354.10	1476.03	1360.05	1357.74	1354.10	1476.36	1482.00	1412.34	1498.16
average	1111.02	1207.54	1131.79	1132.30	1111.14	1208.55	1227.68	1189.20	1255.22
improvement		8.69%	1.87%	1.92%	0.01%	8.78%	10.50%	7.04%	12.98%

Table 4: Effects of the valid inequalities on the clustered instances.

instance	LP	CIC	MI	CIS	ECI	SCIC	ALL	CPLEX	FINAL
c-1-a	480.83	552.53	481.63	547.50	480.83	554.10	588.90	566.09	588.90
c-1-b	416.87	450.63	417.07	420.37	419.00	451.06	450.94	470.63	485.95
c-1-c	538.77	573.37	540.19	587.10	538.77	576.41	624.74	597.82	621.25
c-1-d	340.24	363.60	341.27	347.44	360.46	360.90	368.11	361.26	374.78
c-1-e	442.90	470.47	442.90	511.23	442.90	471.47	539.80	525.18	545.30
c-2-a	717.84	757.90	718.07	720.84	717.89	757.90	760.98	762.37	786.71
c-2-b	641.07	684.32	641.07	644.92	641.07	684.96	687.70	658.38	690.14
c-2-c	655.45	706.77	655.57	661.82	734.47	705.09	711.48	694.37	720.50
c-2-d	744.97	785.03	744.97	747.37	745.53	785.68	788.33	776.94	800.52
c-2-e	623.88	661.44	628.50	627.39	628.36	661.44	663.75	669.84	686.15
c-3-a	914.68	951.53	918.00	915.57	914.68	955.28	955.53	1013.53	1042.42
c-3-b	950.20	1010.96	957.96	953.80	950.20	1023.89	1026.15	979.99	1032.16
c-3-c	1039.84	1099.11	1041.60	1041.40	1039.95	1117.48	1118.25	1111.59	1156.41
c-3-d	1038.38	1078.04	1054.95	1040.48	1038.38	1078.12	1079.17	1173.02	1200.16
c-3-e	1139.64	1186.08	1144.16	1274.97	1139.79	1190.24	1325.70	1172.42	1358.41
c-4-a	837.84	875.13	841.74	837.84	883.68	875.87	875.87	861.59	886.38
c-4-b	967.08	1034.12	967.60	970.05	967.24	1036.58	1038.67	998.58	1046.33
c-4-c	974.59	1041.76	980.67	982.13	975.25	1045.71	1051.56	1017.28	1067.94
c-4-d	1145.71	1207.72	1146.76	1281.84	1153.90	1207.72	1343.85	1200.20	1379.04
c-4-e	820.33	868.40	822.81	821.77	820.33	869.56	871.90	841.60	873.36
average	771.55	817.95	774.37	796.79	779.63	820.47	843.57	822.63	867.14
improvement		6.01%	0.37%	3.27%	1.05%	6.34%	9.33%	6.62%	12.39%

extended cover inequalities lifted the lower bound from 655.45 to 734.47.

After adding all five families of valid inequalities, the lower bounds were significantly improved, from 1111.02 to 1227.68 for the scattered instances and from 771.55 to 843.57 for the clustered instances on average. If the default inequalities generated by CPLEX are included, the lower bounds are further lifted to 1255.22 and 867.14. On the contrary, if only the default inequalities generated by CPLEX are applied, the lower bounds can only be lifted to 1189.20 and 822.63. From this aspect, the average percentage improvements brought by the five families of valid inequalities are 10.50% for the scattered instances and 9.33% for the clustered instances.

5.3. Integer solution results

Tables 5 and 6 report the integer solution results of CPLEX and the branch-and-cut algorithm on the scattered instances and the clustered instances, respectively. The time limits for both algorithms are set to 7200 seconds. Note that the valid inequalities used in the branch-and-cut algorithm contain the five families of the inequalities as well as the default inequalities generated by CPLEX. Columns *LB* present the best lower bound found by the algorithms for each instance within the 7200s computational time. If a lower bound is proved to be optimal, it is marked in bold. The number of nodes explored by the algorithms are reported in Columns *Node*. The running time in seconds, either to find an optimal solution or to terminate due to the time limit, is presented in Columns *Time*. Columns *Sepa* show the time spent by the separation heuristics. The number of valid inequalities separated is reported in Columns *User*. The best upper bounds achieved by both CPLEX and the branch-and-cut algorithm are shown in Columns *UB*.

From Tables 5 and 6, we can see that the branch-and-cut algorithm can solve 14 out of 20 scattered instances and 18 out of 20 clustered instances to optimality, while CPLEX can only solve 7 scattered instances and 8 clustered instances to optimality. For all the instances that can be solved to optimality, the branch-and-cut algorithm explores fewer nodes and takes less time to find an optimal solution. Meanwhile, in the instances which can not be solved to optimality, the best lower bounds yielded by the branch-and-cut algorithm are much better than those yielded by CPLEX. The stronger lower bounds lifted by the five families of valid inequalities contribute to the superiority of the branch-and-cut algorithm.

Table 5: Comparisons with CPLEX for the set of scattered instances.

instance	CPLEX			Branch-and-cut					UB
	LB	Node	Time(s)	LB	Node	Time(s)	Sepa	User	
s-1-a	**1029.80**	224,952	123.91	**1029.80**	3	0.78	0.11	257	1029.80
s-1-b	**988.00**	31,016	19.73	**988.00**	0	0.53	0.17	284	988.00
s-1-c	**1024.60**	121,160	91.73	**1024.60**	0	0.63	0.00	277	1024.60
s-1-d	**1068.70**	525,084	333.59	**1068.70**	1,949	13.88	13.45	1,024	1068.70
s-1-e	**1019.30**	1,081,611	646.69	**1019.30**	47	1.97	1.45	1,335	1019.30
s-2-a	1506.40	7,565,000	7200.00	**1522.40**	154	31.97	29.52	6,408	1522.40
s-2-b	1390.77	7,853,658	7200.00	**1402.00**	1,831	72.50	70.86	6,614	1402.00
s-2-c	**1240.80**	3,039,771	3367.17	**1240.80**	1,760	119.00	116.24	7,570	1240.80
s-2-d	1316.35	5,961,478	7200.00	**1368.90**	38,643	2753.95	2752.11	17,708	1368.90
s-2-e	**1143.20**	4,491,983	5564.04	**1143.20**	9	4.14	2.55	1,374	1143.20
s-3-a	1864.04	5,003,302	7200.00	**1908.00**	274	282.13	277.72	21,807	1908.00
s-3-b	1737.22	4,626,885	7200.00	1809.77	47,688	7200.00	7193.72	4,001	1888.90
s-3-c	1483.41	4,239,400	7200.00	**1505.40**	9,634	3585.74	3576.67	45,267	1505.40
s-3-d	1437.51	3,857,398	7200.00	**1448.20**	4,860	1539.84	1533.80	40,709	1448.20
s-3-e	1579.21	4,508,592	7200.00	**1607.90**	1,021	1326.80	1319.75	61,219	1607.90
s-4-a	1672.82	2,927,220	7200.00	1739.65	3,927	7200.00	7192.27	96,053	1789.70
s-4-b	1541.80	1,434,665	7200.00	1561.18	10,123	7200.00	7187.56	5,381	1776.20
s-4-c	1702.15	3,199,333	7200.00	1767.57	17,723	7200.00	7187.30	2,229	1806.70
s-4-d	1767.06	1,576,005	7200.00	1781.90	7,091	7200.00	7180.61	4,391	1971.40
s-4-e	1731.21	2,926,163	7200.00	1776.86	2,821	7200.00	7191.75	105,386	1897.60
average	1412.22	3,259,734	5187.34	1435.71	7,478	2646.69	2641.38	21,465	

Table 6: Comparisons with CPLEX for the set of clustered instances.

instance	CPLEX			Branch-and-cut					UB
	LB	Node	Time(s)	LB	Node	Time(s)	Sepa	User	
c-1-a	**588.90**	157,974	79.73	**588.90**	0	0.25	0.00	125	588.90
c-1-b	518.64	11,373,750	7200.00	**536.80**	9	0.74	0.47	477	536.80
c-1-c	**625.70**	349,803	197.97	**625.70**	0	0.31	0.02	179	625.70
c-1-d	394.43	15,205,340	7200.00	**411.40**	4,282	19.33	18.98	560	411.40
c-1-e	**545.40**	52,767	28.00	**545.40**	0	0.38	0.00	92	545.40
c-2-a	**935.10**	2,457,806	2168.95	**935.10**	504	9.06	7.99	781	935.10
c-2-b	**827.00**	3,807,758	4066.56	**827.00**	4,536	78.11	76.89	475	827.00
c-2-c	**856.70**	4,657,942	4070.53	**856.70**	30,733	4477.06	4476.03	39,051	856.70
c-2-d	931.79	9,151,921	7200.00	**932.80**	863	14.86	14.08	503	932.80
c-2-e	**762.50**	42,055	45.06	**762.50**	6	1.84	0.53	284	762.50
c-3-a	**1049.40**	2,560,261	6534.20	**1049.40**	127	14.00	9.19	85	1049.40
c-3-b	1203.48	4,021,667	7200.00	**1238.10**	8,515	2771.27	2768.39	42,628	1238.10
c-3-c	1346.38	2,854,433	7200.00	**1385.60**	6,359	980.86	977.41	18,721	1385.60
c-3-d	1599.75	2,333,631	7200.00	**1653.70**	4,935	1556.17	1551.25	3,145	1653.70
c-3-e	1747.92	3,302,902	7200.00	**1787.60**	69	72.97	69.83	6,403	1787.60
c-4-a	1003.57	3,434,800	7200.00	**1018.60**	15,067	3468.31	3462.76	20,800	1018.60
c-4-b	1173.18	1,675,620	7200.00	**1244.10**	15,585	6461.55	6453.22	53,887	1244.10
c-4-c	1265.08	2,288,400	7200.00	1323.37	14,485	7200.00	7191.81	56,140	1325.60
c-4-d	1596.43	3,585,600	7200.00	**1670.80**	3,101	805.17	799.63	11,022	1670.80
c-4-e	897.97	521,296	7200.00	983.60	6,393	7200.00	7192.86	84,046	1016.10
average	993.47	3,691,786	5179.55	1018.86	5,778	1756.61	1753.57	16,970	

Although the branch-and-cut algorithm outperforms CPLEX, the size of the instances that can be optimally solved by the branch-and-cut algorithm is still limited. The branch-and-cut algorithm can only solve medium-size instances to optimality.

5.4. Comparison with the classical 2E-CVRP

Table 7: Comparisons with the classical 2E-CVRP for the set of scattered instances.

instance	2E-CVRP					2E-CVRPGC					Gap
	LB	Node	Time(s)	Sepa	User	LB	Node	Time(s)	Sepa	User	
s-1-a	**599.00**	452	21.88	20.80	5,522	**1029.80**	3	0.78	0.11	257	41.83%
s-1-b	**563.00**	1,688	139.81	138.27	18,156	**988.00**	0	0.53	0.17	284	43.02%
s-1-c	**612.00**	64	3.06	2.13	1,511	**1024.60**	0	0.63	0.00	277	40.27%
s-1-d	**643.00**	4,047	252.92	251.88	19,641	**1068.70**	1,949	13.88	13.45	1,024	39.83%
s-1-e	**622.00**	53	2.16	1.39	1,393	**1019.30**	47	1.97	1.45	1,335	38.98%
s-2-a	672.12	3,958	7200.00	7196.42	190,409	**1522.40**	154	31.97	29.52	6,408	55.85%
s-2-b	699.28	4,059	7200.00	7196.22	201,551	**1402.00**	1,831	72.50	70.86	6,614	50.12%
s-2-c	**683.00**	17	4.24	1.89	1,322	**1240.80**	1,760	119.00	116.24	7,570	44.95%
s-2-d	665.96	4,265	7200.00	7196.20	187,904	**1368.90**	38,643	2753.95	2752.11	17,708	51.35%
s-2-e	**589.00**	652	119.70	116.51	15,892	**1143.20**	9	4.14	2.55	1,374	48.48%
s-3-a	773.26	1,732	7200.00	7189.31	225,238	**1908.00**	274	282.13	277.72	21,807	59.47%
s-3-b	809.92	1,840	7200.00	7190.58	202,003	1809.77	47,688	7200.00	7193.72	4,001	55.25%
s-3-c	706.90	1,955	7200.00	7186.86	197,148	**1505.40**	9,634	3585.74	3576.67	45,267	53.04%
s-3-d	698.65	2,781	7200.00	7192.09	182,277	**1448.20**	4,860	1539.84	1533.80	40,709	51.76%
s-3-e	696.93	1,714	7200.00	7187.89	230,908	**1607.90**	1,021	1326.80	1319.75	61,219	56.66%
s-4-a	779.42	885	7200.00	7175.14	173,849	1739.65	3,927	7200.00	7192.27	96,053	55.20%
s-4-b	785.18	1,639	7200.00	7181.42	172,146	1561.18	10,123	7200.00	7187.56	5,381	49.71%
s-4-c	806.35	816	7200.00	7179.72	164,435	1767.57	17,723	7200.00	7187.30	2,229	54.38%
s-4-d	801.49	992	7200.00	7173.03	174,208	1781.90	7,091	7200.00	7180.61	4,391	55.02%
s-4-e	775.77	1,252	7200.00	7173.39	193,151	1776.86	2,821	7200.00	7191.75	105,386	56.34%
average	699.11	1,743	4707.19	4,697.56	127,933	1435.71	7,478	2646.69	2641.38	21,465	51.31%

The 2E-CVRP can be viewed as a relaxation of the 2E-CVRPGC as a 2E-CVRPGC instance can be transformed into a 2E-CVRP instance by dropping the grouping constraints. In this section, we conduct experiments to analyze the influence of the grouping constraints by comparing the optimal solutions of the 2E-CVRPGC and those of the 2E-CVRP. In order to get the 2E-CVRP

Table 8: Comparisons with the classical 2E-CVRP for the set of clustered instances.

instance	2E-CVRP					2E-CVRPGC					Gap
	LB	Node	Time(s)	Sepa	User	LB	Node	Time(s)	Sepa	User	
c-1-a	**249.00**	0	0.34	0.00	143	**588.90**	0	0.25	0.00	125	57.72%
c-1-b	**225.00**	0	0.20	0.00	105	**536.80**	9	0.74	0.47	477	58.08%
c-1-c	**281.00**	0	0.34	0.00	251	**625.70**	0	0.31	0.02	179	55.09%
c-1-d	**172.00**	0	0.41	0.08	146	**411.40**	4,282	19.33	18.98	560	58.19%
c-1-e	**236.00**	0	0.22	0.00	98	**545.40**	0	0.38	0.00	92	56.73%
c-2-a	**343.00**	0	0.64	0.00	202	**935.10**	504	9.06	7.99	781	63.32%
c-2-b	**329.00**	8	1.64	0.48	415	**827.00**	4,536	78.11	76.89	475	60.22%
c-2-c	**353.00**	0	1.02	0.00	553	**856.70**	30,733	4477.06	4476.03	39,051	58.80%
c-2-d	**383.00**	0	1.11	0.00	579	**932.80**	863	14.86	14.08	503	58.94%
c-2-e	**311.00**	0	0.77	0.00	90	**762.50**	6	1.84	0.53	284	59.21%
c-3-a	**430.00**	0	1.89	0.00	291	**1049.40**	127	14.00	9.19	85	59.02%
c-3-b	**463.00**	19	6.75	3.09	1,116	**1238.10**	8,515	2771.27	2768.39	42,628	62.60%
c-3-c	**496.00**	6	5.98	1.23	546	**1385.60**	6,359	980.86	977.41	18,721	64.20%
c-3-d	**454.00**	0	3.69	0.00	976	**1653.70**	4,935	1556.17	1551.25	3,145	72.55%
c-3-e	478.33	11,395	7200.00	7196.48	91,966	**1787.60**	69	72.97	69.83	6,403	73.24%
c-4-a	**338.00**	17	17.75	10.33	1,513	**1018.60**	15,067	3468.31	3462.76	20,800	66.82%
c-4-b	**424.00**	1,072	508.45	497.69	16,422	**1244.10**	15,585	6461.55	6453.22	53,887	65.92%
c-4-c	**404.00**	27	15.86	9.31	832	1323.37	14,485	7200.00	7191.81	56,140	69.47%
c-4-d	**438.00**	22	12.08	6.39	731	**1670.80**	3,101	805.17	799.63	11,022	73.79%
c-4-e	339.96	6,011	7200.00	7190.45	82,835	983.60	6,393	7200.00	7192.86	84,046	65.44%
average	357.36	929	748.96	745.78	9,991	1018.86	5,778	1756.61	1753.57	16,970	64.92%

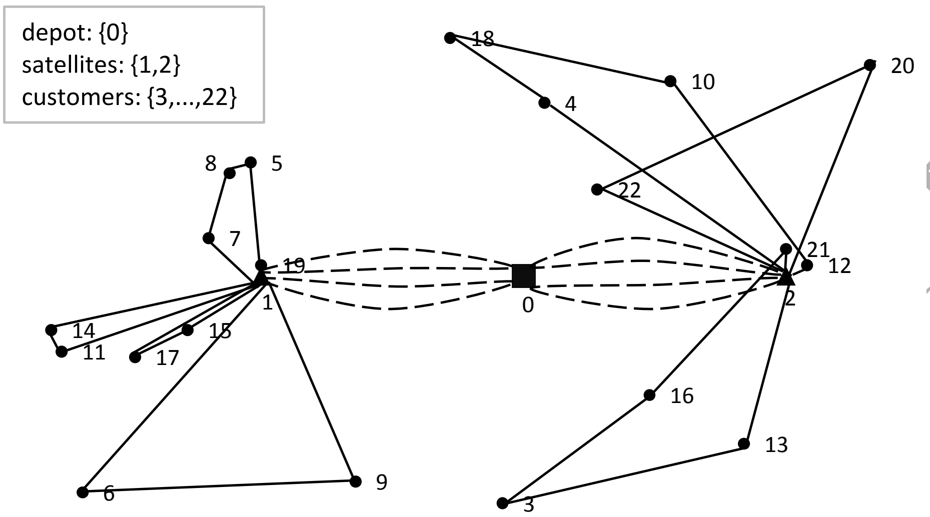

Figure 4: The routing plan of the optimal 2E-CVRP solution of instance s-1-b.

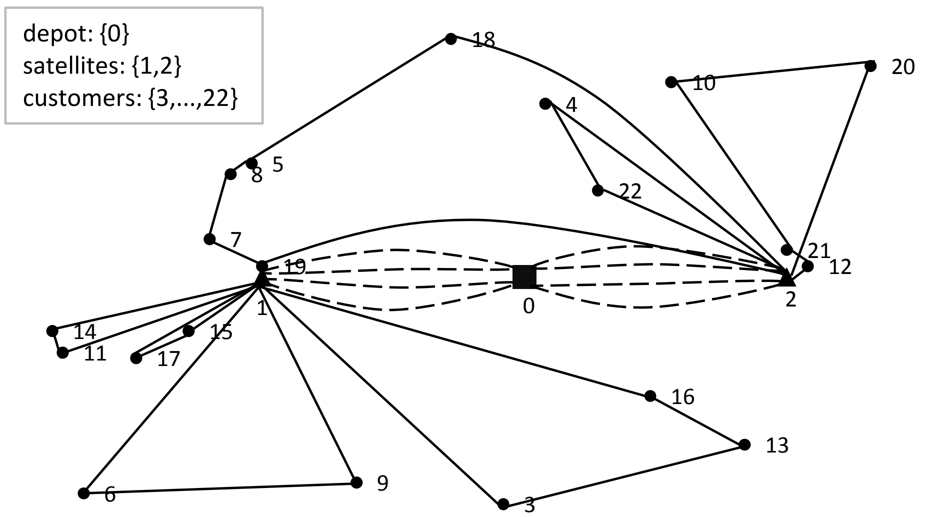

Figure 5: The routing plan of the optimal 2E-CVRPGC solution of instance s-1-b.

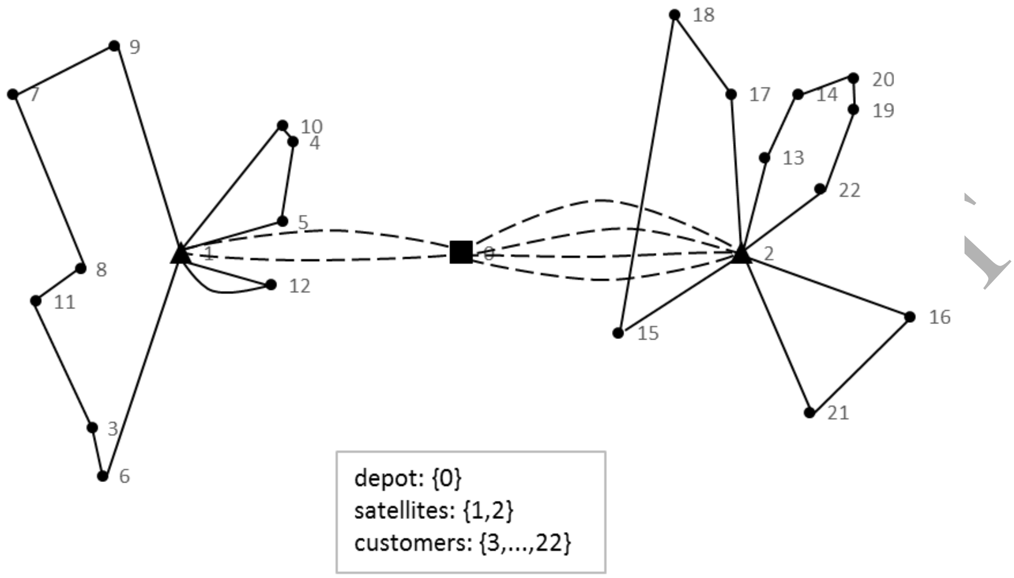

Figure 6: The routing plan of the optimal 2E-CVRP solution of instance c-1-b.

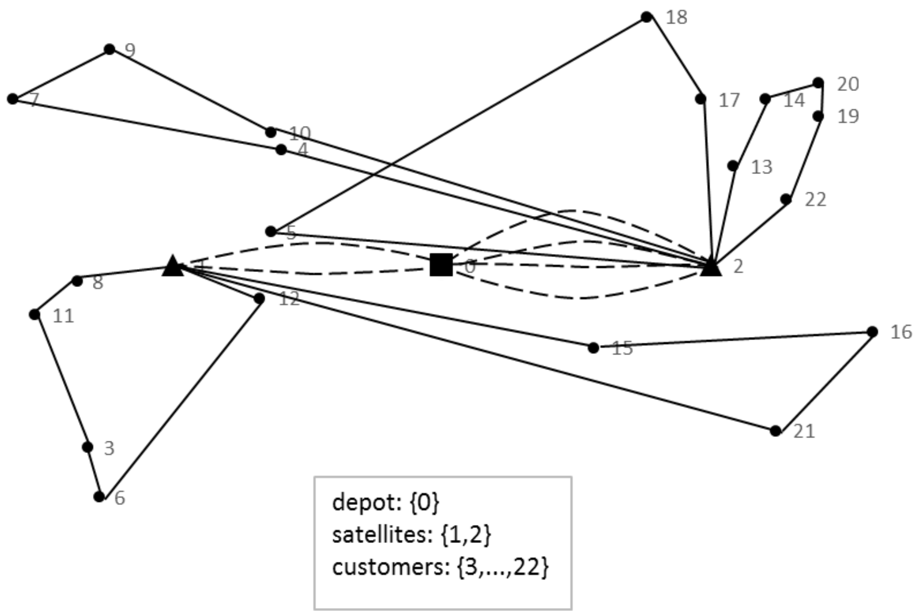

Figure 7: The routing plan of the optimal 2E-CVRPGC solution of instance c-1-b.

optimal solutions, we run the branch-and-cut algorithm but ignore constraints (15), the strengthened capacity inequalities for the customers (25, 28) and the extended cover inequalities (33).

Tables 7 and 8 summarize the computational results. Columns *2E-CVRP* and *2E-CVRPGC* report the computational results of the 2E-CVRP and the 2E-CVRPGC, respectively. The last columns *Gap* give the percentage gaps of the best lower bounds between the 2E-CVRP and the 2E-CVRPGC. Note that if a lower bound is marked in bold, it means this lower bound is indeed an optimal solution.

From Tables 7 and 8, we can find that in terms of solution cost, the differences of the optimal solutions between the 2E-CVRP and 2E-CVRPGC are quite large. This conclusion is supported by the fact that the percentage gaps of the optimal solutions between the 2E-CVRP and the 2E-CVRPGC are all more than 38% among the scattered instances and 55% among the clustered instances. To illustrate the difference of the optimal solutions between the 2E-CVRP and the 2E-CVRPGC, Figures 4 and 5 present the detailed routing plans of optimal solutions of the 2E-CVRP and 2E-CVRPGC in instance s-1-b, respectively, and Figures 6 and 7 present the detailed routing plans of optimal solutions of the 2E-CVRP and 2E-CVRPGC in instance c-1-b, respectively. From these figures, we can see that customers in optimal solutions of the 2E-CVRP are always served by vehicles from the satellites nearby. However, customers in optimal solutions of the 2E-CVRPGC may have to be served by vehicles from the satellites far from them due to the grouping constraints, which results in long vehicle routes and hence high solution cost.

6. Conclusions

In this paper, we investigate the two-echelon capacitated vehicle routing problem with grouping constraints (2E-CVRPGC), where customers are divided into several different groups and customers from the same group must be served by vehicles from the same satellite. The 2E-CVRPGC extends the classic two-echelon capacitated vehicle routing problem (2E-CVRP) and has applications in practice. We introduce a mathematical model to formulate the problem and propose five families of valid inequalities to strengthen the model. Based on the model and the valid inequalities, we propose a branch-and-cut algorithm to tackle the problem. To test the proposed algorithm, we generate a class of scattered instances and a class of clustered instances. Computational results on the two classes of instances show that all of the five families of valid inequalities can improve the lower bounds yielded by the LP relaxation of the model, and the proposed branch-and-cut algorithm can successfully solve 32 out of 40 instances to optimality with a reasonable computation time. To analyze the impact of the grouping constraints, we conduct additional experiments to compare the difference of the optimal solutions between the 2E-CVRP and the 2E-CVRPGC on the same instances. Computational results show that the detailed routing plans of the optimal solutions of the 2E-CVRP and the 2E-CVRPGC are totally different.

Our future research on the 2E-CVRPGC will focus on the design of more powerful exact algorithm. Branch-and-price or other algorithms based on column generation are a class of the most successful exact algorithms to solve many routing problems, including the 2E-CVRP (Baldacci et al., 2013). Therefore, the design of our future exact algorithm will also be based on column generation. In addition, we presented simulation results to our clients and explained the great impacts of the grouping constraints on the solution costs. In the future, we will work on together to partially relax the grouping constraints to achieve a better balance between logistic costs and management complexity.

Acknowledgements

This work was partially supported by grants from the National Natural Science Foundation of China (Grant No. 71501091, 71531009, 71571077, and 71501075), Guangdong Natural Science Funds for Distinguished Young Scholar (Grant No. 2015A030306007), NRF Singapore (Grant No. NRF-RSS2016-004), and Singapore MOE-AcRF-Tier 1 (Grant No. R-266-000-096-133 and R-266-000-096-731).

References

Araque, J. R., Hall, L. A., Magnanti, T. L., 1990. Capacitated trees, capacitated routing, and associated polyhedra.

Augerat, P., Belenguer, J. M., Benavent, E., Corbéran, A., Naddef, D., 1998. Separating capacity constraints in the cvrp using tabu search. European Journal of Operational Research 106 (2), 546–557.

Balas, E., 1975. Facets of the knapsack polytope. Mathematical programming 8 (1), 146–164.

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

Baldacci, R., Mingozzi, A., Wolfler Calvo, R., 2011. An exact method for the capacitated location-routing problem. Operations Research 59 (5), 1284–1296.

Conforti, M., Cornuéjols, G., Zambelli, G., 2014. Integer programming models. In: Integer Programming. Springer, pp. 45–84.

Crainic, T. G., Gendreau, M., Potvin, J. Y., 2009. Intelligent freight-transportation systems: Assessment and the contribution of operations research. Transportation Research Part C: Emerging Technologies 17 (6), 541–557.

Crainic, T. G., Mancini, S., Perboli, G., Tadei, R., 2008. Clustering-based heuristics for the two-echelon vehicle routing problem. CIRRELT, Montréal, CIRRELT-2008-46 http://www. cirrelt. ca/DocumentsTravail/CIRRELT-2008-46. pdf.

Crainic, T. G., Mancini, S., Perboli, G., Tadei, R., 2011. Multi-start heuristics for the two-echelon vehicle routing problem. Springer.

Cuda, R., Guastaroba, G., Speranza, M., 2015. A survey on two-echelon routing problems. doi: http://dx.doi.org/10.1016/j.cor.2014.06.008. URL http://www.sciencedirect.com/science/article/pii/S030505481400166X. 55, 185–199.

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

Feliu, J. G., Perboli, G., Tadei, R., Vigo, D., 2008. The two-echelon capacitated vehicle routing problem. General Information 45 (3), 364–380.

Fukasawa, R., Longo, H., Lysgaard, J., Aragão, M. P. d., Reis, M., Uchoa, E., Werneck, R. F., 2006. Robust branch-and-cut-and-price for the capacitated vehicle routing problem. Mathematical Programming 106 (3), 491–511.

Gavish, B., 1984. The delivery problem: New cutting planes procedure. TIMS XXVI conference, Copenhagen.

Grangier, P., Gendreau, M., Lehud, F., Rousseau, L. M., 2016. An adaptive large neighborhood search for the two-echelon multiple-trip vehicle routing problem with satellite synchronization. European Journal of Operational Research 254 (1), 80–91.

Hemmelmayr, V. C., Cordeau, J. F., Crainic, T. G., 2012. An adaptive large neighborhood search heuristic for two-echelon vehicle routing problems arising in city logistics. Computers & Operations Research 39 (12), 3215–3228.

Jepsen, M., Spoorendonk, S., 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., Letchford, A., 2010. Separation algorithms for 0-1 knapsack polytopes. Mathematical programming 124 (1), 69–91.

Laporte, G., Nobert, Y., 1983. A branch and bound algorithm for the capacitated vehicle routing problem. OR Spectrum 5 (2), 77–85.

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

Lysgaard, J., Letchford, A. N., Eglese, R. W., 2004. A new branch-and-cut algorithm for the capacitated vehicle routing problem. Mathematical Programming 100 (2), 423–445.

Miller, C., Tucker, A., Zemlin, R., 1960. Integer programming formulations and travelling salesman problems. J. Ass. Comp. Mach. 7, 326–329.

Nadizadeh, A., Nasab, H. H., 2014. Solving the dynamic capacitated location-routing problem with fuzzy demands by hybrid heuristic algorithm. European Journal of Operational Research 238 (2), 458–470.

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

Nguyen, V. P., Prins, C., Prodhon, C., 2006. Solving the two-echelon location routing problem by a grasp reinforced by a learning process and path relinking. 4OR 216 (3), 113–126.

Padberg, M., Rinaldi, G., 1987. Optimization of a 532-city symmetric traveling salesman problem by branch and cut. Operations Research Letters 6 (1), 1–7.

Perboli, G., Tadei, R., Masoero, F., 2009a. Models and cuts for the two-echelon vehicle routing problem. In: Proceedings of the International Network Optimization Conference. Vol. 2009.

Perboli, G., Tadei, R., Masoero, F., 2009b. Valid inequalities for the two-echelon capacitated vehicle routing problem. CIRRELT.

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

Ralphs, T. K., Kopman, L., Pulleyblank, W. R., Trotter, L. E., 2003. On the capacitated vehicle routing problem. Mathematical Programming 94 (2-3), 343–359.

Schiffer, M., Walther, G., 2017. The electric location routing problem with time windows and partial recharging. European Journal of Operational Research.

Taniguchi, E., Thompson, R. G., Yamada, T., Van Duin, R., 2001. City Logistics. Network modelling and intelligent transport systems.

Toth, P., Vigo, D., 2014. Vehicle routing: problems, methods, and applications. Vol. 18. Siam.

Zhao, Q. H., Chen, S., Zang, C. X., 2008. Model and algorithm for inventory/routing decision in a three-echelon logistics system . European Journal of Operational Research 191 (3), 623–635.