Petris, Matteo; Archetti, Claudia; Cattaruzza, Diego; Ogier, Maxime; Semet, Frédéric

Article

A Branch-Price-and-Cut algorithm for the Multi-Commodity two-echelon Distribution Problem

EURO Journal on Transportation and Logistics (EJTL)

Provided in Cooperation with:
Association of European Operational Research Societies (EURO), Fribourg

Suggested Citation: Petris, Matteo; Archetti, Claudia; Cattaruzza, Diego; Ogier, Maxime; Semet, Frédéric (2024) : A Branch-Price-and-Cut algorithm for the Multi-Commodity two-echelon Distribution Problem, EURO Journal on Transportation and Logistics (EJTL), ISSN 2192-4384, Elsevier, Amsterdam, Vol. 13, Iss. 1, pp. 1-13, https://doi.org/10.1016/j.ejtl.2024.100139

EURO Journal on Transportation and Logistics 13 (2024) 100139

Contents lists available at ScienceDirect

EURO Journal on Transportation and Logistics

journal homepage: www.elsevier.com/locate/ejtl

A Branch-Price-and-Cut algorithm for the Multi-Commodity two-echelon Distribution Problem

Matteo Petris [a,*], Claudia Archetti [b], Diego Cattaruzza [c], Maxime Ogier [c], Frédéric Semet [c]

[a] Univ. Lille, Inria, CNRS, Centrale Lille, UMR 9189 CRIStAL, F-59000 Lille, France
[b] Department of Information Systems, Decision Sciences and Statistics, ESSEC Business School, Cergy-Pontoise, France
[c] Univ. Lille, CNRS, Inria, Centrale Lille, UMR 9189 CRIStAL, F-59000 Lille, France

ARTICLE INFO

Keywords:
Two echelon routing problems
Multiple commodities
Split delivery
Branch-Price-and-Cut

ABSTRACT

In the Multi-Commodity two-echelon Distribution Problem (MC2DP), multiple commodities are distributed in a two-echelon distribution system involving suppliers, distribution centres and customers. Each supplier may provide different commodities and each customer may request several commodities as well. In the first echelon, capacitated vehicles perform direct trips to transport the commodities from the suppliers to the distribution centres for consolidation purposes. In the second echelon, each distribution centre owns a fleet of capacitated vehicles to deliver the commodities to the customers through multi-stop routes. Commodities are compatible, i.e., they can be mixed in the vehicles. Finally, customer requests can be split by commodities, that is, a customer can be visited by several vehicles, but the total amount of each commodity has to be delivered by a single vehicle. The aim of the MC2DP is to minimize the total transportation cost to satisfy customer demands.

We propose a set covering formulation for the MC2DP where the exponential number of variables relates to the routes in the delivery echelon. We develop a Branch-Price-and-Cut algorithm (BPC) to solve the problem. The pricing problem results in solving an Elementary Shortest Path Problem with Resource Constraints (ESPPRC) per distribution centre. We tackle the ESPPRC with a label setting dynamic programming algorithm which incorporates ng-path relaxation and a bidirectional labelling search. Pricing heuristics are invoked to speed up the procedure. In addition, the formulation is strengthened by integrating capacity cuts and two families of valid inequalities specific for the multiple commodities aspect of the problem.

Our approach solves to optimality 439 over the 736 benchmark instances from the literature. The optimality gap of the unsolved instances is 2.1%, on average.

1. Introduction

In a two-echelon distribution system, goods are transferred from origins (depots, suppliers) to destinations (customers) via intermediate facilities (satellites, distribution centres) (see Guastaroba et al., 2016). In the *collection echelon*, large vehicles bring goods from the origins to the intermediate facilities where consolidation operations are performed. Whereas, in the *delivery echelon*, smaller vehicles are in charge of distributing the goods to the final customers. Routing decisions are usually required in both echelons. Two-echelon systems take advantage of consolidating goods at intermediate facilities and using different fleets within each echelon to reduce overall transportation costs. An example of this delivery strategy can be encountered in city logistics (Cattaruzza et al., 2017; Crainic et al., 2023) where the aim is also to grant the access in urban areas only to environmental-friendly vehicles that usually have a small capacity.

In this article, we consider the two-echelon distribution problem introduced in Gu et al. (2022), namely the *Multi-Commodity two-echelon Distribution Problem* (MC2DP). In this context, origins, intermediate facilities and destinations are referred to as *suppliers*, *distribution centres* and *customers*, respectively. There are few vehicle routing problems which explicitly deal with multiple commodities within a two-echelon distribution system. To the best of our knowledge, among these problems, the MC2DP is the only one considering a *many-to-many* setting. In fact, in the MC2DP, the commodity requested by a customer is not pre-assigned to a specific supplier, so it can be collected at any supplier or subset of suppliers where it is available. The amount of the commodities available at the suppliers is limited. In contrast with the usual setting in the literature, the MC2DP requires routing decisions only in the delivery echelon. Indeed, commodities are collected from

* Corresponding author.
E-mail addresses: matteo.petris@inria.fr (M. Petris), archetti@essec.edu (C. Archetti), diego.cattaruzza@centralelille.fr (D. Cattaruzza), maxime.ogier@centralelille.fr (M. Ogier), frederic.semet@centralelille.fr (F. Semet).

https://doi.org/10.1016/j.ejtl.2024.100139
Received 31 August 2023; Received in revised form 5 July 2024; Accepted 11 July 2024
Available online 14 July 2024

the suppliers and brought to the distribution centres via direct round trips. In the delivery echelon, a fleet of vehicles performing routes starting and ending at the same distribution centre is used to deliver the commodities to the customers. All vehicles involved in the distribution system are capacitated and commodities are *compatible*, i.e., they can be mixed inside all vehicles. Finally, as in the *Commodity constrained Split Delivery Vehicle Routing Problem* (C-SDVRP) (Archetti et al., 2016), customers can be visited by multiple vehicles as long as the demand of a single commodity is served by a single vehicle. The aim of the MC2DP is to determine a distribution plan to satisfy customer demands while respecting the capacity of the vehicles and not exceeding the commodity availabilities at the suppliers and such that the total transportation cost is minimized. The MC2DP finds an application in the short and local fresh food supply chains (Berti and Mulligan, 2016) where farmers supply different agricultural products to canteens, restaurants or supermarkets through indirect sales. Commonly, a single decision maker, such as an association of farmers, coordinates both the collection and delivery echelons. In this context, the farmers are less numerous than delivery points since the maximal supply of one farmer can cover the demand of several customers. Hence, the collection from the farmers is usually performed via direct round trips. Then, the distribution centres perform the consolidation operations and the deliveries to the customers which are done by vehicles performing routes. We refer to Gu et al. (2022) for more details about the problem application.

The authors in Gu et al. (2022) proposed a compact Mixed Integer Linear Programming (MILP) formulation and a sequential heuristic for the MC2DP. The authors decompose the MC2DP in two subproblems: one for the collection from suppliers, and the other one for the delivery to customers. The collection subproblem is modelled as a MILP and solved with a commercial solver while the delivery subproblem is solved by an Adaptive Large Neighbourhood Search (ALNS) algorithm.

The contribution of this paper is to present an exended model and to propose the first ever exact approach based on a Branch-Price-and-Cut (BPC) algorithm to solve the MC2DP. Similar exact approaches have recently been proposed to deal with two-echelon vehicle routing problems (see e.g. Marques et al., 2020; Li et al., 2022; Mhamedi et al., 2022; Marques et al., 2022).

However, our BPC algorithm is designed to take into account explicitly the multi-commodity dimension. Specifically, our algorithm relies on a set covering formulation for the MC2DP where the exponentially-many number of variables correspond to the routes in the delivery echelon starting and ending at each distribution centre. We also strengthen the formulation by the insertion of capacity cuts, valid inequalities arising from the set covering polytope (Balas and Ng, 1989) and a new family of valid inequalities based on the number partitioning problem polytope. While capacity cuts are classical inequalities derived for the *Capacitated Vehicle Routing Problem* (CVRP) (see Laporte et al., 1985), the other two families of inequalities tackle the multi-commodity aspect of the problem. Finally, several state-of-art speed-up techniques are also incorporated in our BPC algorithm.

The remainder of the paper is organized as follows. Section 2 provides a literature review. In Section 3, a formal description of the MC2DP is provided. In Section 4, a set covering formulation is presented along with different families of valid inequalities. Our Branch-Price-and-Cut algorithm is described in Section 5. Finally, in Section 6 we analyse the results obtained by the proposed algorithm on the benchmark instances introduced in Gu et al. (2022) to assess its effectiveness.

2. Literature review

In this section, we review the existing literature on the two-echelon distribution problems, with particular attention to the ones dealing with multiple commodities. The first two-echelon routing problem introduced by Jacobsen and Madsen (1980) was motivated by a specific application. Newspapers have to be distributed from a printing office to sales points possibly passing through some transfer points whose locations are to be decided. Crainic et al. (2004, 2009) proposed a formal description of a rich class of two-echelon routing problems along with some economic insights. However, the seminal problem in this class, namely the *two-echelon Capacitated Vehicle Routing Problem* (2E-CVRP), was introduced in the literature and studied for the first time in Perboli et al. (2011). In the 2E-CVRP, a single commodity has to be transferred from a single origin to several destinations through some intermediate facilities. Two fleets of capacitated vehicles perform routes in the two echelons to transport the commodity from the origin to the intermediate facilities and from the intermediate facilities to the destinations. The objective of the 2E-CVRP is to minimize the total transportation cost of the distribution system. The authors proposed two math-heuristics to solve the problem, a diving and a sub-MIP heuristic.

The 2E-CVRP and related problems have received increasing attention in recent years and many variants have been addressed, e.g., 2E-CVRP with (i) time windows (Mhamedi et al., 2022); (ii) mobile satellites (Li et al., 2020); (iii) synchronization (Grangier et al., 2016) and bi-synchronization (Li et al., 2021b); (iv) simultaneous pickup and delivery (Li et al., 2022); (v) electric vehicles (Breunig et al., 2019) and battery swapping stations (Jie et al., 2019); (vi) real-time transshipment capacity varying (Li et al., 2018); (vii) covering options (Enthoven et al., 2020); (viii) delivery options (Zhou et al., 2018); (ix) stochastic demands (Sluijk et al., 2022). The interested reader may refer to Cuda et al. (2015), Li et al. (2021a) and Sluijk et al. (2023) for recent surveys on the subject.

According to the existing literature, the vast majority of the two-echelon routing problems deal with the single commodity case. Apart from the MC2DP, which is addressed in this paper, only a few works integrate multiple commodities in a two-echelon routing problem (e.g. Dellaert et al., 2021; Jia et al., 2023; Gu et al., 2022). In Dellaert et al. (2021), the authors extended the 2E-CVRP by introducing multiple origins and multiple commodities. In addition, hard time windows are imposed for the delivery at the destinations. In their problem, customers have a commodity demand from a specific origin, i.e., there is a *one-to-one* setting. Several mathematical formulations are proposed and a BPC algorithm is devised to solve the problem. In Jia et al. (2023), the problem setting is similar to the one of Dellaert et al. (2021). However, the multi-commodity aspect is handled with more restrictions: only two origins are considered and each destination requires one commodity per origin (one-to-one setting). The authors developed an ALNS algorithm to solve large-scale instances of the problem. The MC2DP introduced in Gu et al. (2022) differs from Dellaert et al. (2021) and Jia et al. (2023) for three reasons: (i) there is a many-to-many setting for the commodities, i.e. any commodity requested by a customer can be served from any supplier; (ii) suppliers provide commodities in limited amounts; (iii) routing decisions are not required in the collection echelon.

3. Problem description

In the *Multi-Commodity two-Echelon Distribution Problem* (MC2DP), a set of commodities $\mathcal{K}$ is distributed in a system involving a set of *suppliers* (origins) S, a set of *distribution centres* (intermediate facilities) D and a set of *customers* (destinations) C. The system is split in two echelons: the collection echelon where the commodities are collected at the suppliers and brought to the distribution centres, and the delivery echelon where the commodities at the distribution centres are delivered to the customers. More precisely, in the collection echelon, each supplier $i \in S$ provides a maximal amount $P_{ik} \geq 0$ for each commodity $k \in \mathcal{K}$. Note that a supplier $i \in S$ might not supply a commodity $k \in \mathcal{K}$, and in that case, P_{ik} takes value 0. An unlimited fleet of homogeneous vehicles of capacity Q^S performs direct round trips from the distribution centres to collect the commodities from the

suppliers. The vehicles can transport any subset of commodities. Due to the limited capacity of the vehicles, direct round trips between a distribution centre $o \in D$ and a supplier $i \in S$ may be performed by several vehicles. The problem associated with the collection operations can be modelled as a *Multi-commodity Capacitated fixed-charge Network Design Problem* (MCNDP, Magnanti and Wong, 1984) with a specific cost structure: there is a step-wise cost function defined by a unitary cost associated with each vehicle used between a distribution centre and a supplier.

Differently, the problem of distributing the commodities from the distribution centres to the customers is a multi-depot version of the *Commodity constrained Split Delivery Vehicle Routing Problem* (C-SDVRP). Each customer $j \in C$ has a demand $D_{jk} \geq 0$ for all commodities $k \in \mathcal{K}$. The request of customer j is identified by set $\mathcal{K}_j = \{k \in \mathcal{K} : D_{jk} > 0\}$. Each distribution centre owns an unlimited fleet of homogeneous and capacitated vehicles of capacity Q^D which performs routes to deliver the commodities to the customers. Each vehicle has to end its route at its starting distribution centre. As in the collection echelon, a vehicle can be loaded with any commodities. Without loss of generality, we suppose $Q^D \geq \max\{\sum_{k\in\mathcal{K}_j} D_{jk} : j \in C\}$. Furthermore, customer requests can be split, i.e., different vehicles can serve the same customer. However, the demand of a single commodity cannot be split: it has to be delivered by a single vehicle. Note that direct trips from suppliers to customers and inter-connections between distribution centres are not allowed.

Finally, the collection and delivery operations taking place in the two echelons are coordinated at the distribution centres by means of the so-called *load synchronization* strategy (Drexl, 2012): the total amount of each commodity collected at the suppliers by each distribution centre must be sufficient to serve the customer demands of that commodity delivered by a vehicle of that distribution centre.

We formulate the MC2DP on a directed weighed graph $\mathcal{G} = (\mathcal{V}, \mathcal{A})$. Set $\mathcal{V} = S \cup D \cup C$ contains a vertex for each supplier, distribution centre and customer. Arc set $\mathcal{A} = \mathcal{A}_S \cup \mathcal{A}_D$ is defined as the union of two sets of arcs which model the possible vehicle travels in the two echelons. Specifically, set $\mathcal{A}_S = (S \times D) \cup (D \times S)$ includes the arcs modelling the direct trips from suppliers to distribution centres in the collection echelon, whereas $\mathcal{A}_D = (D\times C)\cup(C\times D)\cup(C\times C)$ contains all arcs between customers and between distribution centres and customers. Each arc $(i, j) \in \mathcal{A}$ is assigned with a non-negative cost C_{ij} which represent the transportation cost of a vehicle traversing (i, j). The arc costs are symmetric and satisfy the triangular inequality. In graph $\mathcal{G}$, a route in the delivery echelon is a non-empty circuit starting and ending at a distribution centre $o \in D$. A route is *feasible* if the total amount of commodities delivered to the customers visited along the route does not exceed vehicle capacity Q^D. The cost of any feasible route r is $C_r = \sum_{(i,j)\in\mathcal{A}(r)} C_{ij}$, where $\mathcal{A}(r)$ is the set of arcs traversed by the route. Finally, the total transportation cost of the distribution system arising from the MC2DP is the sum of the cost of the direct round trips in the collection echelon and the routing costs in the delivery echelon.

The aim of the MC2DP is to determine a distribution plan, i.e., the direct round trips in the collection echelon and the routes in the delivery echelon, which satisfies the customer requests, does not exceed the commodity availabilities at the suppliers, satisfies the vehicle capacities in both echelons and respects the load synchronization constraints while minimizing the total transportation cost.

4. Problem formulation

We model the MC2DP by means of a set covering formulation, where the exponentially-many variables are associated with the routes in the delivery echelon.

For each distribution centre $o \in D$, we define $\mathcal{R}_o$ as the set of all feasible routes starting and ending at o. The set of all feasible routes is denoted by $\mathcal{R} = \bigcup_{o\in D} \mathcal{R}_o$. We define a binary coefficient a^r_{jk} with value one if commodity $k \in \mathcal{K}$ is delivered to customer $j \in \mathcal{N}$ by route $r \in \mathcal{R}$ and zero otherwise.

For each supplier $i \in S$ and each distribution centre $o \in D$, we introduce an integer variable x_{io} to represent the number of vehicles traversing arc $(i, o) \in \mathcal{A}_S$. For each $i \in S$, $o \in D$ and $k \in \mathcal{K}$, we define a non-negative continuous variable q^k_{io} that represents the amount of commodity k collected at supplier i by distribution centre o. Finally, for each route $r \in \mathcal{R}$, we introduce a binary variable λ_r taking value one if r is selected in the solution and zero otherwise.

The Set Covering formulation [SC] for the MC2DP reads as follows:

$$[\text{SC}] \quad \min \sum_{(i,o)\in\mathcal{A}_S} 2C_{io}x_{io} + \sum_{r\in\mathcal{R}} C_r\lambda_r \tag{1}$$

$$\text{s.t.} \sum_{o\in D} q^k_{io} \leq P_{ik} \qquad \forall i \in S, \forall k \in \mathcal{K} \tag{2}$$

$$\sum_{k\in\mathcal{K}} q^k_{io} \leq Q^S x_{io} \qquad \forall i \in S, \forall o \in D \tag{3}$$

$$\sum_{r\in\mathcal{R}} a^r_{jk}\lambda_r \geq 1 \qquad \forall j \in C, \forall k \in \mathcal{K}_j \tag{4}$$

$$\sum_{i\in S} q^k_{io} \geq \sum_{r\in\mathcal{R}_o}\sum_{j\in C} a^r_{jk}D_{jk}\lambda_r \qquad \forall o \in D, \forall k \in \mathcal{K} \tag{5}$$

$$x_{io} \in \mathbb{Z}_{\geq 0} \qquad \forall i \in S, \forall o \in D \tag{6}$$

$$q^k_{io} \in \mathbb{R}_{\geq 0} \qquad \forall i \in S, \forall o \in D, \forall k \in \mathcal{K} \tag{7}$$

$$\lambda_r \in \{0, 1\} \qquad \forall r \in \mathcal{R} \tag{8}$$

Objective function (1) minimizes the total transportation cost. Constraints (2) ensure that the commodity availabilities at each supplier are respected. Constraints (3) guarantee that a sufficient number of vehicles perform the collection operations and that the capacity of these vehicles is not exceeded. Covering Constraints (4) impose that each commodity required by a customer is served by at least one route. In addition, the load synchronization constraint linking the collection and delivery echelons is expressed in constraints (5): the quantity of each commodity collected by each distribution centre has to be large enough to satisfy the demand for that commodity delivered by a route of that distribution centre. Finally, Constraints (6), (7) and (8) define variable domains.

4.1. Valid inequalities

In this section, we introduce four families of valid inequalities considered to strengthen formulation [SC]. Two of these inequalities are known in the context of vehicle routing problems, while the other two are tailored to deal with the multi-commodity aspect of the MC2DP. Note that such inequalities are valid for the C-SDVRP, hence for the MC2DP.

In what follows, given a subset of customers $C' \subseteq C$, we define $D(C') = \sum_{j\in C'}\sum_{k\in\mathcal{K}_j} D_{jk}$ to be the total demand requested by the customers in C'. In addition, we introduce a binary coefficient b^r_{ij} with value one if route $r \in \mathcal{R}$ traverses arc $(i, j) \in \mathcal{A}_D$ and zero otherwise. Finally, we define $e^r_{j\mathcal{M}} = \prod_{k\in\mathcal{M}} a^r_{jk}$ to be a binary coefficient equal to one if route r delivers all the commodities of subset $\mathcal{M} \subseteq \mathcal{K}_j$ to customer $j \in C$ and zero otherwise.

Bounds on the number of vehicles

The following inequalities set bounds on the number of vehicles in the collection and delivery echelons:

$$\sum_{(i,o)\in\mathcal{A}_S} x_{io} \geq \left\lceil \frac{D(C)}{Q^S} \right\rceil \tag{9}$$

and

$$\sum_{r\in\mathcal{R}} \lambda_r \geq \lceil \underline{v} \rceil \tag{10a}$$

$$\sum_{r\in\mathcal{R}} \lambda_r \leq \min\{|C|, 2\bar{v}\}. \tag{10b}$$

In inequalities (10a) and (10b), values $\underline{v}$ and $\bar{v}$ are obtained by solving an instance of the *Bin Packing Problem* (BPP), where bins have size equal to the vehicle capacity Q^D, and each customer demand has a corresponding item to be packed with size D_{jk}. Precisely, we solve an integer program for the BPP on such an instance with a commercial solver within a short time limit: $\underline{v}$ and $\bar{v}$ are the obtained lower and upper bounds. If the instance is solved to optimality within the time limit, $\underline{v} = \bar{v}$ holds. The right hand-side of (10b) is the minimum between twice value $\bar{v}$ (see Federgruen and Simchi-Levi, 1995) and the number of customers.

Capacity cuts

Laporte et al. (1985) introduced the capacity cuts to deal with the *Capacitated Vehicle Routing Problem*:

$$\sum_{r\in\mathcal{R}}\left(\sum_{(i,j)\in\delta^-(C')} b^r_{ij}\right)\lambda_r \geq \left\lceil \frac{D(C')}{Q^D}\right\rceil \qquad \forall C'\subseteq C, \tag{11}$$

where $\delta^-(C') = \{(i,j)\in\mathcal{A}_D : i\notin C', j\in C'\}$ is the set of arcs of graph $\mathcal{G}$ reaching a vertex in C'. Given a subset of customers C', inequality (11) states that at least $\lceil D(C')/Q^D\rceil$ vehicles of the delivery echelon are required to cover the requests of the customers in C'.

Set covering polytope

We present a family of valid inequalities inspired by the facet-defining inequalities proposed in Balas and Ng (1989) for the set covering polytope. Although these inequalities were proposed several years ago, to the best of our knowledge, they have not yet been used in BPC algorithms for vehicle routing problems. However, they are similar to the *strong minimum number of vehicles inequalities* introduced by Archetti et al. (2011) in the context of a BPC algorithm for the split delivery vehicle routing problem with time windows.

Let us first briefly present a formulation for the set covering problem. Let $\mathcal{I}$ bet a set of elements to be covered, and $\mathcal{J}$ be a set of subsets of $\mathcal{I}$. We denote by c_j the cost associated to subset $j\in\mathcal{J}$, and d_{ij} a binary parameter that takes value one if element $i\in\mathcal{I}$ is in subset $j\in\mathcal{J}$, and zero otherwise. Let x_j be a binary decision variable taking value one if subset $j\in\mathcal{J}$ is selected, zero otherwise. An integer programming formulation for the set covering problem is

$$\begin{aligned}
&\min \sum_{j\in\mathcal{J}} c_j x_j \\
&\text{s.t.} \sum_{j\in\mathcal{J}} d_{ij}x_j \geq 1 && \forall i\in\mathcal{I}\\
&\quad x_j\in\{0,1\} && \forall j\in\mathcal{J}
\end{aligned}$$

Given a subset $\mathcal{I}'\subseteq\mathcal{I}$, the inequalities introduced in Balas and Ng (1989) reads as follows:

$$2\sum_{j\in\mathcal{J}^{\mathcal{I}'}} x_j + \sum_{j\in\bar{\mathcal{J}}^{\mathcal{I}'}} x_j \geq 2,$$

where $\mathcal{J}^{\mathcal{I}'} = \{j\in\mathcal{J} : d_{ij}=1, \forall i\in\mathcal{I}'\}$ is the set of the elements of $\mathcal{J}$ which cover $\mathcal{I}'$ and $\bar{\mathcal{J}}^{\mathcal{I}'} = \{j\in\mathcal{J} : \sum_{i\in\mathcal{I}'} d_{ij}\geq 1 \wedge \prod_{i\in\mathcal{I}'} d_{ij} = 0\}$ is the set of the elements of $\mathcal{J}$ which contain some, but not all, the elements in $\mathcal{I}'$. The inequalities express how subset $\mathcal{I}'$ may be covered: either it suffices to select a unique element in $\mathcal{J}$ that covers $\mathcal{I}'$, i.e., an element in $\mathcal{J}^{\mathcal{I}'}$, or at least two elements in $\mathcal{J}$ that partially cover $\mathcal{I}'$ have to be selected, i.e., at least two elements in $\bar{\mathcal{J}}^{\mathcal{I}'}$. Under specific conditions, these constraints are facet defining for the set covering polytope.

In what follows, we adapt these inequalities to the MC2DP to express how the subsets of commodities required by a given customer may be covered. For the ease of readability, we introduce the following notation. Let $j\in C$ be a customer and $\mathcal{M}_j\subseteq\mathcal{K}_j$ be a subset of the commodities requested by j. We denote by $\mathcal{R}_j^{\mathcal{M}_j}\subseteq\mathcal{R}$ the subset of routes delivering all commodities in $\mathcal{M}_j$ to j, i.e., $\mathcal{R}_j^{\mathcal{M}_j} = \{r\in\mathcal{R} : e^r_{j,\mathcal{M}_j} = 1\}$.

In addition, we write $\bar{\mathcal{R}}_j^{\mathcal{M}_j}\subseteq\mathcal{R}$ for the subset of routes which deliver some of the commodities in $\mathcal{M}_j$ to j, but not all of them, i.e., $\bar{\mathcal{R}}_j^{\mathcal{M}_j} = \{r\in\mathcal{R} : \sum_{k\in\mathcal{M}_j} a^r_{jk}\geq 1 \wedge e^r_{j,\mathcal{M}_j} = 0\}$.

The set covering polytope inequalities for the MC2DP are defined as follows:

$$2\sum_{r\in\mathcal{R}_j^{\mathcal{M}_j}}\lambda_r + \sum_{r\in\bar{\mathcal{R}}_j^{\mathcal{M}_j}}\lambda_r \geq 2 \qquad \forall j\in C, \forall\mathcal{M}_j\subseteq\mathcal{K}_j. \tag{12}$$

Inequalities (12) state that subset of commodities $\mathcal{M}_j\subseteq\mathcal{K}_j$ of customer $j\in C$ can be covered either by a single route in $\mathcal{R}_j^{\mathcal{M}_j}$ or by at least two routes in $\bar{\mathcal{R}}_j^{\mathcal{M}_j}$. Note that these inequalities are meaningful only if $|\mathcal{M}_j|\geq 3$. Indeed, if $|\mathcal{M}_j| = 2$, they can be retrieved as an aggregation of Covering Constraints (4).

Number partitioning polytope

We propose a novel family of valid inequalities which exploits the multi-commodity aspect of the MC2DP. More precisely, given a customer $j\in C$, these inequalities specify the possible combinations of routes to deliver the set of commodities $\mathcal{K}_j$ required by customer j.

For each customer $j\in C$, we denote by $\mathcal{R}_j^l$ the subset of routes which deliver exactly $l = 1,\dots,|\mathcal{K}_j|$ commodities to j, i.e., $\mathcal{R}_j^l = \{r\in\mathcal{R} : \sum_{k\in\mathcal{K}_j} a^r_{jk} = l\}$.

Equalities

$$\sum_{l=1}^{|\mathcal{K}_j|} l\sum_{r\in\mathcal{R}_j^l}\lambda_r = |\mathcal{K}_j| \qquad \forall j\in C \tag{13}$$

ensure that the selected routes that serve customer j will exactly bring $|\mathcal{K}_j|$ commodities to customer j. As an example, let $\bar{j}\in C$ be a customer having a demand for three commodities, i.e., $|\mathcal{K}_{\bar{j}}| = 3$. Equality (13) for customer $\bar{j}$ states that the commodities of $\mathcal{K}_{\bar{j}}$ can be covered by (i) a single route of $\mathcal{R}^3_{\bar{j}}$ or (ii) one route of $\mathcal{R}^2_{\bar{j}}$ and a route of $\mathcal{R}^1_{\bar{j}}$ or (iii) three routes of $\mathcal{R}^1_{\bar{j}}$.

Proposition 1. *Equalities (13) are valid for the MC2DP. More precisely, inequalities* $\sum_{l=1}^{|\mathcal{K}_j|} l\sum_{r\in\mathcal{R}_j^l}\lambda_r \geq |\mathcal{K}_j|$, $\forall j\in C$, *are implied by Covering Constraints (4) and inequalities*

$$\sum_{l=1}^{|\mathcal{K}_j|} l\sum_{r\in\mathcal{R}_j^l}\lambda_r \leq |\mathcal{K}_j| \qquad \forall j\in C \tag{14}$$

are valid for the MC2DP.

Proof. It is straightforward that equalities (14) are valid for the MC2DP. Hence, we only need to show that $\sum_{l=1}^{|\mathcal{K}_j|} l\sum_{r\in\mathcal{R}_j^l}\lambda_r \geq |\mathcal{K}_j|$, $\forall j\in C$, are implied by Covering Constraints (4). Let $j\in C$ be a customer. By summing up the Covering Constraints (4) associated with j and swapping the summation order, we obtain

$$\sum_{r\in\mathcal{R}}\sum_{k\in\mathcal{K}_j} a^r_{jk}\lambda_r \geq |\mathcal{K}_j|.$$

Let $\mathcal{M}_j^r$ denote the subset of commodities delivered to customer j by route r. We have $\sum_{k\in\mathcal{K}_j} a^r_{jk} = |\mathcal{M}_j^r|$. The proof follows from partitioning the set of routes as $\mathcal{R} = \bigcup_{l=0}^{|\mathcal{K}_j|}\mathcal{R}_j^l$, where we denoted by $\mathcal{R}_j^0$ the subset of routes which do not visit j. Indeed, it holds

$$|\mathcal{K}_j| \leq \sum_{l=0}^{|\mathcal{K}_j|}\sum_{r\in\mathcal{R}_j^l}|\mathcal{M}_j^r|\lambda_r = \sum_{l=1}^{|\mathcal{K}_j|} l\sum_{r\in\mathcal{R}_j^l}\lambda_r. \quad \square$$

Remark that if we model the MC2DP by means of a set partitioning formulation, i.e., we impose the equality in Constraints (4), Equalities (13) become trivial. Indeed, they can be retrieved as an aggregation of the partitioning constraints.

Given a customer $j \in C$ and $l = 1, \dots, |\mathcal{K}_j|$, we introduce an auxiliary variable $y_j^l \in \mathbb{Z}_{\geq 0}$ defined as $y_j^l := \sum_{r \in \mathcal{R}_j^l} \lambda_r$. Now, let

$$\mathcal{F}_j := \{y_j \in \mathbb{Z}_{\geq 0}^{|\mathcal{K}_j|} : \sum_{l=1}^{|\mathcal{K}_j|} l y_j^l \leq |\mathcal{K}_j|\}$$

be the set of the integer points which satisfy inequality (14), rewritten in terms of y_j^l variables.

Proposition 2. *The inequalities defining the convex hull of $\mathcal{F}_j$, $j \in C$, are valid for the MC2DP.*

Determining the external description of a convex set is not an easy task, in particular in large dimensions. However, given that customers require at most three commodities in the benchmark instances of Gu et al. (2022) for the MC2DP, we explicitly derive the external description of the convex hull of sets $\mathcal{F}_j \subseteq \mathbb{Z}^3$, $j \in C$. If the number of commodities is greater than three, software for polyhedral transformations such as PORTA (Christof and Löbel, 2009) or PANDA (Lörwald and Reinelt, 2015) can be used to determine the external description of the convex hull of sets $\mathcal{F}_j$, $j \in C$.

Note that inequalities (14) are meaningful only for customers $j \in C$ who require at least three commodities, i.e., $|\mathcal{K}_j| \geq 3$. The external description of the convex hull of sets $\mathcal{F}_j$, $j \in C$ such that $|\mathcal{K}_j| = 3$ reads as follows:

$$\begin{cases} y_j^1 + 2y_j^2 + 3y_j^3 \leq 3 & \text{(a)} \\ y_j^1 - y_j^2 \geq 0 & \text{(b)} \\ y_j^2 \geq 0 & \text{(c)} \\ y_j^3 \geq 0. & \text{(d)} \end{cases} \tag{15}$$

Inequalities (15c) and (15d) are trivial, indeed, they are implied by the definition of variables y_j^l. Therefore, inequalities (15a) and (15b) are the only meaningful ones in the case of a customer $j \in C$ requiring three commodities ($|\mathcal{K}_j| = 3$); in terms of λ variables, they are expressed respectively as

$$\sum_{l=1}^{|\mathcal{K}_j|} l \sum_{r \in \mathcal{R}_j^l} \lambda_r \leq |\mathcal{K}_j| \qquad \forall j \in C : |\mathcal{K}_j| = 3 \tag{16}$$

$$\sum_{r \in \mathcal{R}_j^1} \lambda_r - \sum_{r \in \mathcal{R}_j^2} \lambda_r \geq 0 \qquad \forall j \in C : |\mathcal{K}_j| = 3. \tag{17}$$

In conclusion, the number partitioning polytope valid inequalities we consider are (16) and (17).

5. Branch-Price-and-Cut algorithm

We solve formulation [SC] by means of a Branch-Price-and-Cut (BPC) algorithm (Barnhart et al., 1998), i.e., a variant of the branch-and-bound algorithm which deals with integer programming model with exponentially-many variables. Specifically, at each node of the branch-and-bound tree, the Master Problem (MP), that is the linear relaxation of formulation [SC], is solved by a column generation procedure (Desrosiers and Lübbecke, 2005). If the solution of the MP is fractional, violated valid inequalities of Section 4.1 may be inserted and the column generation procedure is repeated while some valid inequalities are violated. Finally, branching rules are applied to ensure the integrality of the solution. We impose a time limit as a termination criterion for our BPC algorithm.

In this section, we describe the main components of our BPC algorithm. Specifically, in Section 5.1 we present the column generation scheme applied in our BPC algorithm. In Section 5.2, we detail the management of the valid inequalities, and their impact on the pricing problem. Branching strategies and accelerating techniques are presented in Sections 5.3 and 5.4, respectively.

5.1. Column generation

At each node of the branch-and-bound tree, a column generation procedure solves the MP defined on the exponentially-many variables λ_r, $r \in \mathcal{R}$, which correspond to the routes in the delivery echelon. The starting point is the Restricted Master Problem (RMP). The column generation procedure iteratively solves a Restricted Master Problem (RMP), i.e., the MP restricted to a subset of variables λ_r. At each iteration of the procedure, after the RMP is solved, a subproblem, named *pricing problem* is solved. The aim of the pricing problem is to identify a variable (column) with the smallest reduced cost. If such a column has a negative reduced cost, it is added to the RMP in order to decrease (in a minimization problem) the current value of the solution, and the column generation procedure iterates. The procedure ends when the solution of the pricing problem is a non negative reduced cost column, proving the optimality of the MP.

More precisely, the pricing problem is

$$\text{[PP]} \quad \min\{\bar{C}_r : r \in \mathcal{R}\}$$

where $\bar{C}_r$ denotes the reduced cost of λ_r variable. Note that set of routes $\mathcal{R}$ can be partitioned per distribution centre, i.e., $\mathcal{R} = \bigcup_{o \in D} \mathcal{R}_o$ where $\mathcal{R}_o$ is the set of routes starting and ending at o. Hence, solving [PP] can be done by solving sequentially $|D|$ independent problems with the same structure:

$$\text{[PP}(o)\text{]} \quad \min\{\bar{C}_r : r \in \mathcal{R}_o\}, \quad o \in D.$$

Specifically, the aim of problem [PP(o)] is to determine the most negative reduced cost λ_r, $r \in \mathcal{R}_o$, or to detect that none of them exists. The column generation procedure terminates once all problems [PP(o)], $o \in D$ do not yield any negative reduced cost variable.

In the following, we detail how a problem [PP(o)] for $o \in D$ is formulated and solved. By denoting $\rho_{jk} \geq 0$, $\forall j \in C, k \in \mathcal{K}_j$ and $\sigma_{ok} \geq 0$, $\forall o \in D, k \in \mathcal{K}$ as the optimal dual prices associated with Constraints (4) and (5), respectively, the reduced cost of a λ_r, $r \in \mathcal{R}_o$ variable is defined as follows:

$$\bar{C}_r = C_r - \sum_{j \in C} \sum_{k \in \mathcal{K}_j} a_{jk}^r (\rho_{jk} - D_{jk}\sigma_{ok}). \tag{18}$$

As mentioned in Section 3, the delivery echelon is a multi-depot version of the C-SDVRP. Hence, the problem [PP(o)] is the pricing problem arising in Branch-Price-and-Cut approaches for the C-SDVRP (see Archetti et al., 2015; Gschwind et al., 2019) and is formulated as an *Elementary Shortest Path Problem with Resource Constraints* (ESPPRC) on a multi-graph $\mathcal{G}(o) = (\mathcal{V}(o), \mathcal{A}(o))$. Such graph is analogous to the one presented in Gschwind et al. (2019) to formulate the ESPPRC in the context of the C-SDVRP. Vertex set $\mathcal{V}(o)$ contains two copies o' and o'' of distribution centre o and two copies j' and j'' of each customer $j \in C$. Each arc of set $\mathcal{A}(o)$ is associated with two resources: demand $\bar{D}$ and cost $\bar{C}$. Arc set $\mathcal{A}(o)$ contains:

1. an arc (i'', j') for each arc $(i, j) \in \mathcal{A}$ to model the movement of a vehicle from vertex i to vertex j; the demand and cost are set to $\bar{D}_{i''j'} := 0$ and $\bar{C}_{i''j'} := C_{ij}$, respectively.
2. an arc $(j', j'')^{\mathcal{M}_j}$ for each customer $j \in C$ and each subset $\mathcal{M}_j \subseteq \mathcal{K}_j$ to model the delivery of the commodities of $\mathcal{M}_j$ to j; the demand and cost are set to $\bar{D}_{j'j''}^{\mathcal{M}_j} := \sum_{k \in \mathcal{M}_j} D_{jk}$ and $\bar{C}_{j'j''}^{\mathcal{M}_j} := -\sum_{k \in \mathcal{M}_j} (\rho_{jk} - D_{jk}\sigma_{ok})$, respectively.

Solving problem [PP(o)] results in searching for negative reduced cost elementary paths in $\mathcal{G}(o)$ from o'' to o' such that the resource consumption (demand) does not exceed the vehicle capacity Q^D.

To do so, we adopt a two phase procedure:

Phase 1 computes the Pareto-optimal (demand, cost) pairs $(\bar{D}_{j'j''}^{\mathcal{M}_j}, \bar{C}_{j'j''}^{\mathcal{M}_j})$ for each customer $j \in C$.

Phase 2 solves the ESPPRC on multi-graph $\mathcal{G}(o)$ which includes all arcs of type (i'', j'), and only the Pareto-optimal arcs of type $(j', j'')^{\mathcal{M}_j}$ that have been computed in phase 1. Precisely, the ESPPRC is solved by means of a label setting dynamic programming technique (Feillet et al., 2004) which works with an implicit version of the bidirectional labelling search (see Righini and Salani, 2006; Bode and Irnich, 2012). The elementarity constraints are the bottleneck of such procedure, hence, we partially relax it by solving the *ng-path* relaxation (Baldacci et al., 2011) of the ESPPRC. For each customer $j \in C$, we consider a fixed size *ng-neighbourhood* which includes the 10 closest customers to j and j itself. Remark that such relaxation allows a route to serve the same commodity to the same customer multiple times. Hence, the coefficients of the constraints and valid inequalities need to be updated accordingly: e.g., in the Covering Constraints (4), a^r_{jk} becomes an integer coefficient expressing the number of times customer $j \in C$ is delivered with commodity $k \in \mathcal{K}_j$ by route $r \in \mathcal{R}$.

The reader may refer to Archetti et al. (2015) and Gschwind et al. (2019) for further details. The resolution of the ESPPRCs is the bottleneck of our algorithm, hence, we heuristically solve the ESPPRC with the objective of quickly finding a negative reduced cost column. Precisely, we apply the same heuristic algorithms to solve the ESPPRC as those used in the column generation approach for the C-SDVRP proposed in Petris et al. (2023). One of these heuristics is a two-phase algorithm which exploits the multi-commodity aspect of the problem, while the others are based on reducing the pricing multi-graph by restricting the set of neighbours of the customers and by limiting the total number of splits in a route. When all the heuristics fail to identify a negative reduced cost column, we solve the ESPPRC exactly.

5.2. Management of the valid inequalities

In this section, we first describe how the valid inequalities presented in Section 4.1 are considered in the pricing problem. Then, we present the cutting strategy adopted in our BPC algorithm.

Impact of the valid inequalities on the pricing problem

First, note that inequality (9) imposes a lower bound on the number of vehicles used in the collection echelon. Therefore, it has no impact on the pricing problem. The other inequalities presented in Section 4.1 are all robust, i.e. they do not change the structure of the pricing problem, and their associated dual prices have to be integrated into the objective function of pricing problems [PP(o)], $o \in D$, i.e. on the cost of arcs in multi-graph $\mathcal{G}(o)$.

The arc costs in multi-graph $\mathcal{G}(o)$ are modified in the following way:

Inequalities (10a) and (10b). Let $\tau^+ \geq 0$ and $\tau^- \leq 0$ be the optimal dual prices associated with valid inequalities (10a) and (10b) respectively. The value $\tau^+/2 + \tau^-/2$ is subtracted from the cost of arcs of type (i'', j'), if vertices i or j represent distribution centre o.

Inequalities (11). Let $\xi_{C'} \geq 0$ be the optimal dual prices associated with the capacity cut (11) defined over the subset of customers $C' \subseteq C$. Let $\delta^-(C')$ be the subset of arcs in graph $\mathcal{G}$ entering in vertices of C'. The value $\xi_{C'}$ is subtracted from the cost of arcs (i'', j'), for all $(i, j) \in \delta^-(C')$.

Inequalities (12). Let $\gamma_{j\mathcal{M}_j} \geq 0$ be the optimal dual prices associated with the inequality (12) identified by customer $j \in C$ and commodity subset $\mathcal{M}_j \subseteq \mathcal{K}_j$. The value $2\gamma_{j\mathcal{M}_j}$ is subtracted from the cost of arcs $(j', j'')^{\mathcal{M}'_j}$, for all $\mathcal{M}'_j \subseteq \mathcal{K}_j$ that contain at least all the commodities of $\mathcal{M}_j$, i.e., $\mathcal{M}_j \subseteq \mathcal{M}'_j$. The value $\gamma_{j\mathcal{M}_j}$ is subtracted from the cost of arcs $(j', j'')^{\mathcal{M}'_j}$ for all $\mathcal{M}'_j$ that contain some, but not all, commodities of $\mathcal{M}_j$, i.e. $\mathcal{M}'_j \cap \mathcal{M}_j \neq \emptyset$ and $\mathcal{M}'_j \cap \mathcal{M}_j \neq \mathcal{M}_j$.

Inequalities (16) and (17). Let $j \in C$ be a customer requiring exactly three commodities ($|\mathcal{K}_j| = 3$) and let $\psi \geq 0$ and $\chi \leq 0$ be the optimal dual prices associated with inequalities (16) and (17) defined on j. For all $\mathcal{M}_j \subseteq \mathcal{K}_j$, the cost of arc $(j', j'')^{\mathcal{M}_j}$ is modified as follows: value $|\mathcal{M}_j|\psi$ is subtracted, value χ is added if $|\mathcal{M}_j| = 2$, and value χ is subtracted if $|\mathcal{M}_j| = 1$.

Management of the valid inequalities in the RMP

Valid inequalities on vehicle bounds, namely (9), (10a) and (10b), are included in the formulation from the beginning of the solution procedure. Differently, a cut generation procedure manages the insertion of violated inequalities (11), (12), (16) and (17) in the RMP. Such a procedure is called at each node of the branch-and-bound tree of level at most equal to 5, if the associated solution of the RMP is fractional. Specifically, it separates the inequalities hierarchically according to the sequence : (11), (12), (16), and (17). When the separation of a given inequality fails, we separate the next one in the above order. The separation of inequalities (11) is done using the heuristic algorithms presented in Ralphs et al. (2003), namely the *extended shrinking heuristic* and the *greedy shrinking heuristic*. Then, although, inequalities (12) are exponentially-many, the size of the problem instances allows the separation by enumeration. The same separation strategy is applied for inequalities (16) and (17), whose number is linear in the number of customers $|C|$. Finally, we limit the number of inequalities (11) to 100 in each cut generation round. For the other inequalities, we include all the violated inequalities.

5.3. Branching strategies

Let $(\bar{x}, \bar{q}, \bar{\lambda})$ be a fractional optimal solution of the MP at a certain node of the branch-and-bound tree. We consider seven branching rules that are hierarchically applied. In addition to these rules, the correctness of the algorithm requires the separation of a family of valid inequalities, namely the *strong-degree* inequalities. Rules 1 and 3 are specific for the MC2DP, while the other ones and the family of valid inequalities are used to solve the C-SDVRP by Branch-and-Price. The interested reader can refer to Gschwind et al. (2019) for more details about the branching strategy for the C-SDVRP.

Rule 1 is on the number of vehicles traversing an arc in the collection echelon, i.e., on value $\bar{x}_{io}$, $i \in S$, $o \in D$. Since λ_r variables are not concerned by this rule, there is no impact on the pricing problem.

Rule 2 is on the number of vehicles used at each distribution centre $o \in D$ in the delivery echelon, i.e., on value $\sum_{r \in \mathcal{R}_o} \bar{\lambda}_r$.

Rule 3 forces the assignment of a delivery to a distribution centre. Specifically, given a distribution centre $o \in D$, a customer $j \in C$ and a commodity $k \in \mathcal{K}_j$, we branch on value $p^o_{jk} := \sum_{r \in \mathcal{R}_o} a^r_{jk} \bar{\lambda}_r$. The branching decisions related to this rule can be expressed as follows: commodity k required by customer j is either delivered from distribution centre o, i.e. $\sum_{o' \in D \setminus \{o\}} \sum_{r \in \mathcal{R}_{o'}} a^r_{jk} \lambda_r = 0$; or not delivered from o, i.e. $\sum_{r \in \mathcal{R}_o} a^r_{jk} \lambda_r = 0$. Note that both decisions entail modifications in the pricing problem. As an example, if the first decision is imposed, then we prevent the pricing problem from generating routes starting and ending at distribution centres $o' \in D \setminus \{o\}$ and delivering commodity k to customer j. Arcs of type $(j', j'')^{\mathcal{M}_j}$, with $\mathcal{M}_j \subseteq \mathcal{K}_j$ and $k \in \mathcal{M}_j$, are removed from all multi-graphs $\mathcal{G}(o')$, $o' \in D \setminus \{o\}$.

Rule 4 is on the number of visits to each customer $j \in C$ from a distribution centre $o \in D$.

Rule 5 considers the flow on the edges in the delivery echelon coming from a specific distribution centre.

Rules 6 and 7 implement the Ryan and Foster branching rules (Ryan and Foster, 1981) which force the two customer requests to be served by different routes or by the same route. Such rules imply the addition of non-robust constraints in the RMP . The management of the associated dual variables in the labelling algorithm used to solve the pricing problem can be found in Gschwind et al. (2019).

These seven rules are sufficient to ensure the correctness of the algorithm when only elementary routes are present in formulation [SC]. Indeed, Rule 1 guarantees the integrality of the variables of the collection echelon. Then, regarding the delivery echelon, Rule 3 assigns the deliveries to a specific distribution centre. Once these assignments are done, Rules 2 and 4–7 are enough to guarantee the correctness of the algorithm. Indeed, the delivery echelon is a multi-depot version of the C-SDVRP and such rules ensure the integrality of a solution for the C-SDVRP (see Gschwind et al., 2019). However, as mentioned in Section 5.1, we relax the elementarity requirement of the routes in the second echelon via the ng-path relaxation when solving the pricing problem. Consequently, formulation [SC] may contain routes that serve the same commodity to the same customer more than once. In such a case, applying only Rules 1–7 might lead to a fractional solution as shown by Gschwind et al. (2019) for the C-SDVRP. Hence, after applying Rule 7, *strong-degree* inequalities (Contardo et al., 2014) have to be separated to ensure providing an integer solution. The strong-degree inequalities read as:

$$\sum_{r\in\mathcal{R}} \xi^r_{jk}\lambda_r \geq 1 \qquad \forall j \in C, \forall j \in \mathcal{K}_j,$$

where ξ^r_{jk} is a binary coefficient with value one if route $r \in \mathcal{R}$ delivers customer $j \in C$ with commodity $k \in \mathcal{K}_j$. In our branching strategy, when none of the seven rules are applicable, we separate these inequalities. As for Rules 6 and 7, these inequalities are non-robust. The management of the associated dual variables in the labelling algorithm invoked to solve the pricing problem is described in Contardo et al. (2014).

The branch-and-bound tree is explored according to a best-bound first strategy to favour the improvement of the dual bound. The strategies to select the branching decisions are presented in the following. For rule 2, we branch on the fractional value closest to 0.5. For rules 6 and 7, we branch on the first fractional value that is found. For all the other rules, we consider a two-round strong branching procedure (S., 2012) similar to the one presented in Pessoa et al. (2020). In the first round, we evaluate at most 100 branching candidates according to the *product rule* (Achterberg, 2007). More precisely, each candidate gives rise to two branching decisions d_1 and d_2 and is evaluated by applying such decisions to the RMP and by solving it without generating columns. Then, each candidate is assigned with a score $sc(d_1, d_2) = \max\{\epsilon, \Delta LB_1\} \times \max\{\epsilon, \Delta LB_2\}$, where $\epsilon = 10^{-6}$ and ΔLB_i is the increase of the lower bound obtained by applying decision d_i to the RMP. The three candidates with the highest scores are sent to the second round, where the same evaluation criterion is used to select the winning candidate. Differently from the first round, here, LB_1 and LB_2 are the values of the RMP after a single column generation iteration where the pricing problem is solved heuristically.

The strong branching procedure is employed in nodes of the branch-and-bound tree of level at most 5. In the other levels, we evaluate the branching candidates based on the fractional value closest to 0.5 for all the rules.

5.4. Accelerating strategies

The BPC algorithm incorporates the following accelerating strategies:

Initialization of set $\mathcal{R}$. We initialize the set of routes $\mathcal{R}$ to avoid very large dual prices at the first iterations of the column generation procedure which may slow down the pricing solution (Desaulniers, 2010). Specifically, for each distribution centre $o \in \mathcal{D}$, we include round-trips (0-j-0) to each customer $j \in \mathcal{N}$, which deliver the commodities of each subset $\mathcal{M}_j \subseteq \mathcal{K}_j$ requested by j. In addition, we modified the randomized Clarke-Wright algorithm (CW) (Clarke and Wright, 1964) proposed in Battarra et al. (2008) to take into account the multi-commodity aspect of our problem. The algorithm is run 10 times per distribution centre and the obtained routes are inserted into $\mathcal{R}$.

Heuristic column generators. Before solving the pricing problem to optimality, we consider heuristic column generators to speed up the solution of problems [PP(o)], $o \in \mathcal{D}$. As mentioned in Section 5.1, each problem [PP(o)] is the pricing problem arising in a BPC algorithm for the C-SDVRP. Hence, we apply the same heuristic scheme used in Petris et al. (2023) which proved to be effective in accelerating such pricing problems. This scheme considers two reduced graph heuristics and the two-phase heuristic introduced in Petris et al. (2023) which proved to be effective in dealing with the multi-commodity aspect of the C-SDVRP. The two reduced graph heuristics reduce the size of multi-graphs $\mathcal{G}(o)$, $o \in \mathcal{D}$ by limiting both the possibilities of travelling between customers and of deliveries to customers. In the two-phase heuristic, the aim of the first phase is to compute a set of promising customer sequences by solving the ESPPRC on a modified version of multi-graphs $\mathcal{G}(o)$ where only one delivery per customer is allowed. Specifically, when visiting a customer, the least consuming commodity is delivered and all the profitable dual prices are collected. In the second phase, for each of the customer sequences generated by the first phase, we solve the ESPPRC on the associated acyclic graphs to obtain all negative reduced cost routes arising from the sequence. We refer to Petris et al. (2023) for more details.

Restricted master heuristic. We invoke a restricted master heuristic, which consists in solving the formulation [SC] restricted to the subset of variables generated so far, to obtain good upper bounds. Such a technique helps to reduce the integrality gap (see Archetti et al., 2013). Note that variables λ_r are then binary. We call the restricted master heuristic every 1000 explored nodes in the branch-and-bound tree as well as when the time limit of the algorithm is reached. In this latter case, we apply a local search procedure based on an adapted version of the mathematical programming operator proposed for the C-SDVRP in Gu et al. (2019). Specifically, we generalized such an operator to deal with the two-echelon case. When the restricted master heuristic is called during the tree exploration a time limit of 3 seconds is imposed, while the time limit is 30 seconds when the algorithm terminates.

6. Computational experiments

We implemented the BPC algorithm in C++ and compiled it in release mode under a 64-bit version of MS Visual Studio 2019. IBM CPLEX 12.9.0 (64-bit version) is used as a solver. We performed the experiments on a 64-bit Windows machine equipped with a Intel(R) Xeon(R) Silver 4214 processor with 24 cores hyper-threaded to 48 virtual cores, with a base clock frequency of 2.2 GHz, and 96 GB of RAM. For each run of the algorithm, we impose one hour time limit and allow a single thread.

In this section, first, we describe the characteristics of the benchmark instances for the MC2DP introduced in Gu et al. (2022). Then, we discuss the impact of valid inequalities (12), (16) and (17). Finally, we evaluate the effectiveness of the BPC algorithm against solving the compact formulation for the MC2DP presented in Gu et al. (2022) with a commercial solver and we present the results obtained by the BPC algorithm on the benchmark instances.

Table 1
Characteristics of the sets of instances.

Set	#	Characteristics			
		$\|S\|$	$\|C\|$	$\|\mathcal{K}\|$	Description
$\mathscr{S}$	64	8	30	2, 3	Base set
$\mathscr{S}_1^S$	64	8	30	2, 3	Unbalanced supplier locations (6-2)
$\mathscr{S}_2^S$	64	8	30	2, 3	Unbalanced supplier locations (8-0)
$\mathscr{S}_1^C$	64	8	30	2, 3	Unbalanced customer locations (5-10, with $\delta = -5, 30$)
$\mathscr{S}_2^C$	64	8	30	2, 3	Unbalanced customer locations (5-10, with $\delta = 10, 30$)
$\mathscr{S}_3^C$	64	8	30	2, 3	Unbalanced customer locations (10-5, with $\delta = -5, 30$)
$\mathscr{S}_4^C$	64	8	30	2, 3	Unbalanced customer locations (10-5, with $\delta = 10, 30$)
$\mathscr{S}^O$	32	8	30	2	Unbalanced available amounts at the suppliers
$\mathscr{S}_1^{S_{add}}$	64	10	30	2, 3	Increased number of suppliers to 10
$\mathscr{S}_2^{S_{add}}$	64	12	30	2, 3	Increased number of suppliers to 12
$\mathscr{S}_1^{C_{add}}$	64	8	50	2, 3	Increased number of customers to 50
$\mathscr{S}_2^{C_{add}}$	64	8	70	2, 3	Increased number of customers to 70
Small	36	4, 6	10, 15, 20, 25	2, 3	Small instances

6.1. Benchmark instances

Gu et al. (2022) introduced artificial instances as well as instances arising from a real-world case study in the context of a short and local fresh food supply chain. In the following computational experiments, we only consider the artificial instances. Indeed, the sizes of the instances based on the case study are too large to be tackled efficiently with the BPC algorithm.

First, Gu et al. (2022) generated a base set of 64 artificial instances $\mathscr{S}$ with two distribution centres ($|D| = 2$), eight suppliers ($|S| = 8$) and 30 customers ($|C| = 30$). The features of the delivery echelon are based on the 64 small instances proposed in Archetti et al. (2016) for the C-SDVRP. Each C-SDVRP instance gives rise to a MC2DP instance where the locations of one distribution centre and 15 customers are the ones of the C-SDVRP instance. Such distribution centre and 15 customers are duplicated and their locations are modified by applying a translation of parameter $\delta = (30, 30)$ to their coordinates. Customer demands are also as in the C-SDVRP instance. Four suppliers are randomly located around each distribution centre. The availability of each commodity at the suppliers is calculated as a fraction of the total demand. The commodity availabilities are the same for all the suppliers.

Then, Gu et al. (2022) produced 12 additional sets of instances by applying modifications to one of the characteristics of the base set, such as the suppliers/customers locations, the number of suppliers/customers or the available quantities at the suppliers. In all sets of instances, the number of distribution centres is fixed at two. In Table 1, we summarize the main characteristics of all sets of instances. Each row of the table represents a set of instances. The columns of the table report: *set*: the name of the set of instances; #: the number of instances in the set; $|S|$: the number of suppliers; $|C|$: the number of customers; $|\mathcal{K}|$: the number of commodities; *description*: a brief description of the main characteristic of the set. In such an entry, we write $n_1 - n_2$ to express the distribution of the suppliers/customers around each distribution centre, meaning that n_1 suppliers/customers are located around one distribution centre and n_2 are located around the other one. Parameter δ is a translation parameter used to determine the locations of the customers/suppliers around the two distribution centres. We refer to Gu et al. (2022) for further details regarding the generation of the set of instances.

6.2. Impact of valid inequalities

In this section, we assess the impact of valid inequalities. To do so, we consider the 32 instances of base set $\mathscr{S}$ having three commodities. Indeed, as mentioned in Section 4.1, if the number of commodities is equal to two, inequalities (12), (16) and (17) can be retrieved as an aggregation of Covering Constraints (4).

We examine the following four variants of the BPC algorithm. BPC: valid inequalities on bounds on the number of vehicles are inserted, and no valid inequalities is separated in the course of the algorithm; BPC+CC: only capacity cuts (valid inequalities (11)) are separated; BPC+SC+NP: only the inequalities arising from the set covering polytope (SC), i.e., inequalities (12), and the ones arising from the number partitioning polytope (NP) are separated, i.e., inequalities (16) and (17), are separated; BPC+CC+SC+NP: all valid inequalities are separated.

Each row of Table 2 corresponds to a BPC variant. The first two columns report the average lower bound (*avg.LB*) and time (*avg.t[s]*) at the root node of the branch-and-bound-tree. The next four columns show the results at the end of the execution of the corresponding BPC variant: the average number of nodes of the branch-and-bound tree (*avg.#nodes*), the average lower bound at termination (*avg.LB*) the average time (*avg.t[s]*) and the number of instances solved to optimality (*#opt./#inst.*) over the 32 instances.

As expected, BPC yields the worst results solving only six instances out of the 32 considered. Variant BPC+SC+NP solves an additional instance w.r.t. BPC, however, the improvement of the lower bound at the root node is mediocre. The best results are obtained when the well-established capacity cuts are separated, namely with variants BPC+CC and BPC+CC+SC+NP. Both variants solve the same 14 instances to optimality and yield the best lower bounds at the root node, being on average equal to 1000.35 and 1001.31 in BPC+CC and BPC+CC+SC+NP, respectively. The same remark applies to the lower bounds at termination which is on average equal to 1039.61 in BPC+CC and BPC+CC+SC+NP. In both cases, lower bounds at the root node and at termination improve significantly with respect to BPC. We also observe that the addition of inequalities (12), (16) and (17) in BPC+CC+SC+NP slightly improves the results with respect to BPC+CC in terms of lower bounds at the root node, number of explored branch-and-bound nodes and solution time. Hence, we choose BPC+CC+SC+NP as the configuration for our BPC algorithm.

6.3. Evaluation of the BPC algorithm

The aim of this section is to evaluate the effectiveness of the BPC algorithm. To do so, we compare the results obtained by the BPC algorithm on the instances of set small with the ones obtained by solving a compact formulation for the MC2DP on the same instances with CPLEX 12.8. The latter results are retrieved from Gu et al. (2022) and were obtained on a machine with Intel (R) Core(TM) i7-4600U

Table 2
Comparison of four variants of the BPC algorithm.

BPC variant	Root node		BPC results			
	avg.LB	avg.t [s]	avg.#nodes	avg.LB	avg.t [s]	#opt./#inst.
BPC	983.00	25.30	3235.72	1028.08	3145.03	6/32
BPC+CC	1000.35	54.51	1721.53	1039.61	2484.34	14/32
BPC+SC+NP	985.36	46.19	2970.78	1029.42	2992.08	7/32
BPC+CC+SC+NP	1001.31	76.40	1623.84	1039.61	2454.58	14/32

Table 3
Results on set `small`.

Instances					BPC					CPLEX	
$\|S\|$	$\|C\|$	$\|\mathcal{K}\|$	p	Set	#nodes	LB	UB	Gap [%]	t [s]	Gap [%]	t [s]
4	10	2	0.6	$\mathscr{S}$	161	394.655	394.655	opt	5.26	opt	240
		2	1	$\mathscr{S}$	17	579.522	579.522	opt	0.98	5.55	3600
		3	0.6	$\mathscr{S}$	7787	470.77	470.77	opt	1446.16	opt	378
		2	0.6	$\mathscr{S}^S_1$	101	406.52	406.52	opt	4.28	opt	108
		2	1	$\mathscr{S}^S_1$	11	562.34	562.34	opt	1.19	opt	1441
		3	0.6	$\mathscr{S}^S_1$	227	437.98	437.98	opt	14.10	opt	486
		2	0.6	$\mathscr{S}^S_2$	23	406.52	406.52	opt	1.11	opt	57
		2	1	$\mathscr{S}^S_2$	13	663.52	663.52	opt	0.66	opt	2261
		3	0.6	$\mathscr{S}^S_2$	19	463.58	463.58	opt	0.80	opt	41
4	15	2	0.6	$\mathscr{S}$	181	510.88	510.88	opt	10.75	8.45	3600
		2	1	$\mathscr{S}$	15	742.71	742.71	opt	1.37	17.89	3600
		3	0.6	$\mathscr{S}$	13 785	551.87	551.87	opt	3469.76	6.87	3600
		2	0.6	$\mathscr{S}^S_1$	195	533.43	533.43	opt	13.98	13.82	3600
		2	1	$\mathscr{S}^S_1$	3185	784.05	784.05	opt	349.18	15.79	3600
		3	0.6	$\mathscr{S}^S_1$	3829	553.49	553.49	opt	624.91	20.42	3600
		2	0.6	$\mathscr{S}^S_2$	33	590.55	590.55	opt	2.32	7.25	3600
		2	1	$\mathscr{S}^S_2$	17	893.09	893.09	opt	1.67	20.62	3600
		3	0.6	$\mathscr{S}^S_2$	117	590.71	590.71	opt	11.41	15.26	3600
4	20	2	0.6	$\mathscr{S}$	45	636.71	636.71	opt	7.64	22.09	3600
		2	1	$\mathscr{S}$	233	1007.04	1007.04	opt	24.61	31.75	3600
		3	0.6	$\mathscr{S}$	63	708.62	708.62	opt	24.26	28.15	3600
		2	0.6	$\mathscr{S}^S_1$	139	659.37	659.37	opt	35.82	37.59	3600
		2	1	$\mathscr{S}^S_1$	13 848	1069.54	1077.43	0.74	3631.00	30.18	3600
		3	0.6	$\mathscr{S}^S_1$	1209	768.57	768.57	opt	528.33	37.92	3600
		2	0.6	$\mathscr{S}^S_2$	43	713.16	713.16	opt	7.29	27.87	3600
		2	1	$\mathscr{S}^S_2$	5	1177.46	1177.46	opt	0.98	43.37	3600
		3	0.6	$\mathscr{S}^S_2$	157	835.00	835.00	opt	74.17	37.47	3600
6	25	2	0.6	$\mathscr{S}$	209	815.15	815.15	opt	70.15	–	3600
		2	1	$\mathscr{S}$	661	1184.62	1184.62	opt	227.48	31.32	3600
		3	0.6	$\mathscr{S}$	555	826.12	826.12	opt	334.55	58.7	3600
		2	0.6	$\mathscr{S}^S_1$	287	784.11	784.11	opt	70.08	37.76	3600
		2	1	$\mathscr{S}^S_1$	1111	1258.91	1258.91	opt	310.47	54.77	3600
		3	0.6	$\mathscr{S}^S_1$	4073	876.72	908.00	3.57	3632.66	87.9	3600
		2	0.6	$\mathscr{S}^S_2$	35	881.02	881.02	opt	9.95	–	3600
		2	1	$\mathscr{S}^S_2$	99	1367.61	1367.61	opt	29.11	60.19	3600
		3	0.6	$\mathscr{S}^S_2$	15	939.52	939.52	opt	18.84	65.95	3600

processor with a base clock frequency of 2.10 GHz and with 16 GB of RAM. A time limit of one hour is imposed on both methods.

Table 3 presents the results of the comparison. Each row of the table corresponds to an instance in set `small`. The first five columns report some characteristics of the instance (see Section 6.1). The following five columns report the results of the BPC algorithm: *#nodes*: number of nodes of the branch-and-bound tree; *LB*: lower bound at termination; *UB*: value of the best solution found; *gap[%]*: percentage optimality gap ($100((UB - LB)/LB)$) if the instance is not solved to optimality, *opt* otherwise; *t[s]*: total computational time in seconds of the BPC algorithm. Last, in the last two columns, we report the optimality gap (*gap[%]*) and computational time (*t[s]*) obtained by Gu et al. (2022) when solving the compact formulation. In column *gap[%]*, a '-' indicates that CPLEX was not able to provide a feasible solution.

The results of Table 3 show that the BPC algorithm proved to be effective as it provides 34 optima over 36 instances. The two unsolved instances are with 20 and 25 customers and are left with an optimality gap of 0.74% and 3.57%, respectively. Conversely, the performance of the compact formulation deteriorates as the size of the instances grows. The formulation provides only eight optima, all obtained for

Table 4
Aggregated results on the instances solved to optimality by the BPC algorithm.

Instances			BPC results								
				#nodes					dev. Gu et al. (2022)		
Set	$\|\mathcal{K}\|$	#	#opt.	avg.	min.	max.	avg.gap^{root} [%]	avg.t [s]	avg.	min.	max.
$\mathscr{S}$	2	32	31	791.19	11	7037	3.50	241.78	−0.19	−2.21	0.00
	3	32	14	751.79	23	2197	4.24	921.14	−0.95	−2.61	0.00
$\mathscr{S}_1^S$	2	32	30	2126.47	47	21 125	4.90	621.02	−0.19	−1.69	0.00
	3	32	15	872.00	29	2674	4.51	837.63	−1.02	−3.08	0.00
$\mathscr{S}_2^S$	2	32	26	3350.61	51	18 367	4.93	768.70	−1.65	−5.31	0.00
	3	32	20	1932.25	47	12 259	4.95	1417.92	−2.11	−6.71	0.00
$\mathscr{S}_1^C$	2	32	29	667.83	13	2669	3.82	430.16	−0.61	−2.94	0.00
	3	32	15	600.13	39	1547	3.48	872.74	−1.13	−4.04	0.00
$\mathscr{S}_2^C$	2	32	27	674.85	5	3213	4.08	449.58	−0.66	−3.00	0.00
	3	32	15	780.53	81	1855	3.98	1058.08	−1.01	−3.63	0.00
$\mathscr{S}_3^C$	2	32	24	2627.50	25	41 105	4.58	747.43	−0.65	−3.64	0.00
	3	32	15	1040.20	141	4269	4.81	1125.06	−0.91	−3.49	0.00
$\mathscr{S}_4^C$	2	32	26	958.85	15	11 217	4.21	471.72	−0.88	−3.70	0.00
	3	32	13	978.23	219	2109	4.06	909.64	−0.83	−3.09	0.00
$\mathscr{S}^O$	2	32	14	5925.57	435	20 135	6.44	1360.41	−2.34	−4.24	0.00
$\mathscr{S}_1^{S_{add}}$	2	32	30	1339.33	79	12 299	5.34	444.41	−0.35	−1.99	0.00
	3	32	17	727.18	87	2247	4.57	904.53	−0.16	−2.55	0.00
$\mathscr{S}_2^{S_{add}}$	2	32	32	1160.37	1	17 735	6.43	284.21	−0.12	−1.77	0.00
	3	32	23	622.13	65	2651	7.07	826.92	−0.09	−1.64	0.00
$\mathscr{S}_1^{C_{add}}$	2	32	16	1900.87	43	5481	3.08	1292.03	−0.23	−1.12	0.00
	3	32	4	404.25	27	728	2.53	875.52	−0.04	−0.16	0.00
$\mathscr{S}_2^{C_{add}}$	2	32	3	2884.33	663	5855	1.68	1159.56	−0.71	−0.97	−0.36
	3	32	0	–	–	–	–	–	–	–	–

instances with 10 customers, and it fails to return a feasible solution for two instances with 25 customers. The average optimality gap of the remaining 26 instances is 31.73%. Finally, when both approaches prove the optimality of a solution, the BPC algorithm is generally faster than the compact formulation by at least one order of magnitude.

6.4. Results on the whole testbed

In this section, we present a summary of the results obtained by the BPC algorithm in Tables 4 and 6 on the 12 sets of benchmark instances with one-hour time limit. The instance-by-instance results can be found at https://hal.inria.fr/hal-03946477v1.

In Table 4, we report results for the instances solved to optimality and Table 6 summarizes the results for the remaining instances. Each row of both tables corresponds to a subset of instances from the same set and with the same number of commodities. The first three columns of the tables report some information about the instance subset (see Section 6.1). The column headings of Table 4 are defined as follows: *#opt.*: number of instances solved to optimality; *#nodes avg./min./max.*: average/minimum/maximum number of nodes of the branch-and-bound tree; *avg.gap^{root} [%]*: average optimality gap at the root node expressed as a percentage, i.e., $100((OPT - LB^{root})/LB^{root})$, where OPT is the value of the optimal solution found by the BPC algorithm and LB^{root} is the lower bound at the root node after the valid inequalities have been inserted; *avg.t[s]*: average computational time; *dev. Gu et al. (2022) avg./min./max.*: average/minimum/maximum deviation from the best solution value UB reported in Gu et al. (2022), i.e., $100((OPT - UB)/UB)$.

Table 4 shows that the BPC algorithm identifies 439 optima over the 736 instances. The number of nodes of the branch-and-bound tree varies widely: it ranges from 1 to 41105 and its average is 1458 while its standard deviation is 3237. Note that we found no correlation between the number of nodes of the branch-and-bound tree and the gap at the root node. The average time needed to prove the optimality of a solution is 720 seconds. Among the 439 optima provided by the BPC algorithm, 416 are obtained on the 10 sets of instances with 30 customers (first ten sets in Table 4).

We observe that, except for set $\mathscr{S}^O$, the BPC algorithm behaves homogeneously on instances with 30 customers, i.e., on $\mathscr{S}$, $\mathscr{S}_1^S$, $\mathscr{S}_2^S$, $\mathscr{S}_1^C$, $\mathscr{S}_2^C$, $\mathscr{S}_3^C$, $\mathscr{S}_4^C$, $\mathscr{S}_1^{S_{add}}$ and $\mathscr{S}_2^{S_{add}}$. Indeed, the BPC algorithm solves to optimality at least 60.9% of the 64 instances belonging to each set. This percentage increases to 73.4% and 85.9% for the two sets of instances with a larger number of suppliers (see $\mathscr{S}_1^{S_{add}}$ and $\mathscr{S}_2^{S_{add}}$ in Table 4). Increasing the number of suppliers seems to make the instances easier to solve. In addition, in each of the sets with 30 customers, the BPC algorithm proves the optimality of almost all the instances with two commodities (at least 24 out of 32) and of around half of the instances with three commodities (on average 16 out of 32). Hence, we can conclude that the BPC algorithm seems rather insensitive with respect to the distinctive characteristics of the sets of instances with 30 customers, i.e., unbalanced customer/supplier locations and an increased number of suppliers. Conversely, increasing the number of customers has a major impact on the performance of the BPC algorithm. Indeed, when the number of customers increases to 50 and 70 (see sets $\mathscr{S}_1^{C_{add}}$ and $\mathscr{S}_2^{C_{add}}$ in Table 4), the number of optima decreases to 20 and 3, respectively.

Finally, we note that the sequential heuristic of Gu et al. (2022) was able to identify 220 out of 439 optima. For the remaining instances, the BPC algorithm improves the solution values found by Gu et al. (2022) by 1.46% on average (see the last three columns of Table 4).

To better assess the difficulty of solving these 439 instances to optimality, in Table 5, we report some statistics about the optimal solutions. Each row corresponds to a subset of the instances. The first three columns respectively report the name of the set of instances (*set*), the number of instances in the set (#) and of those solved to optimality by the BPC algorithm (*#opt*). The next three columns report statistics regarding the collection echelon: *vehicles avg.LB*: average lower bound (right hand-side of Constraint (9)) on the number of vehicles in the collection echelon; *vehicles avg.#*: average number of vehicles used in the collection echelon; *avg.#suppl. visits*: average number of visits to suppliers. The last four columns report statistics regarding the delivery echelon: *vehicles avg.LB*: average lower bound (right hand-side of Constraint (10a)) on the number of vehicles used in the delivery echelon; *vehicles avg.#*: average number of vehicles used in the delivery echelon;

Table 5
Statistics about the optimal solutions.

Instances			Collection			Delivery				
			Vehicles			Vehicles		Avg.customers visits [%]		
Set	#	#opt	avg.LB	avg.#	avg.#suppl. visits	avg.LB	avg.#	1	2	3
$\mathscr{S}$	64	45	12.87	13.78	1.74	13.27	14.02	89.04	10.96	0.00
$\mathscr{S}_1^S$	64	45	12.58	13.62	1.81	12.98	13.67	88.44	11.41	0.15
$\mathscr{S}_2^S$	64	46	11.93	12.89	1.74	12.24	12.70	87.90	11.88	0.22
$\mathscr{S}_1^C$	64	44	11.75	12.57	1.61	11.95	12.82	89.85	10.15	0.00
$\mathscr{S}_2^C$	64	42	11.57	12.43	1.58	11.79	12.62	90.32	9.68	0.00
$\mathscr{S}_3^C$	64	39	12.13	13.00	1.68	12.54	13.38	89.49	10.43	0.09
$\mathscr{S}_4^C$	64	39	12.82	13.62	1.74	13.26	14.15	88.63	11.03	0.34
$\mathscr{S}^O$	32	14	12.71	13.86	1.73	13.64	14.29	76.43	23.57	0.00
$\mathscr{S}_1^{S_{add}}$	64	47	12.85	13.98	1.44	13.23	13.94	87.80	12.20	0.00
$\mathscr{S}_2^{S_{add}}$	64	55	12.22	13.84	1.28	12.55	13.36	90.12	9.88	0.00
$\mathscr{S}_1^{C_{add}}$	64	20	22.40	23.10	3.19	23.40	24.85	89.10	10.80	0.10
$\mathscr{S}_2^{C_{add}}$	64	3	43.33	44.00	6.10	49.33	52.00	98.10	1.90	0.00

Table 6
Aggregated results on the instances not solved to proven optimality by the BPC algorithm.

Instances			BPC results									
				#nodes			Gap [%]				dev. Gu et al. (2022)	
Set	$\|\mathcal{K}\|$	#	#notOpt.	avg.	min.	max.	avg.	min.	max.	avg.gap^{root} [%]	avg.	min.	max.
$\mathscr{S}$	2	32	1	653.00	653	653	3.31	3.31	3.31	14.08	2.07	2.07	2.07
	3	32	18	2319.17	29	13 135	2.44	0.01	6.46	6.80	0.92	−1.95	5.85
$\mathscr{S}_1^S$	2	32	2	3880.00	940	6820	3.38	0.41	6.35	11.97	1.01	−0.06	2.09
	3	32	17	3057.65	19	16 126	3.01	0.20	10.01	8.82	0.91	−2.67	5.10
$\mathscr{S}_2^S$	2	32	6	12 069.00	941	46 006	0.37	0.06	1.03	6.92	−1.15	−3.61	0.07
	3	32	12	3212.42	40	13 254	1.87	0.01	4.71	6.69	−1.30	−5.89	3.67
$\mathscr{S}_1^C$	2	32	3	29 380.33	12 463	40 874	0.98	0.03	1.77	4.09	−0.58	−0.90	−0.29
	3	32	17	1913.18	1	10 229	2.61	0.34	7.11	6.68	0.16	−1.78	3.36
$\mathscr{S}_2^C$	2	32	5	18 695.80	7	40 567	0.78	0.03	2.82	3.55	−0.43	−1.00	0.02
	3	32	17	1856.53	23	9311	2.58	0.21	6.09	7.00	0.08	−1.67	2.86
$\mathscr{S}_3^C$	2	32	8	4371.88	380	12 552	1.09	0.12	2.25	5.34	0.17	−0.92	1.94
	3	32	17	2070.71	29	12 984	2.72	0.03	10.89	7.65	0.85	−1.31	10.69
$\mathscr{S}_4^C$	2	32	6	4876.50	342	12 698	1.09	0.10	1.94	5.78	0.11	−1.13	1.62
	3	32	19	1463.68	10	12 005	2.47	0.18	5.96	7.23	0.41	−1.68	2.26
$\mathscr{S}^O$	2	32	18	9788.83	1033	50 109	2.24	0.13	9.17	8.15	0.05	−2.98	4.85
$\mathscr{S}_1^{S_{add}}$	2	32	2	774.50	763	786	2.37	1.05	3.69	18.86	1.32	0.69	1.96
	3	32	15	2484.00	37	12 241	2.60	0.00	8.76	10.65	0.65	−1.77	2.40
$\mathscr{S}_2^{S_{add}}$	2	32	0	–	–	–	–	–	–	–	–	–	–
	3	32	9	2990.11	16	14 444	1.99	0.07	6.03	9.21	0.75	0.00	4.05
$\mathscr{S}_1^{C_{add}}$	2	32	16	4300.00	180	32 352	1.15	0.00	3.52	6.27	0.25	−0.84	2.11
	3	32	28	1344.14	1	6585	2.60	0.08	10.19	6.06	0.65	−1.20	4.63
$\mathscr{S}_2^{C_{add}}$	2	32	29	2141.07	337	14 413	1.15	0.04	3.78	3.64	−0.18	−1.83	2.98
	3	32	32	684.22	1	2820	2.45	0.04	9.31	3.97	0.26	−2.32	2.93

avg. customers visits[%]: average percentage of the customers visited one, two or three times.

The number of vehicles used in the collection echelon is rather tight to the lower bound. Suppliers are visited on average between one and two times, exception made for the instances with additional customers where suppliers are visited 3.19 and 6.10 times, on average. In the delivery echelon, the number of vehicles is also rather tight to the lower bound, except for the instances of set $\mathscr{S}_2^{C_{add}}$. In each set of instances, at least 76.43% of the customers in the instances is served with one visit, i.e., with no splits. Around 10% of the customers is visited twice, except for the instances of sets $\mathscr{S}^O$ and $\mathscr{S}_2^{C_{add}}$ where such percentage increases to 23.57 and decreases to 1.90, respectively. Note that in the latter case the information may not be significant as the average is computed considering only three instances. The percentage of customers visited three times is negligible.

Table 6 reports the results on the instances not solved to proven optimality by the BPC algorithm. The meaning of the rows and columns in Table 6 is similar to the ones of Table 4. The differences are: the column *#opt.* is replaced with the column *#notOpt.* which indicates the number of instances not solved to proven optimality, and the column *avg.t[s]* is replaced by the columns *gap[%] avg./min./max.* reporting the average, minimum, and maximum optimality gap computed as $100((UB - LB)/LB)$, where UB is the value of the best solution found by the BPC algorithm and LB is the lower bound when the time limit is reached. Similarly, the average gap at the root node *avg.gap^{root}[%]*: and the average deviation from the best solution value reported in Gu et al. (2022) are computed by replacing the optimal value OPT with UB.

The BPC algorithm cannot prove the optimality for 297 instances. For these instances, the average optimality gap at the root node is 6.63%. However, the exploration of the branch-and-bound tree allows the optimality gap to be reduced to 2.1% on average. The final optimality gap is larger than 5% for only 29 instances. The number of nodes of the branch-and-bound tree follows the trend observed in Table 4: it varies greatly, with an average of 3429 nodes, while the standard deviation is 6809. The comparison with Gu et al. (2022) (last three columns of Table 6) shows mixed results. The BPC algorithm finds larger upper bounds for 161 instances. On these instances, the average deviation is 1.23%. For 24 instances, the BPC algorithm finds the same value as the one reported by Gu et al. (2022). Finally, for

the remaining 112 instances, the BPC algorithm provides a lower value with an average improvement of 1.02%.

7. Conclusions

In this paper, we presented a Branch-Price-and-Cut (BPC) algorithm to solve the Multi-Commodity two-echelon Distribution Problem (MC2DP), a two-echelon routing problem where multiple commodities are sent from suppliers to customers via distribution centres. The collection operations are done by capacitated vehicles performing direct round trips between the distribution centres and the suppliers. The delivery operations are also performed by capacitated vehicles. Each delivery vehicle performs a route starting and ending at the same distribution centre. Customers are allowed to be visited multiple times, provided that the amount of a single commodity is delivered at once by a single vehicle. Commodities can be mixed inside all vehicles. The objective is to minimize the transportation costs of the distribution system.

The BPC algorithm incorporates several state-of-the-art accelerating techniques and three families of robust valid inequalities: capacity cuts, valid inequalities arising from the set covering polytope, and a new family of valid inequalities based on the number partitioning polytope. The inequalities improve the lower bound at the root node and reduce the number of nodes of the branch-and-bound tree and the computational time. The BPC algorithm is able to solve to optimality nearly 60% of the benchmark instances introduced in Gu et al. (2022) within one-hour time limit. The final optimality gap is reasonable for the remaining instances, with an average value of 2.1%. Finally, we identified 331 new best-known solutions compared to the results of Gu et al. (2022).

The main issue with the instances left unsolved by the BPC algorithm is the large optimality gap at the root node. To overcome this difficulty, future research should be devoted to the inclusion of new dedicated valid inequalities. Adding non-robust valid inequalities known for routing problems is also an interesting perspective. However, it would lead to more difficult pricing problems to solve. In addition, Gu et al. (2022) proposed a sequential heuristic for the MC2DP. Therefore, another line of research could be the development of heuristic algorithms that address the problem from an integrated point of view.

CRediT authorship contribution statement

Matteo Petris: Writing – review & editing, Writing – original draft, Visualization, Validation, Supervision, Software, Resources, Project administration, Methodology, Investigation, Funding acquisition, Formal analysis, Data curation, Conceptualization. **Claudia Archetti:** Writing – review & editing, Writing – original draft, Visualization, Validation, Supervision, Software, Resources, Project administration, Methodology, Investigation, Funding acquisition, Formal analysis, Data curation, Conceptualization. **Diego Cattaruzza:** Writing – review & editing, Writing – original draft, Visualization, Validation, Supervision, Software, Resources, Project administration, Methodology, Investigation, Funding acquisition, Formal analysis, Data curation, Conceptualization. **Maxime Ogier:** Writing – review & editing, Writing – original draft, Visualization, Validation, Supervision, Software, Resources, Project administration, Methodology, Investigation, Funding acquisition, Formal analysis, Data curation, Conceptualization. **Frédéric Semet:** Writing – review & editing, Writing – original draft, Visualization, Validation, Supervision, Software, Resources, Project administration, Methodology, Investigation, Funding acquisition, Formal analysis, Data curation, Conceptualization.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

Achterberg, T., 2007. Constraint Integer Programming (Ph.D. thesis). Technische Universitat Berlin.

Archetti, C., Bianchessi, N., Speranza, M., 2013. Optimal solutions for routing problems with profits. Discrete Appl. Math. 161, 547–557. http://dx.doi.org/10.1016/j.dam.2011.12.021.

Archetti, C., Bianchessi, N., Speranza, M.G., 2015. A branch-price-and-cut algorithm for the commodity constrained split delivery vehicle routing problem. Comput. Oper. Res. 64, 1–10. http://dx.doi.org/10.1016/j.cor.2015.04.023, URL: http://www.sciencedirect.com/science/article/pii/S0305054815001148.

Archetti, C., Bouchard, M., Desaulniers, G., 2011. Enhanced branch and price and cut for vehicle routing with split deliveries and time windows. Transp. Sci. 45, 285–298.

Archetti, C., Campbell, A.M., Speranza, M.G., 2016. Multicommodity vs. single-commodity routing. Transp. Sci. 50, 461–472. http://dx.doi.org/10.1287/trsc.2014.0528.

Balas, E., Ng, S.M., 1989. On the set covering polytope: I. all the facets with coefficients in {0, 1, 2}. Math. Program. 43, 57–69. http://dx.doi.org/10.1007/bf01582278.

Baldacci, R., Mingozzi, A., Roberti, R., 2011. New route relaxation and pricing strategies for the vehicle routing problem. Oper. Res. 59, 1269–1283. http://dx.doi.org/10.1287/opre.1110.0975.

Barnhart, C., Johnson, E.L., Nemhauser, G.L., Savelsbergh, M.W.P., Vance, P.H., 1998. Branch-and-price: Column generation for solving huge integer programs. Oper. Res. 46, 316–329. http://dx.doi.org/10.1287/opre.46.3.316.

Battarra, M., Golden, B., Vigo, D., 2008. Tuning a parametric Clarke–Wright heuristic via a genetic algorithm. J. Oper. Res. Soc. 59, 1568–1572. http://dx.doi.org/10.1057/palgrave.jors.2602488.

Berti, G., Mulligan, C., 2016. Competitiveness of small farms and innovative food supply chains: The role of food hubs in creating sustainable regional and local food systems. Sustainability 8 (616).

Bode, C., Irnich, S., 2012. Cut-first branch-and-price-second for the capacitated arc-routing problem. Oper. Res. 60, 1167–1182. http://dx.doi.org/10.1287/opre.1120.1079.

Breunig, U., Baldacci, R., Hartl, R., Vidal, T., 2019. The electric two-echelon vehicle routing problem. Comput. Oper. Res. 103, 198–210. http://dx.doi.org/10.1016/j.cor.2018.11.005.

Cattaruzza, D., Absi, N., Feillet, D., González-Feliu, J., 2017. Vehicle routing problems for city logistics. EURO J. Transp. Logist. 6, 51–79. http://dx.doi.org/10.1007/s13676-014-0074-0.

Christof, T., Löbel, A., 2009. Polyhedron representation transformation algorithm. URL: http://porta.zib.de/#about.

Clarke, G., Wright, J.W., 1964. Scheduling of vehicles from a central depot to a number of delivery points. Oper. Res. 12, 568–581. http://dx.doi.org/10.1287/opre.12.4.568.

Contardo, C., Cordeau, J.F., Gendron, B., 2014. An exact algorithm based on cut-and-column generation for the capacitated location-routing problem. INFORMS J. Comput. 26, 88–102. http://dx.doi.org/10.1287/ijoc.2013.0549.

Crainic, T.G., Feliu, J.G., Ricciardi, N., Semet, F., Van Woensel, T., 2023. 10. Operations research for planning and managing city logistics systems. In: Handbook on City Logistics and Urban Freight: 0. pp. 190–223.

Crainic, T.G., Ricciardi, N., Storchi, G., 2004. Advanced freight transportation systems for congested urban areas. Transp. Res. C 12, 119–137. http://dx.doi.org/10.1016/j.trc.2004.07.002.

Crainic, T.G., Ricciardi, N., Storchi, G., 2009. Models for evaluating and planning city logistics systems. Transp. Sci. 43, 432–454. http://dx.doi.org/10.1287/trsc.1090.0279.

Cuda, R., Guastaroba, G., Speranza, M., 2015. A survey on two-echelon routing problems. Comput. Oper. Res. 55, 185–199. http://dx.doi.org/10.1016/j.cor.2014.06.008.

Dellaert, N., Woensel, T.V., Crainic, T.G., Saridarq, F.D., 2021. A multi-commodity two-echelon capacitated vehicle routing problem with time windows: Model formulations and solution approach. Comput. Oper. Res. 127, 105154. http://dx.doi.org/10.1016/j.cor.2020.105154.

Desaulniers, G., 2010. Branch-and-price-and-cut for the split-delivery vehicle routing problem with time windows. Oper. Res. 58, 179–192. http://dx.doi.org/10.1287/opre.1090.0713.

Desrosiers, J., Lübbecke, M.E., 2005. A primer in column generation. In: Column Generation. Springer-Verlag, pp. 1–32. http://dx.doi.org/10.1007/0-387-25486-2_1.

Drexl, M., 2012. Synchronization in vehicle routing—a survey of VRPs with multiple synchronization constraints. Transp. Sci. 46, 297–316. http://dx.doi.org/10.1287/trsc.1110.0400.

Enthoven, D.L., Jargalsaikhan, B., Roodbergen, K.J., uit het Broek, M.A., Schrotenboer, A.H., 2020. The two-echelon vehicle routing problem with covering options: City logistics with cargo bikes and parcel lockers. Comput. Oper. Res. 118, 104919. http://dx.doi.org/10.1016/j.cor.2020.104919.

Federgruen, A., Simchi-Levi, D., 1995. Analysis of vehicle routing and inventory-routing problems. Handb. Oper. Res. Manag. Sci. 8, 297–373.

Feillet, D., Dejax, P., Gendreau, M., Gueguen, C., 2004. An exact algorithm for the elementary shortest path problem with resource constraints: Application to some vehicle routing problems. Networks 44, 216–229. http://dx.doi.org/10.1002/net.20033.

Grangier, P., Gendreau, M., Lehuédé, F., Rousseau, L.M., 2016. An adaptive large neighborhood search for the two-echelon multiple-trip vehicle routing problem with satellite synchronization. European J. Oper. Res. 254, 80–91. http://dx.doi.org/10.1016/j.ejor.2016.03.040.

Gschwind, T., Bianchessi, N., Irnich, S., 2019. Stabilized branch-price-and-cut for the commodity-constrained split delivery vehicle routing problem. European J. Oper. Res. 278, 91–104. http://dx.doi.org/10.1016/j.ejor.2019.04.008.

Gu, W., Archetti, C., Cattaruzza, D., Ogier, M., Semet, F., Speranza, M.G., 2022. A sequential approach for a multi-commodity two-echelon distribution problem. Comput. Ind. Eng. 163, 107793. http://dx.doi.org/10.1016/j.cie.2021.107793.

Gu, W., Cattaruzza, D., Ogier, M., Semet, F., 2019. Adaptive large neighborhood search for the commodity constrained split delivery VRP. Comput. Oper. Res. 112, 104761. http://dx.doi.org/10.1016/j.cor.2019.07.019.

Guastaroba, G., Speranza, M.G., Vigo, D., 2016. Intermediate facilities in freight transportation planning: A survey. Transp. Sci. 50, 763–789. http://dx.doi.org/10.1287/trsc.2015.0631.

Jacobsen, S., Madsen, O., 1980. A comparative study of heuristics for a two-level routing-location problem. European J. Oper. Res. 5, 378–387. http://dx.doi.org/10.1016/0377-2217(80)90124-1.

Jia, S., Deng, L., Zhao, Q., Chen, Y., 2023. An adaptive large neighborhood search heuristic for multi-commodity two-echelon vehicle routing problem with satellite synchronization. J. Ind. Manage. Optim. 19, 1187–1210. http://dx.doi.org/10.3934/jimo.2021225.

Jie, W., Yang, J., Zhang, M., Huang, Y., 2019. The two-echelon capacitated electric vehicle routing problem with battery swapping stations: Formulation and efficient methodology. European J. Oper. Res. 272, 879–904. http://dx.doi.org/10.1016/j.ejor.2018.07.002.

Laporte, G., Nobert, Y., Desrochers, M., 1985. Optimal routing under capacity and distance restrictions. Oper. Res. 33, 1050–1073. http://dx.doi.org/10.1287/opre.33.5.1050.

Li, H., Chen, J., Wang, F., Bai, M., 2021a. Ground-vehicle and unmanned-aerial-vehicle routing problems from two-echelon scheme perspective: A review. European J. Oper. Res. 294, 1078–1095. http://dx.doi.org/10.1016/j.ejor.2021.02.022.

Li, H., Liu, Y., Jian, X., Lu, Y., 2018. The two-echelon distribution system considering the real-time transshipment capacity varying. Transp. Res. B 110, 239–260. http://dx.doi.org/10.1016/j.trb.2018.02.015.

Li, H., Wang, H., Chen, J., Bai, M., 2020. Two-echelon vehicle routing problem with time windows and mobile satellites. Transp. Res. B 138, 179–201. http://dx.doi.org/10.1016/j.trb.2020.05.010.

Li, H., Wang, H., Chen, J., Bai, M., 2021b. Two-echelon vehicle routing problem with satellite bi-synchronization. European J. Oper. Res. 288, 775–793. http://dx.doi.org/10.1016/j.ejor.2020.06.019.

Li, J., Xu, M., Sun, P., 2022. Two-echelon capacitated vehicle routing problem with grouping constraints and simultaneous pickup and delivery. Transp. Res. B 162, 261–291. http://dx.doi.org/10.1016/j.trb.2022.06.003.

Lörwald, S., Reinelt, G., 2015. Panda: a software for polyhedral transformations. EURO J. Comput. Optim. 3, 297–308.

Magnanti, T.L., Wong, R.T., 1984. Network design and transportation planning: Models and algorithms. Transp. Sci. 18, 1–55. http://dx.doi.org/10.1287/trsc.18.1.1.

Marques, G., Sadykov, R., Deschamps, J.C., Dupas, R., 2020. An improved branch-cut-and-price algorithm for the two-echelon capacitated vehicle routing problem. Comput. Oper. Res. 114, 104833. http://dx.doi.org/10.1016/j.cor.2019.104833.

Marques, G., Sadykov, R., Dupas, R., 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. Transp. Sci. http://dx.doi.org/10.1287/trsc.2022.1136.

Mhamedi, T., Andersson, H., Cherkesly, M., Desaulniers, G., 2022. A branch-price-and-cut algorithm for the two-echelon vehicle routing problem with time windows. Transp. Sci. 56, 245–264. http://dx.doi.org/10.1287/trsc.2021.1092.

Perboli, G., Tadei, R., Vigo, D., 2011. The two-echelon capacitated vehicle routing problem: Models and math-based heuristics. Transp. Sci. 45, 364–380. http://dx.doi.org/10.1287/trsc.1110.0368.

Pessoa, A., Sadykov, R., Uchoa, E., Vanderbeck, F., 2020. A generic exact solver for vehicle routing and related problems. Math. Program. 183, 483–523. http://dx.doi.org/10.1007/s10107-020-01523-z.

Petris, M., Archetti, C., Cattaruzza, D., Ogier, M., Semet, F., 2023. A heuristic with a performance guarantee for the commodity constrained split delivery vehicle routing problem. URL: https://hal.inria.fr/hal-03924873. working paper or preprint.

Ralphs, T., Kopman, L., Pulleyblank, W., Trotter, L., 2003. On the capacitated vehicle routing problem. Math. Program. 94, 343–359. http://dx.doi.org/10.1007/s10107-002-0323-0.

Righini, G., Salani, M., 2006. Symmetry helps: Bounded bi-directional dynamic programming for the elementary shortest path problem with resource constraints. Discrete Optim. 3, 255–273. http://dx.doi.org/10.1016/j.disopt.2006.05.007.

Ryan, D.M., Foster, B.A., 1981. An integer programming approach to scheduling. In: Computer Scheduling of Public Transport Urban Passenger Vehicle and Crew Scheduling. pp. 269–280.

S., Røpke, 2012. Branching decisions in branch-and-cut-and-price algorithms for vehicle routing problems. In: Presentation in Column Generation 2012.

Sluijk, N., Florio, A.M., Kinable, J., Dellaert, N., Van Woensel, T., 2022. A chance-constrained two-echelon vehicle routing problem with stochastic demands. Transp. Sci. http://dx.doi.org/10.1287/trsc.2022.1162.

Sluijk, N., Florio, A.M., Kinable, J., Dellaert, N., Woensel, T.V., 2023. Two-echelon vehicle routing problems: A literature review. European J. Oper. Res. 304, 865–886. http://dx.doi.org/10.1016/j.ejor.2022.02.022.

Zhou, L., Baldacci, R., Vigo, D., Wang, X., 2018. A multi-depot two-echelon vehicle routing problem with delivery options arising in the last mile distribution. European J. Oper. Res. 265, 765–778. http://dx.doi.org/10.1016/j.ejor.2017.08.011.