CIRRELT
1971 – 2021

CIRRELT-2022-05

A Novel Reformulation for the Single-Sink Fixed-Charge Transportation Problem

Robin Legault
Jean-François Côté
Bernard Gendron

February 2022

Document de travail également publié par la Faculté des sciences de l'administration de l'Université Laval, sous le numéro FSA-2022-002

Bureau de Montréal
Université de Montréal
C.P. 6128, succ. Centre-Ville
Montréal (Québec) H3C 3J7
Tél : 1-514-343-7575
Télécopie : 1-514-343-7121

Bureau de Québec
Université Laval,
2325, rue de la Terrasse
Pavillon Palasis-Prince, local 2415
Québec (Québec) G1V 0A6
Tél : 1-418-656-2073
Télécopie : 1-418-656-2624

A Novel Reformulation for the Single-Sink Fixed-Charge Transportation Problem

Robin Legault[1,2], Jean-François Côté[1,3,*], Bernard Gendron[1,2]

1. Interuniversity Research Centre on Enterprise Networks, Logistics and Transportation (CIRRELT)
2. Department of Computer Science and Operations Research, Université de Montréal
3. Department of Operations and Decision Systems, Université Laval, Québec, Canada

Abstract. The single-sink fixed-charge transportation problem is known to have many applications in the area of Manufacturing and Transportation as well to be an important subproblem of the fixed-charge transportation problem. However, even the best algorithms from the literature do not succeed in fully exploiting the structure of this problem, to the point of being surpassed by modern general-purpose mixed-integer programming solvers for large instances. We introduce a novel reformulation of the problem and study its theoretical properties. This reformulation leads to a range of new upper and lower bounds as well as specific dominance relations, linear relaxations, and filtering procedures. The resulting algorithm includes a heuristic phase and an exact phase, the main step of which is to solve a very small number of knapsack subproblems. Computational experiments are presented for existing and new types of data instances. These tests indicate that the new algorithm systematically reduces the resolution time of the state-of-the-art exact methods by several orders of magnitude.

Keywords: Integer programming, fixed-charge transportation problem, Lagrangian relaxation, Knapsack problem.

Acknowledgements. This research was supported by the Natural Sciences and Engineering Research Council of Canada through the Discovery Grants 2017-06054 and 2021-00028. We thank A. Klose for making available the source code of his algorithms.

---

* Corresponding author: Jean-Francois.Cote@cirrelt.ca

Dépôt légal – Bibliothèque et Archives nationales du Québec
Bibliothèque et Archives Canada, 2022

1 Introduction

The single-sink fixed-charge transportation problem (SSFCTP) can be stated as a distribution problem in which a single customer acting as a sink and having access to a set of n suppliers $j = 1, \ldots, n$ must acquire D units of the same item. Each supplier j can ship up to b_j units to the sink at a fixed-charge f_j as well as a cost c_j per unit. The problem is to minimize the total cost while respecting the demand D of the sink. The classical formulation of the SSFCTP, that we name P1, is

$$[\text{P1}] \quad Z^1 = \min \sum_{j=1}^{n} (c_j x_j + f_j y_j) \tag{1}$$

$$\text{s.t.} \sum_{j=1}^{n} x_j = D, \tag{2}$$

$$0 \leq x_j \leq b_j y_j \qquad \text{for } j = 1, \ldots, n, \tag{3}$$

$$y_j \in \{0, 1\} \qquad \text{for } j = 1, \ldots, n. \tag{4}$$

The binary decision variable y_j is equal to 1 if the fixed cost is paid on node j. In this case, as specified in constraint (3), the usage variable x_j can take any positive value less than or equal to b_j. The demand constraint is given by equation (2). It is assumed that each supplier offers at least one item, that all the unit and fixed costs are nonnegative and that the total offer suffices to satisfy the demand, i.e. $1 \leq b_j \leq D$, $c_j \geq 0$ and $f_j \geq 0$ for each supplier $j \in \{1, \ldots, n\}$ and $\sum_{j=1}^{n} b_j \geq D$. Furthermore, the demand D as well as the number of shipments x_j to be made from each supplier and the capacities b_j are assumed to be integer-valued.

This $\mathcal{NP}$-hard problem (see [1] for a proof of $\mathcal{NP}$-hardness) has various practical applications [2] and is an important subproblem of the fixed-charge transportation problem (FCTP) [3], [4]. Furthermore, staircase transportation cost functions can be introduced to the SSFCTP, leading to the so-called *single-sink, fixed-charge, multiple-choice transportation problem*, which arises as a relaxation of more general minimum-cost network flow problems [5]. The SSFCTP is also a special case of a relaxation of the multicommodity capacitated fixed-charge network design problem [6].

Two main approaches have been used for the resolution of the SSFCTP in the literature. The first one consists in an implicit enumeration algorithm, proposed by Herer et al. [2], that refines an algorithm from Haberl [7]. The second one is a dynamic programming method introduced by Alidaee and Kochenberger [8]. Both these methods were revisited and significantly improved by Klose [1]. In particular, some of these improvements over the original algorithms directly exploit the similarities between the SSFCTP and the binary knapsack problem (KP) to take advantage of ideas first developed by Martello and Toth [9].

This paper takes a step further in this direction by introducing a new mathematical formulation of the problem which allows expressing both a relaxation and subproblems of the SSFCTP as KPs. The algorithm we introduce will thus be referred to as the *knapsack transformation algorithm* (KTA). It is composed of a heuristic phase, a filtering phase and an exact phase that are executed sequentially.

The different models developed in this article aim to reduce the search space as much as possible while ensuring that at least one optimal solution to the original problem remains acces-

sible. Doing so, the fast algorithms that have been proposed over time for solving the KP and the rich literature that covers many of its variants (see for example [10], [11], [12], [13], [14]) can be fully mobilized to solve the SSFCTP efficiently.

Our paper offers five main contributions that will be detailed in Sections 2, 3, 4, 5 and 7 respectively.

- We introduce model P2, a binary non-linear programming reformulation of the problem. This new mathematical formulation takes advantage of an important property from the literature to exploit the strong similarity which links the SSFCTP to the KP.
- A new heuristic method is proposed. Its main step is to solve knapsack subproblems, noted P3, that are obtained through a relaxation of P2. We present an in-depth study of the relations between specific subsets of solutions to models P1, P2 and P3. These properties are used to extract several useful bounds and to introduce a new dominance relation as well as a strong linear relaxation.
- Two new filtering techniques are introduced. The first method exploits a new strict total order on the suppliers while the second one corresponds to a strengthened linear filtering procedure. Together, they drastically reduce the size of the subproblems to be solved in the last phase of KTA.
- We present a transformation of the subproblem which arises after allowing a unique node to supply a positive number of items that is less than its capacity. This reformulation, which is at the core of the exact phase of KTA, is also expressed as a KP.
- We introduce new instances, with the aim of providing an in-depth analysis of the performance of KTA and the existing techniques of the literature. The structure of problems resulting from various generation methods is studied theoretically and empirically.

The remainder of the paper is structured as follows. Section 6 contains a detailed summary of KTA. In addition to new generation methods, Section 7 presents the results of our numerical experiments, including an analysis of the performance of each step of our algorithm. Section 8 concludes the article.

Since there is a large number of propositions in this work, we have decided to regroup all the proofs in Appendix 1 to lighten the reading.

2 Novel Reformulation

First, let us introduce some definitions that will be useful throughout the article.

Definition 2.1. *A **complete** node j in a solution $(\boldsymbol{x}, \boldsymbol{y})$ to P1 is a node such that $x_j = b_j$.*

Definition 2.2. *A **partial** node j in a solution $(\boldsymbol{x}, \boldsymbol{y})$ to P1 is a node such that $1 \leq x_j \leq b_j - 1$.*

Definition 2.3. *An **unused** node j in a solution $(\boldsymbol{x}, \boldsymbol{y})$ to P1 is a node such that $x_j = 0$.*

From there, the following well-known property of the SSFCTP, of which a proof is given in [15], shall be recalled.

Proposition 2.4. *There exists an optimal solution to P1 which contains at most one partial node. Furthermore, if an optimal solution contains a partial node $p \in \{1,\dots,n\}$, then $y_j = 0$ for each node $j \in \{1,\dots,n\}$ such that $c_j > c_p$.*

The classical interpretation of the SSFCTP is expressed as a decision problem on the number of shipments to be made from each supplier to the sink. This interpretation can be modified to bring the formulation of the SSFCTP even closer to that of the already closely related KP. Using Proposition 2.4, the integer-valued decision variables of P1 can be replaced by binary variables representing the selection of each node in the solution, among which only one can be partially used. This reinterpretation leads to the following new model.

$$\text{[P2]} \quad Z^2 = \min \sum_{j=1}^{n}(c_j b_j + f_j)y_j - \Big(\sum_{j=1}^{n} b_j y_j - D\Big)\sum_{j=1}^{n} c_j z_j \tag{5}$$

$$\text{s.t.} \sum_{j=1}^{n} b_j y_j \geq D, \tag{6}$$

$$\sum_{j=1}^{n} b_j y_j - D \leq \sum_{j=1}^{n} b_j z_j - 1, \tag{7}$$

$$\sum_{j=1}^{n} z_j = 1, \tag{8}$$

$$z_j \leq y_j \qquad \text{for } j = 1,\dots,n, \tag{9}$$

$$y_j, z_j \in \{0,1\} \qquad \text{for } j = 1,\dots,n. \tag{10}$$

In this binary non-linear formulation, a node $j \in \{1,\dots,n\}$ is used if and only if $y_j = 1$, leading to a cost of $c_j b_j + f_j$. Then, if constraint (6) does not hold with equality, the excess offer $\sum_{j=1}^{n} b_j y_j - D$ is left on the partial node $p \in \{1,\dots,n\}$ such that $z_p = 1$, which leads to a reimbursement of c_p for each excess unit. Constraint (7) ensures that a positive number of shipments is made from the partial node. If (6) holds with equality, the binary variable z_j can be set to 1 on any of the nodes used in the solution without changing the quantity ordered from this supplier or the objective function value, for which the second term is then zero.

The following propositions show that the resolution of the original SSFCTP can always be achieved through P2.

Proposition 2.5. *Let $(\boldsymbol{y}^2, \boldsymbol{z}^2)$ be a feasible solution to P2. A feasible solution to P1, with the same objective value, is given by $(\boldsymbol{x}^1, \boldsymbol{y}^1)$, where $y_j^1 = y_j^2$ and $x_j^1 = b_j y_j^2 - (\sum_{i=1}^{n} b_i y_i^2 - D) z_j^2$ for all $j \in \{1,\dots,n\}$.*

Proposition 2.6. *The optimal objective value is the same for P1 and P2, i.e. $Z = Z^1 = Z^2$.*

In addition, it is possible to strengthen the formulation of P2 to reduce the search space while preserving the validity of the preceding propositions.

Definition 2.7. *The strict total order $\prec$ on the set of nodes $\{1,\dots,n\}$ is defined as follows. For two nodes indexed by i and j: $i \prec j \iff \big((c_i < c_j)$ or $(c_i = c_j$ and $b_i < b_j)$ or $(c_i = c_j$ and $b_i = b_j$ and $i < j)\big)$. We denote by $\succ$ its inverse relation.*

Table 1: Lower and upper bounds by step of the heuristic phase

		LB on Z	UB on Z	LB on $Z\|(p$ is partial)	LB on $x_p\|(p$ is partial)	UB on $x_p\|(p$ is partial)	LB on the partial node's variable cost
H1)	Elementary Bounds	$Z^{\text{LP}}_{\text{LB}}$	Z^{G}_{UB}	-	-	-	-
H2)	Knapsack Transformation	-	$Z^{\text{P3}}_{\text{UB}}$	$Z^{\text{P3}}_{\text{LB}(p)}$	$LB^{\text{P3}}_{x_p}$	$UB^{\text{P3}}_{x_p}$	$C^{\text{P3}}_{\min}$
H3)	Dominance Relation	-	-	-	$LB^{\text{Dom}}_{x_p}$	-	-
H4)	Strong Linear Relaxation	-	-	$Z^{\text{LP}}_{\text{LB}(p)}$	$LB^{\text{LP}}_{x_p}$	$UB^{\text{LP}}_{x_p}$	-

Proposition 2.8. *There exists an optimal solution to P2 for which $z_j = 1$ if and only if $y_j = 1$ and $j \succ i$ for all $i \neq j$ such that $y_i = 1$. In other words, replacing constraints (8) and (9) by the following non-linear constraint:*

$$z_j = y_j \mathbb{1}(j \succ i \ \ \forall i \neq j : y_i = 1) \qquad \text{for } j = 1, \ldots, n \tag{11}$$

would not change the optimal value of P2 or invalidate Propositions 2.5 or 2.6.

Neither the initial formulation of P2 nor its transformation that includes constraint (11) are intended to be solved directly. However, given a vector $\boldsymbol{y}$ that specifies the set of suppliers to be used in a solution, Proposition 2.8 offers a direct rule that can be applied for the selection of a partial node. Using it, the SSFCTP reduces to identifying the optimal subset of nodes which must provide a positive number of items. This idea will motivate the introduction of P3, a transformation of P2 which can be solved efficiently to identify such subsets.

3 Lower and Upper Bounds

This section presents the four main steps of the heuristic phase in their order of execution. First, the classical linear relaxation of the SSFCTP and a simple greedy upper bound are briefly presented in Section 3.1. The rest of the section consists of new contributions. Section 3.2 develops model P3, a KP derived from a relaxation of P2. The resolution of this reformulation provides several bounds and represents the central part of KTA. Section 3.3 presents a new dominance relation between partial nodes, while Section 3.4 introduces a strong linear relaxation method which can be applied after fixing a supplier p as the partial node. These last two methods use the theory developed in Section 3.2 to eliminate as many potential partial nodes as possible.

Different types of bounds will be extracted from each step. More specifically, bounds on the objective value, on the conditional objective value and on the usage x_p given that node p is used as the partial node and on the partial node's variable cost will be obtained. To facilitate the understanding of the heuristic phase and the relations between its steps, Table 1 summarizes these bounds and introduces their notation.

3.1 Elementary Bounds

We start by considering a pair of simple lower and upper bounds which constitutes the first step of KTA. Although generally weak, these bounds can in some cases reduce the number of steps required by the iterative process of the subsequent sections.

3.1.1 Classical Linear Relaxation

A classical lower bound on the objective value of P1 is obtained by solving its linear relaxation.

$$Z_{\mathrm{LB}}^{\mathrm{LP}} = \min \sum_{j=1}^{n} e_j x_j \tag{12}$$

$$\text{s.t.} \sum_{j=1}^{n} x_j = D, \tag{13}$$

$$0 \leq x_j \leq b_j \quad \text{for } j = 1, \ldots, n, \tag{14}$$

where $e_j = c_j + f_j/b_j$.

Given that the nodes are sorted in non-increasing order of linearized cost $e_1 \leq e_2 \leq \cdots \leq e_n$, the optimal solution to the LP relaxation can be calculated in $\mathcal{O}(n)$ and is given by

$$x_j = \begin{cases} b_j & \text{for } j = 1, \ldots, s-1, \\ D - \sum_{j=1}^{s-1} b_j & \text{for } j = s, \\ 0 & \text{for } j = s+1, \ldots, n, \end{cases}$$

where the split supplier $s \in \{1, \ldots, n\}$ is such that

$$\sum_{j=1}^{s-1} b_j < D \qquad \text{and} \qquad \sum_{j=1}^{s} b_j \geq D. \tag{15}$$

The optimality of this solution can be demonstrated by considering the dual of the linear program (12), as it is done by Klose [1]. Throughout this article, it will be assumed that $e_1 \leq e_2 \leq \cdots \leq e_n$. This ordering will be required by various steps of our algorithm. We can note that, even if the nodes were not initially sorted, computing $Z_{\mathrm{LB}}^{\mathrm{LP}}$ could still be done in $\mathcal{O}(n)$ using the algorithm from Balas and Zemel [16] to determine the split node in linear time.

3.1.2 Greedy Upper Bound

Görtz and Klose [17] analyzed a range of popular greedy algorithms for the SSFCTP and showed that the upper bounds they offer can be arbitrarily bad. Nevertheless, it can be useful to derive a simple upper bound from the optimal solution to (12)-(14) in $\mathcal{O}(n)$. Our slightly modified version of the greedy algorithm for the SSFCTP uses Definition 2.7 to improve this classical method from the literature. Let $\boldsymbol{x}^*$ be the optimal solution to the LP relaxation and p the maximal element of the strictly totally ordered set $(Y^{\mathrm{LP}}, \prec)$, where $Y^{\mathrm{LP}} = \{1, 2, \ldots, s\}$ is the set of nodes that are used in the linear relaxation. Also, let us denote by $k = \min\{\sum_{i=1}^{s} b_i - D, b_p\}$ the

number of additional items that can be shipped from node s instead of node p without violating constraint (3). A feasible solution to P1 is given by $(\boldsymbol{x}, \boldsymbol{y})$, where

$$y_j = \begin{cases} 1 & \text{for } j = 1, \dots, s, \\ 0 & \text{for } j = s+1, \dots, n, \end{cases}$$

$$x_j = \begin{cases} x_j^* - k \cdot \mathbb{1}(j = p) & \text{for } j = 1, \dots, s-1, \\ x_j^* + k \cdot \mathbb{1}(j \neq p) & \text{for } j = s, \\ 0 & \text{for } j = s+1, \dots, n. \end{cases}$$

The objective value of this solution is $Z_{\text{UB}}^{\text{G}} = Z_{\text{LB}}^{\text{LP}} + f_s - (e_s - c_s)x_s^* - k(c_p - c_s)$. Increasing the number of units used on the split node by k to reduce the usage of supplier p reduces the objective value by $k(c_p - c_s)$ compared to the classical greedy upper bound, in which nodes $j = 1, \dots, s-1$ are completely used and node s provides the remaining demand.

3.2 Knapsack Transformation

We now propose a transformation of P2, noted P3, in which constraint (7) is relaxed and the decision variables z_j, $j \in \{1, \dots, n\}$ are removed. Furthermore, the reimbursement rate $\sum_{j=1}^{n} c_j z_j$ per excess unit is replaced by a multiplier λ, which can be interpreted as an approximation of the unit cost of the optimal solution's partial node, if it exists. This transformation of P2 exactly corresponds to a min-knapsack problem (Min-KP) with weight b_j and cost $(c_j - \lambda)b_j + f_j$ on item $j \in \{1, \dots, n\}$ and demand D.

$$[\text{P3}(\lambda)] \quad Z^3(\lambda) = \min \sum_{j=1}^{n} (c_j b_j + f_j) y_j - \Big(\sum_{j=1}^{n} b_j y_j - D \Big) \lambda \tag{16}$$

$$= \lambda D + \min \sum_{j=1}^{n} \big((c_j - \lambda) b_j + f_j \big) y_j \tag{17}$$

$$\text{s.t.} \sum_{j=1}^{n} b_j y_j \geq D, \tag{18}$$

$$y_j \in \{0, 1\} \qquad \text{for } j = 1, \dots, n. \tag{19}$$

It is well known that a Min-KP can be transformed into a standard KP in linear time [18]. The previous problem is equivalent to

$$Z^3(\lambda) = \lambda D + \sum_{j=1}^{n} \big((c_j - \lambda) b_j + f_j \big) - \max \sum_{j=1}^{n} \big((c_j - \lambda) b_j + f_j \big) \bar{y}_j \tag{20}$$

$$\text{s.t.} \sum_{j=1}^{n} b_j \bar{y}_j \leq \sum_{j=1}^{n} b_j - D, \tag{21}$$

$$\bar{y}_j \in \{0, 1\} \qquad \text{for } j = 1, \dots, n. \tag{22}$$

where $y_j = 1 - \bar{y}_j$ for $j = 1, \dots, n$.

Since the binary program P3 does not include explicit decision variables that would allow selecting a partial node, extracting feasible solutions to the SSFCTP from solutions to P3 will require a systematic procedure to nominate a partial node. The following definition serves this purpose.

Definition 3.1. *Let $\boldsymbol{y}^3$ be a feasible solution to P3. The critical node p^3 is defined as the maximal element of the strictly totally ordered set $(Y^3, \prec)$, where $Y^3 = \left\{j \in \{1, 2, \ldots, n\} : y_j^3 = 1\right\}$ is the set of nodes that are used in the solution.*

Different sets of solutions to each of the previous formulations are now introduced. We note:

- $S^1 \subset \mathbb{N}^n \times \{0,1\}^n$ the set of feasible solutions to P1.
- $\tilde{S}^1 \subseteq S^1$ the subset of feasible solutions $(\boldsymbol{x}^1, \boldsymbol{y}^1)$ to P1 that include at most one partial node, selected as the maximal element of the strictly totally ordered set $(Y^1, \prec)$, where $Y^1 = \left\{j \in \{1, \ldots, n\} : y_j^1 = 1\right\}$ and in which no unnecessary fixed cost is paid (i.e. $y_j^1 \leq x_j^1$ $\forall j$).
- $S^2 \subset \{0,1\}^n \times \{0,1\}^n$ the set of feasible solutions to P2.
- $\tilde{S}^2 \subseteq S^2$ the subset of feasible solutions $(\boldsymbol{y}^2, \boldsymbol{z}^2)$ to P2 that respect constraint (11).
- $S^3 \subset \{0,1\}^n$ the set of feasible solutions to P3.
- $\tilde{S}^3 \subseteq S^3$ the subset of feasible solutions $\boldsymbol{y}^3$ to P3 such that $\sum_{j=1}^n b_j y_j^3 - D \leq b_{p^3} - 1$, where p^3 respects Definition 3.1.

Bijective functions between these sets are now considered. These functions allow finding feasible solutions to P1, the classical formulation of the SSFCTP, using feasible solutions to P2 and P3, which are defined on smaller decision spaces. The bijection proofs are given in Appendix 1.

- $\tilde{f}_{2,1} : \tilde{S}^2 \to \tilde{S}^1$, $\tilde{f}_{2,1}(\boldsymbol{y}^2, \boldsymbol{z}^2) = (\boldsymbol{x}^1, \boldsymbol{y}^1)$, where $y_j^1 = y_j^2$ and $x_j^1 = b_j y_j^2 - (\sum_{i=1}^n b_i y_i^2 - D) z_j^2$ for all $j \in \{1, \ldots, n\}$.
- $\tilde{f}_{3,2} : \tilde{S}^3 \to \tilde{S}^2$, $\tilde{f}_{3,2}(\boldsymbol{y}^3) = (\boldsymbol{y}^2, \boldsymbol{z}^2)$, where $y_j^2 = y_j^3$ and $z_j^2 = \mathbb{1}(j = p^3)$ for all $j \in \{1, \ldots, n\}$.
- $\tilde{f}_{3,1} : \tilde{S}^3 \to \tilde{S}^1$, $\tilde{f}_{3,1}(\boldsymbol{y}^3) = \tilde{f}_{2,1}\big(\tilde{f}_{3,2}(\boldsymbol{y}^3)\big)$.

For $(\boldsymbol{x}^1, \boldsymbol{y}^1)$ a solution to P1, $(\boldsymbol{y}^2, \boldsymbol{z}^2)$ a solution to P2 and $\boldsymbol{y}^3$ a solution to P3(λ), their corresponding objective value will respectively be noted $Z^1(\boldsymbol{x}^1, \boldsymbol{y}^1)$, $Z^2(\boldsymbol{y}^2, \boldsymbol{z}^2)$ and $Z^3(\lambda, \boldsymbol{y}^3)$.

The following proposition is at the core of KTA. It allows us to restrict our search for an optimal solution to P1 to solutions in the subset $\tilde{S}^1$, which can be directly discovered from solutions to P3 through $\tilde{f}_{3,1}$.

Proposition 3.2. *$\tilde{S}^1$ contains at least one optimal solution to P1.*

Function $\tilde{f}_{3,1}$ only gives a solution to P1 when a solution to P3 that is an element of $\tilde{S}^3$ is considered. Consequently, the following proposition gives a sufficient condition on the value of the multiplier λ so that the resolution of P3(λ) leads to a feasible solution to P1.

Proposition 3.3. *For any multiplier value $\lambda < e_1$, any optimal solution to P3(λ) is an element of $\tilde{S}^3$.*

When a solution $\boldsymbol{y}^3 \in \tilde{S}^3$ is obtained, the bijections defined above can be used to identify the corresponding feasible solutions to P2 and P1. The relation between the objective value of these solutions can be stated as follows

Proposition 3.4. *Let $\boldsymbol{y}^3 \in \tilde{S}^3$ be a solution to P3. Then,*

$$Z^1\big(\tilde{f}_{3,1}(\boldsymbol{y}^3)\big) = Z^2\big(\tilde{f}_{3,2}(\boldsymbol{y}^3)\big) = Z^3(\lambda, \boldsymbol{y}^3) + (\lambda - c_{p^3})\Big(\sum_{j=1}^{n} b_j y_j^3 - D\Big).$$

The next propositions state essential properties of P3(λ) that will be widely exploited throughout the algorithm.

Proposition 3.5. *$Z^3(\lambda)$ is a lower bound on the objective value of any solution $(\boldsymbol{y}^2, \boldsymbol{z}^2)$ to P2 for which $c_j z_j^2 \leq \lambda$ for all $j \in \{1, \ldots, n\}$. Equivalently, $Z^3(\lambda)$ is a lower bound on the objective value of any solution $(\boldsymbol{x}^1, \boldsymbol{y}^1) \in \tilde{S}^1$ which does not contain a partial node or contains a partial node with a variable cost inferior or equal to λ.*

Proposition 3.6. *Let $\lambda_1 \neq \lambda_2$ be two positive values and suppose that an optimal solution $\boldsymbol{y}^3$ to P3(λ_1) is known, with objective value $Z^3(\lambda_1, \boldsymbol{y}^3) = Z^3(\lambda_1)$. Then, $Z^3(\lambda_1) - (\sum_{j=1}^{n} b_j y_j^3 - D)(\lambda_2 - \lambda_1)$ is an upper bound on $Z^3(\lambda_2)$.*

Proposition 3.7. *$Z^3(\lambda)$ is a concave function.*

In addition to a near-optimal feasible solution to the SSFCTP, the previous propositions lead to various bounds that will be presented in the following sections. These bounds help at reducing the computation time of the exact phase of KTA. They use a set L of multipliers λ for which P3(λ) has been solved. Adding elements to L will generally improve the bounds, but solving too many KPs is computationally cumbersome. Finding an appropriate set of multipliers is therefore an important and non-trivial problem.

The closeness of λ to the unit cost of the partial node of the optimal solution will help at finding a good incumbent while having a diverse set of multipliers will better restrict the set of potential partial nodes. Based on these ideas, we introduce the heuristic `P3Search`, which produces a set L of small cardinality allowing the bounds of Sections 3.2.1 to 3.2.4 to be relatively tight. The algorithm starts by solving P3(λ) for $\lambda = \min\big\{e_s,\ \max_{j\in\{1,\ldots,n\}}\{c_j\}\big\}$. In the subsequent steps, λ is set to the highest unit cost among the nodes selected in the previous solution. This process is repeated until a same multiplier is visited twice, λ exceeds the linearized cost of the split node or a solution without a partial node is found. Additional comments are given in Section 3.2.5 on the choice of the initial multiplier value.

Algorithm 1 `P3Search`

$L \leftarrow \emptyset$
$\lambda \leftarrow \min\left\{e_s,\ \max_{j\in\{1,\dots,n\}}\{c_j\}\right\}$
do
  $\boldsymbol{y}^\lambda \leftarrow \arg\min_{\boldsymbol{y}\in S^3}\left\{Z^3(\lambda,\boldsymbol{y})\right\}$
  $L \leftarrow L \cup \{\lambda\}$
  $\lambda \leftarrow \max_{j\in\{1,\dots,n\}}\{y_j^\lambda c_j\}$
while ($\lambda \notin L$ and $\lambda \leq e_s$ and $\sum_{j=1}^n b_j y_j^\lambda > D$)

An example of the execution of `P3Search` is provided in Appendix 3.

3.2.1 Upper bound $Z_{\text{UB}}^{\text{P3}}$

First, we present the upper bound $Z_{\text{UB}}^{\text{P3}}$ on the objective value of the SSFCTP that can be extracted from the information retrieved through this algorithm.

For a given multiplier value $\lambda \in L$, let $\boldsymbol{y}^\lambda$ be the optimal solution to P3(λ) identified during the execution of `P3Search` and let p^λ be its critical node as given by Definition 3.1. Let us denote by $M = \{\lambda \in L : \boldsymbol{y}^\lambda \in \tilde{S}^3\}$ the set of multipliers such that $\tilde{f}_{3,2}(\boldsymbol{y}^\lambda)$ is a feasible solution to P2. Proposition 3.4 thus directly gives an upper bound on the SSFCTP's objective value. For the other multipliers $\lambda \in L\backslash M$, an upper bound is given by $Z^3(\lambda)+\lambda(\sum_{j=1}^n b_j y_j^\lambda - D)-(c_{p^\lambda}b_{p^\lambda}+f_{p^\lambda})$ and corresponds to the objective value that is obtained when using all the nodes $j \neq p^\lambda$ such that $y_j^\lambda = 1$. Since $\sum_{j=1}^n b_j y_j^\lambda - b_{p^\lambda} \geq D$ by definition of M, the total offer associated with this solution is greater or equal to D and its value is therefore an upper bound on Z since all the costs c_j and f_j are assumed to be positive. In summary, for each value $\lambda \in L$,

$$Z_{\text{UB}}^{\text{P3}(\lambda)} = \begin{cases} Z^3(\lambda) + (\lambda - c_{p^\lambda})(\sum_{j=1}^n b_j y_j^\lambda - D) & \text{if } \sum_{j=1}^n b_j y_j^\lambda - D \leq b_{p^\lambda} - 1, \\ Z^3(\lambda) + \lambda(\sum_{j=1}^n b_j y_j^\lambda - D) - (c_{p^\lambda} b_{p^\lambda} + f_{p^\lambda}) & \text{otherwise.} \end{cases}$$

The upper bound given by P3 is noted $Z_{\text{UB}}^{\text{P3}} = \min_{\lambda\in L}\{Z_{\text{UB}}^{\text{P3}(\lambda)}\}$.

3.2.2 Lower bounds $C_{\min}^{\text{P3}}$ on the partial node's variable cost

Second, we introduce two lower bounds $C_{\min}^{\text{P3}_1}$ and $C_{\min}^{\text{P3}_2}$ on the variable cost of the partial node of any solution from $\tilde{S}^1$ that is susceptible to improve the incumbent value $Z_{\text{UB}}^{\text{P3}}$. By excluding potential partial nodes, these bounds contribute to reduce the number of knapsack subproblems that will have to be solved during the exact phase of the algorithm.

If $Z_{\text{UB}}^{\text{P3}} = Z_{\text{UB}}^{\text{P3}(\lambda^*)}$ for a multiplier value $\lambda^* \in M$ such that $c_{p^{\lambda^*}} = \lambda^*$, then $Z_{\text{UB}}^{\text{P3}} = Z^3(\lambda^*)$. In this case, Proposition 3.5 implies that the objective value of a solution from $\tilde{S}^1$ that includes a partial node p with a variable cost $c_p \leq \lambda^*$ cannot be less than $Z_{\text{UB}}^{\text{P3}}$. This leads to the lower bound $C_{\min}^{\text{P3}_1} = \lambda^*$.

If such a multiplier $\lambda^* \in M$ does not exist, we set $C_{\min}^{\text{P3}_1} = 0$. In this case, a slightly weaker bound $C_{\min}^{\text{P3}_2}$ can still be calculated. Let us represent the points $\left(\lambda, Z^3(\lambda)\right)$ such that $\lambda \in L$

in a graph with λ on the $x-$axis and $Z^3(\lambda)$ on the $y-$axis. We note $\lambda_0 = \max\{\lambda \in L : Z^3(\lambda) > Z_{\text{UB}}\}$ and $\lambda_1 = \min\{\lambda \in L : Z^3(\lambda) < Z_{\text{UB}}\}$, where Z_{UB} is the incumbent objective value. The equation of the line passing through points $\big(\lambda_0, Z^3(\lambda_0)\big)$ and $\big(\lambda_1, Z^3(\lambda_1)\big)$ is given by $g(\alpha) = Z^3(\lambda_0) + (\alpha - \lambda_0)\frac{Z^3(\lambda_1) - Z^3(\lambda_0)}{\lambda_1 - \lambda_0}$. Consequently, for $\alpha^* = \lambda_0 + (\lambda_1 - \lambda_0)\frac{Z^3(\lambda_0) - Z_{\text{UB}}}{Z^3(\lambda_0) - Z^3(\lambda_1)}$, $g(\alpha^*) = Z_{\text{UB}}$ and Proposition 3.7 therefore implies that $Z_{\text{UB}} \leq Z^3(\alpha^*)$. By Proposition 3.5, we conclude that Z_{UB} is a lower bound on the objective value of any solution to P1 whose partial node's variable cost c_p is inferior or equal to α^*. In other words, $C^{\text{P3}_1}_{\min} = \alpha^*$ is a valid lower bound on the partial node's variable cost. If both sets $\max\{\lambda \in L : Z^3(\lambda) > Z_{\text{UB}}\}$ and $\min\{\lambda \in L : Z^3(\lambda) < Z_{\text{UB}}\}$ are empty, then $C^{\text{P3}_2}_{\min} = 0$.

In short, the best lower bound on the partial node's variable cost that can be extracted from the data collected in `P3Search` is

$$C^{\text{P3}}_{\min} = \begin{cases} C^{\text{P3}_1}_{\min} & \text{if } C^{\text{P3}_1}_{\min} > 0, \\ C^{\text{P3}_2}_{\min} & \text{otherwise.} \end{cases}$$

3.2.3 Lower bound $Z^{\text{P3}}_{\text{LB}(p)}$ given that a node p is partial

Third, we detail a lower bound on the objective value of any solution $(\boldsymbol{x}^1, \boldsymbol{y}^1) \in \tilde{S}^1$ that uses a specific supplier p as a partial node.

This bound follows from Propositions 3.5 and 3.7. If $c_p \in L$, it is directly given by $Z^3(c_p)$. Otherwise, for $\lambda_0 = \max\{\lambda \in L : \lambda < c_p\}$ and $\lambda_1 = \min\{\lambda \in L : \lambda > c_p\}$, a lower bound on $(Z|p$ is partial) is given by $Z^3(\lambda_0) - (c_p - \lambda_0)\frac{Z^3(\lambda_0) - Z^3(\lambda_1)}{\lambda_1 - \lambda_0}$. This bound corresponds to the $y-$axis value of the point of $x-$axis value c_p for the line passing by points $\big(\lambda_0, Z^3(\lambda_0)\big)$ and $\big(\lambda_1, Z^3(\lambda_1)\big)$. We will denote this lower bound by $Z^{\text{P3}}_{\text{LB}(p)}$ for each potential partial node p.

3.2.4 Lower and upper bounds $LB^{\text{P3}}_{x_p}$ and $UB^{\text{P3}}_{x_p}$ on x_p given that p is partial

Finally, the last bounds that are directly extracted from the pairs $\big(\lambda, Z^3(\lambda)\big)$, $\lambda \in L$ concern the quantity ordered from supplier p when it is used as a partial node.

Let $\boldsymbol{y}^3 \in \tilde{S}^3$ be a feasible solution to P3(λ) with a corresponding solution $\tilde{f}_{3,1}(\boldsymbol{y}^3) = (\boldsymbol{x}^1, \boldsymbol{y}^1) \in \tilde{S}^1$ for which $p^3 = p$, with $c_p \geq C^{\text{P3}}_{\min}$. For $(\boldsymbol{x}^1, \boldsymbol{y}^1)$ to improve the incumbent objective value Z_{UB}, one must have

$$Z^1(\boldsymbol{x}^1, \boldsymbol{y}^1) \overset{(3.4)}{=} Z^3(\lambda, \boldsymbol{y}^3) + \Big(\sum_{j=1}^{n} b_j y^3_j - D\Big)(\lambda - c_p) < Z_{\text{UB}}. \tag{23}$$

Also, for $\boldsymbol{y}^3$ not to have been discovered in `P3Search`, the inequality

$$Z^3(\lambda, \boldsymbol{y}^3) \geq Z^3(\lambda) \tag{24}$$

must hold $\forall\ \lambda \in L$. The objective value $Z^3(\lambda, \boldsymbol{y}^3)$, as a function of λ, corresponds to the line of equation $\sum_{j=1}^{n}(b_j c_j + f_j)y^3_j - (\sum_{j=1}^{n} b_j y^3_j - D)\lambda$. Since the absolute value of its slope is

given by the excess supply $\sum_{j=1}^{n} b_j y_j^3 - D$ associated with solution $\boldsymbol{y}^3$, the previous constraints offer both an upper bound and a lower bound on this amount for a solution with partial node's variable cost c_p. By equations (23) and (24), we conclude that the inequality

$$\Big(\sum_{j=1}^{n} b_j y_j^1 - D\Big)(c_p - \lambda) + Z_{\text{UB}} > Z^3(\lambda),\ \forall \lambda \in L \tag{25}$$

must hold for $(\boldsymbol{x}^1, \boldsymbol{y}^1)$ to improve the incumbent solution to P1. This is equivalent to the following conditions.

$$\sum_{j=1}^{n} b_j y_j^1 - D > \frac{Z^3(\lambda) - Z_{\text{UB}}}{c_p - \lambda} \qquad \forall \lambda \in L : \lambda < c_p, \tag{26}$$

$$\sum_{j=1}^{n} b_j y_j^1 - D < \frac{Z^3(\lambda) - Z_{\text{UB}}}{c_p - \lambda} \qquad \forall \lambda \in L : \lambda > c_p. \tag{27}$$

Since solutions from $\tilde{S}^1$ include at most one partial node, the number of shipments to be made from p is directly given by $x_p = b_p - (\sum_{j=1}^{n} b_j y_j^1 - D)$. Also, the excess supply is always an integer. This leads to the bounds

$$\begin{cases} LB_{x_p}^{\text{P3}} = b_p - \min_{\{\lambda \in L : \lambda > c_p\}} \left\{ \left\lceil \frac{Z^3(\lambda) - Z_{\text{UB}}}{c_p - \lambda} - 1 \right\rceil \right\}, \\ \\ UB_{x_p}^{\text{P3}} = b_p - \max_{\{\lambda \in L : \lambda < c_p\}} \left\{ \left\lfloor \frac{Z^3(\lambda) - Z_{\text{UB}}}{c_p - \lambda} + 1 \right\rfloor \right\}. \end{cases}$$

When considering solutions where node p is partial, we can therefore narrow down our search to solutions in which $LB_{x_p}^{\text{P3}} \leq x_p \leq UB_{x_p}^{\text{P3}}$.

3.2.5 Choice of initial value of λ in `P3Search`

Our computational experiments have shown that taking $\lambda = \max_{j \in \{1,2,\ldots,s\}}\{c_j\}$ as the initial multiplier value generally leads to identifying the best reachable upper bound $Z_{\text{UB}}^{\text{P3}}$ in a minimal number of iterations. This follows from the fact that this value is generally a good approximation of the maximum unit cost among the nodes used in an optimal solution to P1.

We nevertheless propose to use $\lambda = \min\left\{e_s,\ \max_{j \in \{1,\ldots,n\}}\{c_j\}\right\}$ instead. It is frequent for the equality $\sum_{j=1}^{n} b_j y_j^{\lambda} = D$ to hold despite this higher initial value of λ which favours solutions with excess offer. In this case, if $\max_{j \in \{1,\ldots,n\}}\{c_j\} \leq e_s$, then $C_{\min}^{\text{P3}_1} = \max_{j \in \{1,\ldots,n\}}\{c_j\}$. This means that any solution including a partial node can be ignored and $\tilde{f}_{3,1}(\boldsymbol{y}^{\lambda})$ is an optimal solution to the problem. These cases are frequent enough and the execution time reduction associated with this early proof of optimality sufficiently large to justify this choice of initial multiplier.

3.3 Dominance Relation

Some of the most expensive calculations of KTA are performed given that a node p is fixed as partial and must be repeated for each node that may be partial in an optimal solution.

Consequently, the efficiency of the algorithm relies in good part on our ability to exclude a significant proportion of candidates before performing these calculations. For the remaining candidates, restricting their partial use x_p to a narrow interval $[LB_{x_p}, UB_{x_p}]$ can still reduce the computing time. The following dominance relation, by seeking to improve the bounds of Section 3.2.4, serves both these purposes.

Definition 3.8. *If q is a node such that $q \succ p$ and $f_q \leq f_p$, then supplier p is said to be dominated by q as a partial node for any usage $x_p \leq LB_{x_p}^{Dom}(q) = \begin{cases} b_q & \text{if } c_q = c_p, \\ \min\left\{\frac{f_p - f_q}{c_q - c_p}, b_q\right\} & \text{otherwise.} \end{cases}$*

Indeed, for any solution $(x^1, y^1) \in \tilde{S}^1$ in which $x_p \leq LB_{x_p}^{\text{Dom}}(q)$ units are shipped from the partial node p, one can replace p by the unused node q to obtain a feasible solution with an objective value of at most $Z^1(x^1, y^1)$. This leads to the following lower bound on x_p given that p is partial.

Proposition 3.9. *Let $LB_{x_p}^{Dom}$ be defined as follows.*

$$LB_{x_p}^{Dom} = \max_{\left\{q \in \{1,\ldots,n\} : q \succ p \text{ and } f_q \leq f_p\right\}} \left\{\lfloor LB_{x_p}^{Dom}(q) + 1 \rfloor\right\} \tag{28}$$

There exists an optimal solution $(x^1, y^1) \in \tilde{S}^1$ to P1 in which p is not partial or $x_p \geq LB_{x_p}^{Dom}$.

3.4 Strong Linear Relaxation

By taking advantage of the theory developed in Section 3.2, the classical linear relaxation of the SSFCTP can be adapted to pursue the same objective as the previous section. We propose a stronger linear relaxation that is computationally more demanding than the dominance relation, but leads to a better lower bound on x_p given that a node p is partial. It also offers an upper bound on this same variable as well as a a lower bound $Z_{\text{LB}(p)}^{\text{LP}}$ which aims to improve the one described in Section 3.2.3.

3.4.1 Lower bound $Z_{\mathbf{LB}(p)}^{\mathbf{LP}}$ given that a node p is partial

By considering only the solutions to P1 that are elements of $\tilde{S}^1$ and respect the bounds LB_{x_p} and UB_{x_p} computed previously, a second lower bound on the objective value of (P1$|p$ is partial) can be obtained. Forcing all the nodes j such that $j \succ p$ to be unused and adding constraints $LB_{x_p} \leq x_p \leq UB_{x_p}$ to the linear relaxation (12)-(14) leads to the following problem.

$$Z_{\text{LB}(p)}^{\text{LP}} = \min\ f_p + c_p x_p + \sum_{\substack{j=1 \\ j \neq p}}^{n} e_j x_j \tag{29}$$

$$\text{s.t.} \sum_{j=1}^{n} x_j = D, \tag{30}$$

$$LB_{x_p} \leq x_p \leq UB_{x_p}, \tag{31}$$

$$0 \leq x_j \leq b_j \mathbb{1}(j \prec p) \text{ for } j = 1, \ldots, p-1, p+1, \ldots, n. \tag{32}$$

If $UB_{x_p} + \sum_{\substack{j=1 \\ j \prec p}}^{n} b_j < D$, then no feasible solution $(\boldsymbol{x}^1, \boldsymbol{y}^1) \in \tilde{S}^1$ containing p as a partial node and respecting constraint (31) exists. In this case, $Z^{\mathrm{LP}}_{\mathrm{LB}(p)} = +\infty$. Otherwise, let $r^p, t^p \in \{1, \ldots, n\}$ be the suppliers such that

$$\sum_{\substack{j=1 \\ j \prec p}}^{r^p - 1} b_j < D - UB_{x_p} \leq \sum_{\substack{j=1 \\ j \prec p}}^{r^p} b_j, \tag{33}$$

$$\sum_{\substack{j=1 \\ j \prec p}}^{t^p - 1} b_j < D - LB_{x_p} \leq \sum_{\substack{j=1 \\ j \prec p}}^{t^p} b_j. \tag{34}$$

All the nodes $j < r^p$ such $j \prec p$ must be completely used, while all the nodes $j > t^p$ different from p must be unused in the optimal solution to respect the demand constraint. Finally, let s^p be the last node different from p to be used in the optimal solution to the strong linear relaxation.

$$s^p = \begin{cases} r^p & \text{if } e_{r^p} \geq c_p, \\ \max\left\{ j \in \{r^p, \ldots, t^p\} : e_j < c_p \text{ and } j \prec p \right\} & \text{otherwise.} \end{cases} \tag{35}$$

The optimal solution to model (29)-(32) is given by

$$x_j = \begin{cases} b_j & \text{for } \ j \in \{1, \ldots, s^p - 1\} : j \prec p, \\ D - UB_{x_p} - E_p & \text{for } j = s^p \text{ if } e_j \geq c_p, \\ \min\{b_j,\ D - LB_{x_p} - E_p\} & \text{for } j = s^p \text{ if } e_j < c_p, \\ D - E_p - x_{s^p} & \text{for } j = p, \\ 0 & \text{for } \ j \in \{1, \ldots, p-1, p+1, \ldots, n\} : j \succ p \text{ or } j > s^p, \end{cases}$$

where $E_p = \sum_{\substack{j=1 \\ j \prec p}}^{s^p - 1} b_j$ is the total offer on the nodes respecting constraint (32) and preceding the split node s^p.

To obtain this solution, LB_{x_p} units are first used on node p. Afterwards, units are taken on the nodes having the lowest linearized cost e_j and respecting constraint (32) until the remaining offer $UB_{x_p} - LB_{x_p}$ available on p suffices to fill the demand. This step corresponds to using nodes $j \leq r^p - 1$ such that $j \prec p$ completely. If $e_{r^p} \geq c_p$, then $s^p = r^p$ by (35). In this case, the maximal offer UB_{x_p} is used on supplier p and the remaining demand of $D - UB_{x_p} - E_p$ units is satisfied using node s^p. Otherwise, the following nodes that also respect constraint (32) are used until all the demand is satisfied or a node with linearized cost $e_j \geq c_p$ is reached. The remaining demand $D - LB_{x_p} - E_p - x_{s^p} \in \{0, 1, \ldots, UB_{x_p} - LB_{x_p}\}$ is then filled using more units from supplier p.

By construction, $Z^{\mathrm{LP}}_{\mathrm{LB}(p)}$ is a lower bound on the objective value of any solution $(\boldsymbol{x}^1, \boldsymbol{y}^1) \in \tilde{S}^1$ containing p as a partial node and respecting the previously calculated bounds LB_{x_p} and UB_{x_p} on its usage x_p. Since KTA only considers solutions to P1 that are elements of $\tilde{S}^1$, supplier p no longer have to be considered as a potential partial node if $Z^{\mathrm{LP}}_{\mathrm{LB}(p)} \geq Z_{\mathrm{UB}}$, where Z_{UB} is the incumbent objective value.

3.4.2 Lower and upper bounds $LB_{x_p}^{\mathrm{LP}}$ and $UB_{x_p}^{\mathrm{LP}}$ on x_p given that p is partial

When the lower bound $Z_{\mathrm{LB}(p)}^{\mathrm{LP}}$ does not allow excluding p as a potential partial node, a strong linear relaxation can be used to tighten bounds LB_{x_p} and UB_{x_p} in (31). These two bounds are defined as follows:

$$\begin{cases} LB_{x_p}^{\mathrm{LP}} = \min\{x : Z_{\mathrm{LB}(p)}^{\mathrm{LP}} < Z_{\mathrm{UB}} \text{ given that } x_p = x\}, \\ UB_{x_p}^{\mathrm{LP}} = \max\{x : Z_{\mathrm{LB}(p)}^{\mathrm{LP}} < Z_{\mathrm{UB}} \text{ given that } x_p = x\}. \end{cases}$$

They respectively correspond to the lowest and the highest usage x_p on the partial node p that can be enforced in a solution to the strong linear relaxation without the optimal value $Z_{\mathrm{LB}(p)}^{\mathrm{LP}}$ to reach the incumbent value Z_{UB}.

Suppose that the optimal solution $\boldsymbol{x}^*$ to model (29)-(32) has been obtained as previously explained with an optimal value $Z_{\mathrm{LB}(p)}^{\mathrm{LP}} < Z_{\mathrm{UB}}$.

If the optimal usage x_p^* on the partial node p is such that $x_p^* = LB_{x_p}$, then the current lower bound LB_{x_p} cannot be improved and $LB_{x_p}^{\mathrm{LP}}$ is set to LB_{x_p}. Otherwise, $LB_{x_p}^{\mathrm{LP}}$ can be obtained by taking the unused units $b_j - x_j^*$ on nodes $s^p, s^p + 1, \ldots$ respecting constraint (32) to reduce the usage x_p of the partial node. Each time a unit is taken from j instead of p, the objective value increases by $(e_j - c_p)$. This process must be repeated until $x_p = LB_{x_p}$, the cost of the linear relaxation reaches Z_{UB} or all the offer on nodes $j \prec p$ has been used. The lower bound $LB_{x_p}^{\mathrm{LP}}$ is given by the first value x_p for which one of the previous conditions is met.

Similarly, if $x_p^* = UB_{x_p}$, then the current upper bound UB_{x_p} cannot be improved and $UB_{x_p}^{\mathrm{LP}}$ is set to UB_{x_p}. Otherwise, $UB_{x_p}^{\mathrm{LP}}$ can be obtained by reducing the usage x_j^* on nodes $s^p, s^p - 1, \ldots$ while increasing the number of shipments x_p made from the partial node. Each time a unit is taken from j instead of p, the objective value increases by $(c_p - e_j)$. This process must be repeated until $x_p = UB_{x_p}$, the cost of the linear relaxation reaches Z_{UB} or no more units are used on nodes different from p. Once again, $UB_{x_p}^{\mathrm{LP}}$ corresponds to the usage x_p on the partial node when one of the previous conditions is first met.

4 Filtering

The various bounds described in the previous section generally allow restricting the set of nodes that may be partially used in a solution that improves the incumbent to a very small subset of suppliers $P \subseteq \{1, \ldots, n\}$. From there, solving P1 exactly will require to solve a knapsack subproblem for each of the remaining candidates $p \in P$. To reduce as much as possible the number of nodes to be considered in each of these subproblems, a filtering of the non-partial nodes is carried out beforehand.

4.1 Weak Linear Filtering

When considering the classical linear relaxation (12)-(14), as mentioned by Klose [1], if the flow x_j on a node $j < s$ is fixed to 0, the LP bound will increase by at least $b_j(e_s - e_j)$, since b_j units will have to be taken on suppliers $i \geq s$ instead of supplier j. Hence, if $Z_{\mathrm{LB}}^{\mathrm{LP}} + b_j(e_s - e_j) > Z_{\mathrm{UB}}$,

then node j cannot be unused in an optimal solution to P1. Consequently, when solving (P1$|p$ is partial) for a partial node $p \neq j$, the number x_j of units ordered from supplier j can be fixed to b_j by trichotomy, since p has already been fixed as the only partial node in the solution.

Similarly, if the flow x_j on a supplier $j > s$ is fixed to b_j, these b_j units will each be paid at a cost of e_j instead of a cost inferior or equal to e_s. The LP bound will therefore increase by at least $b_j(e_j - e_s)$. If $Z_{\text{LB}}^{\text{LP}} + b_j(e_j - e_s) > Z_{\text{UB}}$, then node j cannot be complete in an optimal solution to P1. In this case, when solving (P1$|p$ is partial) for a partial node $p \neq j$, the number of shipments x_j to be made from j can be fixed to 0.

4.2 Strict Total Order Filtering

Since we only consider solutions to P1 that are elements of $\tilde{S}^1$, when a partial node $p \in P$ is fixed in a given solution, the quantity ordered from nodes j such that $j \succ p$ must be zero, by definition of $\tilde{S}^1$. The solutions that can be obtained during the exact phase of the algorithm always contain a partial node. Hence, if we denote the maximal element of the strictly totally ordered set $(P, \prec)$ by q, then it follows that any node $j \succ q$ will have to be unused in the solution to each remaining subproblem.

Given the set P of potential partial nodes and its maximal element q, this reasoning leads to the following reformulation of the LP relaxation (12)-(14).

$$Z^{\text{LP}(q)} = \min \sum_{j=1}^{n} e_j x_j \tag{36}$$

$$\text{s.t.} \sum_{j=1}^{n} x_j = D, \tag{37}$$

$$0 \leq x_j \leq \big(1 - \mathbb{1}(j \succ q)\big) b_j \text{ for } j = 1, \ldots, n. \tag{38}$$

This problem corresponds to the linear program (12)-(14) solved for the subset of suppliers $Q = \big\{j \in \{1, \ldots, n\} : j \prec q \text{ or } j = q\big\}$. Consequently, its optimal solution $\boldsymbol{x}^*$ is obtained as explained in Section 3.1.1 considering only this subset. It is used directly to improve the weak linear filtering described in the previous section.

4.3 Strong Linear Filtering

Let $\boldsymbol{x}^*$ be the optimal solution to the LP relaxation (36)-(38) and $s^q \in Q$ its split node, defined as follows:

$$\sum_{\substack{i=1 \\ i \in Q}}^{s^q - 1} b_i < D \leq \sum_{\substack{i=1 \\ i \in Q}}^{s^q} b_i. \tag{39}$$

The bounds of Section 4.1 are obviously strengthened when all the nodes that are not part of set Q are forced to be unused. However, an even stronger bound can be calculated for each node $j \in Q$ when the remaining offer $b_{s^q} - x^*_{s^q}$ on the split node (or the number $x^*_{s^q}$ of units taken on it) is insufficient for b_j more units to be respectively taken or unused from s^q. When

x_j is fixed to 0 for a node $j < s^q$ such that $j \in Q$, the bound $Z^{\mathrm{LP}(q)}$ will increase by

$$k_j^0 = \begin{cases} b_j(e_{s^q} - e_j) & \text{if } b_j \leq b_{s^q} - x^*_{s^q}, \\ (b_{s^q} - x^*_{s^q})e_{s^q} + \sum\limits_{\substack{i=s^q+1 \\ i\in Q}}^{t^j-1} b_i e_i + \Big(b_j - (b_{s^q} - x^*_{s^q}) - \sum\limits_{\substack{i=s^q+1 \\ i\in Q}}^{t^j-1} b_i\Big)e_{t^j} - b_j e_j & \text{otherwise.} \end{cases}$$

The first case occurs when the remaining offer on the split node suffices to fill the demand of b_j units resulting from the removal of node j from the solution. In the second case, each node $i \in \{s^q, \ldots, t^j - 1 : i \in Q\}$ is completely used and the remaining demand of $\big(b_j - (b_{s^q} - x^*_{s^q}) - \sum_{\substack{i=s^q+1 \\ i\in Q}}^{t^j-1} b_i\big)$ units is filled using supplier $t^j \in \{s^q + 1, \ldots, n\} \cap Q$, which respects

$$\sum_{\substack{i=s^q+1 \\ i\in Q}}^{t^j-1} b_j < b_j - (b_{s^q} - x^*_{s^q}) \leq \sum_{\substack{i=s^q+1 \\ i\in Q}}^{t^j} b_j. \tag{40}$$

Supplier t^j corresponds to the split node of problem (36)-(38) when the offer b_j is set to 0 on node j. If $\sum_{\substack{i=1 \\ j\neq i\in Q}}^{n} b_i < D$, then no feasible solution $(\boldsymbol{x}^1, \boldsymbol{y}^1) \in \tilde{S}^1$ exists in which node j is unused. Therefore, k_j^0 can be fixed to $+\infty$ in this case. Otherwise, $Z^{\mathrm{LP}(q)} + k_j^0$ corresponds to the optimal value of the LP relaxation on subset Q when supplier j is removed from the problem.

For $j > s^q$, a similar bound can be obtained by reducing the number of shipments made from nodes $i \in \{s, s-1, \ldots 1 : i \in Q\}$ by b_j in total, so that b_j units can be shipped by supplier j. Hence, when x_j is fixed to b_j for a node $j > s^q$ such that $j \in Q$, the bound $Z^{\mathrm{LP}(q)}$ will increase by

$$k_j^1 = \begin{cases} b_j(e_j - e_{s^q}) & \text{if } b_j \leq x^*_{s^q}, \\ b_j e_j - x^*_{s^q} e_{s^q} - \sum\limits_{\substack{i=r^j+1 \\ i\in Q}}^{s^q-1} b_i e_i - \Big(b_j - x^*_{s^q} - \sum\limits_{\substack{i=r^j+1 \\ i\in Q}}^{s^q-1} b_i\Big)e_{r^j} & \text{otherwise,} \end{cases}$$

where supplier $r^j \in \{1, \ldots, s^q - 1\} \cap Q$ is such that

$$\sum_{\substack{i=r^j+1 \\ i\in Q}}^{s^q-1} b_j < b_j - x^*_{s^q} \leq \sum_{\substack{i=r^j \\ i\in Q}}^{s^q-1} b_j. \tag{41}$$

Supplier r^j is the split node of model (36)-(38) when the demand D is replaced by $D - b_j$ and $Z^{\mathrm{LP}(q)} + k_j^1$ corresponds to the optimal value of the LP relaxation on subset Q when supplier j is completely used.

Note that variants of k_j^0 and k_j^1 can be applied to the split node $j = s^q$ to respectively conclude, in some cases, that s^q cannot be unused or completely used in an optimal solution $(\boldsymbol{x}^1, \boldsymbol{y}^1) \in \tilde{S}^1$. These simple adjustments are not detailed here for concision concerns.

In summary, the filtering procedure we just introduced allows defining two subsets of nodes $R, T \subseteq Q$ defined as follows:

$$\begin{cases} R = \left\{ j \in \{1, \dots, s^q - 1\} : Z^{\mathrm{LP}(q)} + k_j^0 > Z_{\mathrm{UB}} \right\}, \\ T = \left\{ j \in \{s^p + 1, \dots, n\} : Z^{\mathrm{LP}(q)} + k_j^1 > Z_{\mathrm{UB}} \right\}. \end{cases}$$

In the following section, when solving a subproblem (P1$|p$ is partial) for any node $p \in P$, the problem size will be reduced by fixing

$$y_j = \begin{cases} 1 & \forall\ j \in R : j \neq p, \\ 0 & \forall\ j \in T : j \neq p, \end{cases}$$

$$x_j = \begin{cases} b_j & \forall\ j \in R : j \neq p, \\ 0 & \forall\ j \in T : j \neq p. \end{cases}$$

As will be seen in Section 7, this method filters the vast majority of suppliers for all the classes of problems we considered.

5 Exact Method

After solving P3(λ) for each multiplier $\lambda \in L$ as described in Section 3.2, if the incumbent value Z_{UB} is such that $Z_{\mathrm{UB}} \leq Z^3(\lambda)$ for at least one value $\lambda \geq 0$, then it follows from Proposition 3.5 that a solution to P1 must contain a partial node to improve the current upper bound. In this case, finding an optimal solution $(\boldsymbol{x}^1, \boldsymbol{y}^1) \in \tilde{S}^1$ to (P1$|p$ is partial) for each of the remaining candidates $p \in P$ suffices to conclude the resolution of the SSFCTP. The exact phase of KTA is therefore to solve each of these $|P|$ problems individually. This approach owes its efficiency to the extremely restricted cardinality of P at this point and to the fact that the filtering methods of Section 4 fix the vast majority of the decision variables for each subproblem.

5.1 Knapsack Transformation

Since we only consider solutions $(\boldsymbol{x}^1, \boldsymbol{y}^1) \in \tilde{S}^1$, each subproblem (P1$|p$ is partial) can be reduced to a KP. In the following formulation, which is very similar to model (20)-(22), the fixed cost f_p is initially paid and the maximal flow UB_{x_p} on the partial node is used. The total cost $c_j b_j + f_j$ is also paid on all the other nodes. Adding a node $j \neq p$ or a group of items from p in the knapsack can then be seen as removing them from our solution to the SSFCTP. The problem is therefore to maximize the total cost of the unused nodes and that of the unused units of the partial node while keeping enough active nodes so that the demand constraint of the original

problem is respected.

$$[\mathrm{P4}_p] \quad Z_p^4 = f_p + UB_{x_p}c_p + \sum_{\substack{j=1\\ j\neq p}}^{n}(c_j b_j + f_j) - \max\Big(\sum_{\substack{j=1\\ j\neq p}}^{n}(c_j b_j + f_j)\bar{y}_j + \sum_{j=n+1}^{n+d^p}(c_p\hat{b}_j)\bar{y}_j\Big) \tag{42}$$

$$\text{s.t.} \sum_{\substack{j=1\\ j\neq p}}^{n} b_j\bar{y}_j + \sum_{j=n+1}^{n+d^p}\hat{b}_j\bar{y}_j \leq \sum_{\substack{j=1\\ j\neq p}}^{n} b_j + UB_{x_p} - D, \tag{43}$$

$$\bar{y}_j \in \{0,1\} \qquad \text{for } j = 1,\ldots,n+d^p. \tag{44}$$

The capacity of this KP, which is given by the right-hand side of constraint (43), corresponds to the number of units that can be left unused on the nodes of the original problem given that p is partial and that at most UB_{x_p} can be used on it.

The integer $d^p = \lceil \log_2(UB_{x_p} - LB_{x_p} + 1)\rceil$ is such that $\sum_{j=1}^{d^p-1} 2^{j-1} < UB_{x_p} - LB_{x_p} \leq \sum_{j=1}^{d^p} 2^{j-1}$. Each item $j \in \{1,\ldots,p-1,p+1,\ldots,n\}$ has a weight of b_j and a profit of $c_j b_j + f_j$, while the d^p items $j \in \{n+1,\ldots,n+d^p\}$ compose the binary decomposition of the remaining offer $UB_{x_p} - LB_{x_p}$ on the partial node p after an initial number of shipments LB_{x_p} has been made from it. Each item $j \in \{n+1,\ldots,n+d^p-1\}$ has a weight of $\hat{b}_j = 2^{j-n-1}$ and the item $j = n+d^p$ has a weight of $\hat{b}_j = (UB_{x_p} - LB_{x_p}) - \sum_{j=n+1}^{n+d^p-1} 2^{j-n-1}$. Each item $j \in \{n+1,\ldots,n+d^p\}$ has a profit of $c_p\hat{b}_j$.

This decomposition method has been introduced by Martello and Toth [18]. By selecting one of the 2^{d^p} possible subsets of the items $\{n+1,\ldots,n+d^p\}$, any total weight in the set $\{0,1,\ldots,UB_{x_p} - LBx_p\}$ and no weight outside this range can be obtained.

Setting a variable $\bar{y}_j$ to 0 in the KP (42)-(44) for $j \in \{1,\ldots,p-1,p+1,\ldots,n\}$ corresponds to using node j completely in the solution to P1, while j will be unused if $\bar{y}_j = 1$. Applying the filtering methods detailed in Sections 4.2 and 4.3, one can directly set

$$\bar{y}_j = \begin{cases} 0 & \forall\ j \in \{1,\ldots,p-1,p+1,\ldots,n\} : j \in R, \\ 1 & \forall\ j \in \{1,\ldots,p-1,p+1,\ldots,n\} : j \in T \cup \{i : i \succ p\}. \end{cases} \tag{45}$$

Furthermore, setting a variable $\bar{y}_j$ to 1 for $j \in \{n+1,\ldots,n+d^p\}$ corresponds to reducing the number of shipments to be made from p by $\hat{b}_j$. Consequently, this decomposition allows the number of units to be used on node p to take on any value $x_p \in \{LB_{x_p},\ldots,UB_{x_p}\}$ by introducing only $d^p \in \mathcal{O}\big(\log(b_p)\big)$ additional items to the knapsack subproblem.

Letting $(\bar{\boldsymbol{y}}^*)$ be an optimal solution to (42)-(44), an optimal solution $(\boldsymbol{x}^1, \boldsymbol{y}^1) \in \tilde{S}^1$ to (P1$|p$ is partial) is given by

$$y_j^1 = \begin{cases} 1 & \text{for } j = p, \\ 1 - \bar{y}_j^* & \text{for } j = 1,\ldots,p-1,p+1,\ldots,n, \end{cases} \tag{46}$$

$$x_j^1 = \begin{cases} UB_{x_p} - \sum_{i=n+1}^{n+d^p}\hat{b}_j\bar{y}_i^* & \text{for } j = p, \\ b_j(1-\bar{y}_j^*) & \text{for } j = 1,\ldots,p-1,p+1,\ldots,n, \end{cases} \tag{47}$$

and its objective value is Z_p^4.

Solving P4$_p$ for each candidate $p \in P$ to find an optimal solution $(\boldsymbol{x}^1, \boldsymbol{y}^1) \in \tilde{S}^1$ to (P1$|p$ is partial) with objective value Z_p^4 marks the end of KTA.

6 General Outline of KTA

This section organizes and summarizes how the models and the different bounds developed in the preceding sections are integrated in our algorithm. In both the heuristic and the exact phases, knapsack subproblems must be solved. To do so, COMBO, the state-of-the-art algorithm presented in [19], is employed. The default parameters of COMBO are used.

6.1 Negative Costs Transformation

It has previously been assumed that all the fixed and variable costs f_j and c_j are nonnegative. The principal instances of the SSFCTP previously explored in the literature also respect this assumption. However, the SSFCTP arises as a Lagrangian relaxation subproblem of the FCTP ([3], [4]). In these cases, the unit costs c_j of the SSFCTP are obtained by subtracting a positive dual variable from the unit costs of the FCTP. These important instances can consequently contain negative unit costs c_j. Fortunately, a simple transformation can be performed to center the minimal unit cost c_j to 0 as well as to to eliminate any negative fixed cost f_j.

Proposition 6.1. *Redefining the unit costs by $\hat{c}_j = c_j - \theta \ \forall j \in \{1, \ldots, n\}$, where $\theta = \min\{c_j\}$, and solving the resulting problem $\hat{P}1$ leads to the same optimal solution $(\boldsymbol{x}^1, \boldsymbol{y}^1)$ as for the original problem, with optimal value $\hat{Z}^1 = Z^1 - D\theta$.*

Proposition 6.2. *Fixing $y_j = 1$ for each node j such that $f_j < 0$ and solving the resulting reduced problem leads to an optimal solution to the original problem, with the same objective value.*

6.2 Heuristic Phase

The execution of the algorithm itself begins by performing the heuristic phase, which essentially consists in calculating the bounds given in Section 3. It can be decomposed as follows:

H1)

1. Calculate the lower and upper bounds $Z_{\text{LB}}^{\text{LP}}$ and Z_{UB}^{G}. Set the global lower and upper bounds Z_{LB} and Z_{UB} to $Z_{\text{LB}}^{\text{LP}}$ and Z_{UB}^{G} respectively.
2. If $Z_{\text{LB}} = Z_{\text{UB}}$, an optimal solution has been found and the algorithm terminates.

The time complexity of this step is $\mathcal{O}(n)$.

H2)

1. Apply `P3Search`. Terminate the execution after 5 iterations if the halting conditions have not been met earlier. If $\min_{\lambda \in L}\{Z_{\text{UB}}^{\text{P3}(\lambda)}\} < Z_{\text{UB}}$, update the incumbent solution.
2. If $Z_{\text{UB}} > Z^3(\lambda_0)$, where $\lambda_0 = \min\{\lambda \in L\}$, solve P3(0) and update the incumbent solution accordingly. Otherwise, proceed to point 3. Since $Z_{\text{UB}}^{\text{P3}(0)} \leq Z^3(0)$ by definition of $Z_{\text{UB}}^{\text{P3}(\lambda)}$, solutions without a partial node can now be ignored by Proposition 3.5.
3. Calculate the lower bound $C_{\min}^{\text{P3}}$ on the partial node's variable cost.
4. Let $P = \left\{j \in \{1, 2, \ldots, n\} : c_j > C_{\min}^{\text{P3}}\right\}$ denote the set of nodes which could strictly improve the incumbent solution when fixed as the partial node. Calculate the lower bound $Z_{\text{LB}(p)}^{\text{P3}}$ and initialize the bound $Z_{\text{LB}(p)}$ to this value for each node $p \in P$. Remove p from the set of candidates P if $Z_{\text{LB}(p)} \geq Z_{\text{UB}}$. Otherwise, set the bounds LB_{x_p} and UB_{x_p} to $LB_{x_p}^{\text{P3}}$ and $UB_{x_p}^{\text{P3}}$ respectively.
5. If $P = \emptyset$, an optimal solution has been found and the execution of KTA is completed.

The dominant operation is the execution of the `P3Search` algorithm. Since P3(λ) is solved for at most a constant number of multiplier values, the time complexity of this step is the same as solving a KP with n items and capacity $(\sum_{j=1}^{n} b_j - D)$. It is given by $\mathcal{O}(2^{t-s+1})$ in worst case, where s and t correspond to the bounds of the decision core built in COMBO.

H3)

1. For each node $p \in P$, calculate bound $LB_{x_p}^{\text{Dom}}$ and update LB_{x_p} accordingly. For computational efficiency concerns, only consider suppliers $q \in P$ as potential dominant nodes in (28). Remove p from the set of candidates if $LB_{x_p} > UB_{x_p}$.

Denoting the cardinality of set P at the beginning of a step S by $|P_{\text{S}}|$, the time complexity of step H3 is bounded by $\mathcal{O}(|P_{\text{H3}}|^2)$ in worst case.

H4)

1. For each node $p \in P$, calculate the bound $Z_{\text{LB}(p)}^{\text{LP}}$ and update $Z_{\text{LB}(p)}$ accordingly. If $Z_{\text{LB}(p)} < Z_{\text{UB}}$, calculate $LB_{x_p}^{\text{LP}}$ and $UB_{x_p}^{\text{LP}}$ to improve LB_{x_p} and UB_{x_p}.
2. If $P = \emptyset$ after this step, an optimal solution has been found and the algorithm terminates.
3. If the optimal solution does not have a partial node, it has necessarily already been found during step H2. Setting the global lower bound Z_{LB} to $\min_{p \in P}\{Z_{\text{LB}(p)}\}$ is therefore valid and terminates the heuristic phase.

Using a naive implementation, the time complexity of this step is $\mathcal{O}(|P_{\text{H4}}| \cdot n)$. However, respecting the strict total order $\prec$ on the nodes of set P when computing the values $Z_{\text{LB}(p)}^{\text{LP}}$ makes it possible to effectively adjust the solution to the strong linear relaxation when updating the partial node p. Doing so, the complexity of this step can be reduced to $\mathcal{O}\big(n \log(n) + |P_{\text{H4}}| \cdot \min\{n, \frac{\max\{b_j\}}{\min\{b_j\}}\}\big)$. See Appendix 2 for an explanation of this theoretical complexity.

6.3 Filtering Phase

The SSFCTP is solved exactly by finding an optimal solution to (P1|p is partial) for each remaining candidate $p \in P$. The filtering phase is applied prior to the exact resolution of these subproblems in order to reduce their size.

F)

1. Execute the strict total order filtering of Section 4.2 to identify nodes $j \in \{i \in Q : i < s^q\}$ that cannot be unused and nodes $j \in \{i \in Q : i > s^q\}$ that cannot be complete in an optimal solution from $\tilde{S}^1$.
2. Apply the strong filtering method of Section 4.3 to the nodes that have not been fixed by the strict total order filtering. The resulting sets R and T respectively contain lists of nodes $j \in Q$ that must be complete and unused in a solution $(\boldsymbol{x}^1, \boldsymbol{y}^1) \in \tilde{S}^1$ to (P1|p is partial) for any node $p \in P$ such that $p \neq j$.

The time complexity of this step is $\mathcal{O}\big(n \cdot \min\{n, \frac{\max\{b_j\}}{\min\{b_j\}}\}\big)$ in worst case, as detailed in Appendix 2.

6.4 Exact Phase

Using the sets R and T constructed in the previous phase, the KPs (P1|p is partial) are reduced and solved exactly for each p in P.

E)

1. For each node $p \in P$, find an optimal solution to P4_p by solving the KP (42)-(44) after reducing its size by applying the filtering rules (45). If Z_p^4 improves the incumbent value Z_{UB}, update the incumbent solution by applying equations (46) and (47).
2. At the end of this step, the incumbent solution $(\boldsymbol{x}^*, \boldsymbol{y}^*) \in \tilde{S}^1$ is optimal. Its objective value is $Z_{\text{UB}} = Z$.

The exact phase has a complexity of $\mathcal{O}(|P_{\text{E}}| \cdot 2^{t-s+1})$, where s and t correspond to the bounds of the decision core built in COMBO.

In summary, a total of $\mathcal{O}(1 + |P_{\text{E}}|)$ KPs have to be solved during steps H2 and E. The complexity of the other steps of KTA is bounded by $\mathcal{O}\big(n \log(n) + |P_{\text{H3}}|^2 + n \cdot \min\{n, \frac{\max\{b_j\}}{\min\{b_j\}}\}\big) \subseteq \mathcal{O}(n^2)$.

7 Computational Experiments

There are two main objectives guiding our computational experiments. The first is to compare the performance of our algorithm with that of the state-of-the-art algorithms from the SSFCTP literature as well as a recent MIP solver on a range of original and existing classes of instances. The second objective is to provide an in-depth analysis of the performance of KTA and its constituent steps.

To do so, our algorithm, which has been coded in C, has been compared to the original C implementation of the enumerative algorithm (EA) and the dynamic programming algorithm (DP) proposed by Klose [1], that are the state of the art in the literature, and to Gurobi's MIP solver (version 9.1.1).

All experiments were conducted on a machine with an Intel(R) i7-10875H CPU @ 2.30GHz along with 32 GB of RAM operated with Windows 10 Pro (version 1909) using the Gnu C compiler (version 10.2.0). The instances and detailed results are available at `https://github.com/robinlegault/SSFCTP`

7.1 Classes of Problems

Four different classes of problems are considered, for each of which a set of parameters is covered. These four groups represent a total of 77 types of problems. Initially, 10 instances have been generated using each of them, for a total of 770 instances.

The first two classes are inspired by Pisinger [20], who studied KPs of varying hardness, while the last two are obtained using the generation methods described by Klose [1]. Each class of problems defines a specific generation method for the demand D of the sink and the capacity b_j, the unit cost c_j, the fixed cost f_j and the total cost $u_j = c_j b_j + f_j$ of node $j \in \{1, \dots, n\}$.

- Group 1: **Uncorrelated data instances**

  This instance group is inspired by the *Uncorrelated data instances* described by Pisinger [20] for the KP. It aims to produce easy instances of different sizes that preserve approximately the same structure independently of the number of nodes generated.

  When generating a group of N instances of n suppliers $j \in \{1, \dots, n\}$, the instance $i \in \{1, \dots, N\}$ is defined as follows:

  $b_j \overset{\text{iid}}{\sim} \text{unif}\{0.5n, n\}$,

  $c_j \overset{\text{iid}}{\sim} \text{unif}(8, 12)$,

  $f_j \overset{\text{iid}}{\sim} \text{unif}\{5n, 10n\}$,

  $$D_i = \frac{i}{N+1} \sum_{j=1}^{n} b_j.$$

  A set of $N = 10$ problems has been tested for each value of $n \in \{500, 1000, 5000, 10000, 25000\}$.

  Increasing simultaneously the number of suppliers, their capacity and their fixed cost as well as the demand of the sink limits the influence of n on the structure of the problem. In particular, the proportion of instances for which the optimal solution contains a partial node remains approximately constant regardless of the problem size. The definition of the demands D_i aims to vary the percentage of suppliers that will be used in the optimal solution to each instance of a same parametrization. This way, the performance measure of the algorithms to be tested ought to be more robust to variations in the problems' structure.

- Group 2: **Correlated data instances**

  This instance group is inspired by the *Almost strongly correlated instances* proposed in

[20] for the KP. The number of nodes and the value of the correlation parameter can be adjusted to produce instances of scalable difficulty having different structures.

The instance $i \in \{1, \ldots, N\}$ with n suppliers $j \in \{1, \ldots, n\}$ is defined as follows:

$$
\begin{aligned}
&b_j \overset{\text{iid}}{\sim} \text{unif}\{5000, 10000\},\\
&u_j = b_j + \alpha_j \cdot \frac{10000}{\beta}, \quad \text{where} \quad \alpha_j \overset{\text{iid}}{\sim} \text{unif}(-1, 1),\\
&f_j \overset{\text{iid}}{\sim} \text{unif}\{0.3 \cdot u_j, 0.5 \cdot u_j\},\\
&c_j = \frac{u_j - f_j}{b_j},\\
&D_i = \frac{i}{N+1} \sum_{j=1}^{n} b_j.
\end{aligned}
$$

A set of $N = 10$ problems has been generated for each of the 20 combinations of parameters with $n \in \{500, 1000, 5000, 10000, 25000\}$ and $\beta \in \{5, 10, 100, 1000\}$.

While n controls the number of suppliers, β is the correlation parameter linking the linearized cost of a node to its capacity. This correlation increases with β, which, up to a point, makes it more difficult to filter nodes. Consequently, when β and n are both set to a high value, one could expect the resulting problem to be harder. However, when $\beta \to +\infty$, the linearized cost e_j converges to 1 almost surely for each node. In this case, the structure of the SSFCTP becomes very similar to that of a subset-sum problem. More precisely, any solution $(\mathbf{y}, \mathbf{z}) \in S^2$ respecting $\sum_{j=1}^{n} b_j y_j = D$ would be optimal, with an objective value $Z = D$ that reaches the classical linear relaxation lower bound $Z_{\text{LB}}^{\text{LP}}$. Hence, this type of instance could in fact be relatively simple to solve. Indeed, since the capacities b_j take their value in a fixed interval, finding a subset of nodes whose total capacity is exactly D becomes easier as the number of nodes increases.

- Group 3: **Uncorrelated data instances with fixed demand**
  This class of instances is obtained by the generation method used by Klose [1] for his first group of test problems. It produces instances of different sizes that have a constant demand.

  The properties of these uncorrelated instances are determined by two parameters. The b-ratio $B_r = 100 \cdot D / \sum_{j=1}^{n} b_j$ estimates the percentage of nodes that are used in an optimal solution, while the f-ratio $F_r = \bar{f}/(\bar{c}\bar{b})$ is the ratio between the average fixed cost $\bar{f} = (1/n) \sum_{j=1}^{n} f_j$ and the product of the average unit cost $\bar{c} = (1/n) \sum_{j=1}^{n} c_j$ with the average capacity $\bar{b} = (1/n) \sum_{j=1}^{n} b_j$ of the suppliers.

  The instance $i \in \{1, \ldots, N\}$ with n suppliers $j \in \{1, \ldots, n\}$ is first defined as follows:

$$
\begin{aligned}
&\tilde{b}_j \overset{\text{iid}}{\sim} \text{unif}(3, 5),\\
&c_j \overset{\text{iid}}{\sim} \text{unif}(8, 12),\\
&\tilde{f}_j \overset{\text{iid}}{\sim} \text{unif}(3, 5),\\
&D_i = 100{,}000.
\end{aligned}
$$

The capacities b_j and the fixed costs f_j are then obtained by scaling the values $\tilde{b}_j$ and $\tilde{f}_j$ to meet the desired b-ratio and f-ratio. Finally, they are rounded to the nearest integer. A set of $N = 10$ problems has been tested for each of the 48 combinations of parameters used in [1], i.e. the parametrizations with $n \in \{500, 1000, 5000, 10000\}$, $B_r \in \{5, 10, 25, 50\}$ and $F_r \in \{0.3, 0.6, 1.0\}$.

This generation method, first introduced by Herer et al. [2], produces very easy uncorrelated instances. In particular, since the demand D does not depend on the number of suppliers, as n grows, the range of capacities b_j becomes excessively small, leading to highly structured instances in which many suppliers i *strictly dominate* other nodes j, as defined in [2] (i.e. $b_j \leq b_i$ and $f_i + c_i x \leq f_j + c_j x \ \ \forall\ x \in \{1, \ldots, b_j\}$).

- Group 4: **Correlated data instances with small capacities**
  This class of instances is obtained by the generation method used by Klose [1] for his second group of test problems. The generation method introduces a strong correlation between the capacity of a node and its total cost.

  The instance $i \in \{1, \ldots, N\}$ with n suppliers $j \in \{1, \ldots, n\}$ is defined as follows:

  $$b_j \overset{\text{iid}}{\sim} \text{unif}\{10, 100\},$$
  $$u_j \overset{\text{iid}}{\sim} \text{unif}\{b_j, 2 \cdot b_j\},$$
  $$f_j = \phi_j \cdot u_j, \quad \text{where} \quad \phi_j \overset{\text{iid}}{\sim} \text{unif}(0.75, 1),$$
  $$c_j = \frac{u_j - f_j}{b_j},$$
  $$D_i = 0.5 \sum_{j=1}^{n} b_j.$$

  A set of $N = 10$ problems has been tested for each number of suppliers $n \in \{500, 1000, 5000, 10000\}$.

  This generation method is the only one that was used by Klose [1] to test his algorithms on correlated data instances. However, this class of problems is once again highly structured. First, the capacities of the suppliers are small and undiversified. Also, since the values ϕ_j are taken from the interval $(0.75, 1)$, the importance of the fixed cost significantly exceeds that of the unit costs. This property makes these instances very similar to KPs. Indeed, if ϕ_j was equal to 1 for each node, the resulting SSFCTP would be equivalent to a Min-KP with weight b_j and cost f_j on item j and demand D. We can consequently expect step H2 of KTA, which specifically solves KPs that approximate the original SSFCTP, to offer excellent bounds for this class of problems.

7.2 Computational Results

We now present the average computation time required by KTA, by the DP and EA methods of Klose [1], and by Gurobi to solve instances from each group of problems. For each parametrization, $N = 10$ instances have been solved by each method, with a time limit of 5 minutes (300,000 ms) per instance. By construction of the SSFCTP, the relative optimality gap obtained by elementary lower and upper bounds mechanically decreases as the number of suppliers and the similarity between linearized costs increases. To adequately compare the performance of the

Table 2: Average CPU times (ms) for Group 1 (Uncorrelated)

n	KTA	DP	EA	Gurobi
500	0.1	28.7	6.6	137.8
1,000	0.3	163.4	48.8	192.6
5,000	1.1	6,892.7	838.1	1,361.2
10,000	2.2	39,541.4	1,943.6	2,665.6
25,000	6.0	66,384.5$^{(7)}$	20,879.1	10,612.9

$^{(\cdot)}$ Number of instances, out of 10, that were not solved to optimality within the time limit of 5 minutes (300,000 ms) or could not be solved due to a memory error. The reported average is only based on the instances that were solved within the time limit.

algorithms, the optimality tolerance has therefore been set to 0. The other default parameters of the Gurobi MIP solver were left unchanged.

Results from Table 2 clearly show that our algorithm significantly outperforms the other methods for the uncorrelated data instances of Group 1. The time required by KTA to solve these instances seems to grow linearly with n, while the empirical time complexity of DP and EA show exponential growth. Moreover, 7 of the 10 instances of size $n = 25{,}000$ could not be solved by the DP algorithm within the 5 minutes time limit. Interestingly, as the size of the problems grows, the general-purpose Gurobi MIP solver starts to outperform the two algorithms of Klose [1], which still represented the state of the art of the SSFCTP literature.

The same trends emerge even more clearly from Table 3. In addition, while the computing time required by DP, EA and Gurobi drastically increases with the correlation parameter β, the performance of KTA remains essentially unchanged by β for a given number of suppliers. Even more, for $n = 25{,}000$ and $\beta = 1{,}000$, where both DP and EA were unable to solve a single instance in less than 300,000 ms, KTA only required 6.0 ms on average, which is 33% less than the time required for the weakly correlated instances that were generated with $\beta = 5$. In comparison, the time required by Gurobi was multiplied by more than 7, from 13,558.3 ms to 96,583.9 ms, for the same pair of parametrizations. As explained previously, the structure of Group 2 instances become very similar to that of the subset-sum problem when both n and β grow, making the resulting problems straightforward to solve by appropriate approaches. These results confirm that the procedure used by KTA effectively exploits this structure. More specifically, step H2 systematically identifies an optimal solution and proves its optimality for Group 2 test problems with $n \geq 10{,}000$ and $\beta \geq 100$ by solving a single KP, which can be done in a few milliseconds (see Section 7.4 for further details).

The problems of Groups 3 and 4 once again confirm the superiority of our algorithm over the existing methods. As noticed by Klose [1], the execution time of both DP and EA increases as the f-ratio approaches 1. While this same behaviour is clearly observable for Gurobi, KTA shows the opposite tendency, especially for the problems of larger size. To explain this, we can note that the probability for the optimal solution to contain a partial node decreases for large values of n. As shown in Table 6, while 90% of the optimal solutions to the problems of Group 3

Table 3: Average CPU times (ms) for Group 2 (Correlated)

n	β	KTA	DP	EA	Gurobi
500	5	0.3	1,226.9	2.5	104.7
	10	0.3	892.2	4.3	143.0
	100	0.5	5,484.6	117.6	1,045.7
	1,000	1.6	17,942.9	1,658.4	3,538.3
1,000	5	0.4	1,103.9	7.1	124.7
	10	0.4	1,446.7	16.9	194.3
	100	0.9	10,770.8	766.0	1,623.8
	1,000	1.4	56,891.8	7,708.3	5,814.7
5,000	5	1.7	7,530.5	367.5	829.6
	10	1.7	9,136.5	1,709.7	693.7
	100	1.9	79,393.8	31,709.9	13,121.3
	1,000	1.7	24,092.2$^{(9)}$	139,554.8$^{(5)}$	17,871.7
10,000	5	3.2	10,215.4	977.6	2,274.1
	10	3.4	8,389.9	3,393.3	3,238.0
	100	2.4	84,854.4$^{(6)}$	107,727.6	26,031.2
	1,000	3.4	44,854.4$^{(8)}$	215,663.3$^{(8)}$	54,575.1
25,000	5	9.0	22,975.6$^{(1)}$	13,533.5	13,558.3
	10	6.2	39,935.5	30,926.3	22,579.6
	100	6.7	66,884.5$^{(7)}$	84,229.2$^{(9)}$	67,885.7
	1,000	6.0	*	*	96,583.9

$*$ None of the 10 instances has been solved to optimality within the time limit.

Table 4: Average CPU times (ms) of KTA, DP and EA for Group 3 (Uncorrelated with fixed demand)

n	B_r	KTA, F_r			DP, F_r			EA, F_r			Gurobi, F_r		
		0.3	0.6	1	0.3	0.6	1	0.3	0.6	1	0.3	0.6	1
500	5	0.3	0.3	0.3	570.4	1,271.4	1,568.0	0.6	8.5	7.5	77.9	92.3	66.5
	10	0.3	0.2	0.3	268.2	335.7	553.5	1.0	3.9	26.4	95.7	120.9	88.1
	25	0.3	0.2	0.2	73.8	111.9	112.3	5.1	4.2	14.0	96.5	83.8	108.9
	50	0.2	0.1	0.1	24.8	41.6	48.5	4.6	8.3	16.1	90.8	108.8	127.1
1,000	5	0.5	0.5	0.5	348.8	390.1	855.5	3.9	20.0	42.4	85.9	106.8	214.5
	10	0.5	0.4	0.4	84.5	240.1	361.9	10.1	44.4	42.1	86.1	149.9	165.7
	25	0.4	0.3	0.2	50.2	60.9	73.4	9.5	20.2	31.0	135.2	145.2	171.5
	50	0.3	0.1	0.2	19.5	17.3	45.5	6.0	35.8	37.0	178.6	123.6	226.3
5,000	5	2.0	1.1	1.2	84.8	166.4	312.6	157.6	256.6	19,856.7	240.2	325.5	397.0
	10	1.7	0.5	0.4	73.5	116.2	116.7	101.8	660.5	337.0	257.1	355.2	324.1
	25	1.3	0.4	0.4	34.8	44.0	64.4	62.8	89.3	106.5	355.8	387.1	380.7
	50	0.4	0.4	0.4	20.1	30.8	62.9	40.5	83.4	167.1	301.7	398.9	612.8
10,000	5	3.7	1.3	0.9	80.9	155.6	324.3	524.2	3,432.2	13,799.1	433.1	515.4	675.7
	10	3.1	0.8	0.8	75.3	167.9	114.8	218.9	414.3	3,123.2	491.2	421.4	736.0
	25	2.1	0.8	0.9	21.5	75.9	53.7	72.3	239.0	297.6	424.9	716.0	723.6
	50	0.8	0.8	0.7	18.2	35.7	40.1	85.0	229.9	198.0	568.2	826.0	663.7

Table 5: Average CPU times (ms) of KTA, DP and EA for Group 4 (Correlated with small capacities)

n	KTA	DP	EA	Gurobi
500	0.1	1.6	5.5	52.2
1,000	0.1	3.2	149.6	118.3
5,000	0.5	15.0	$20{,}429.4^{(5)}$	371.1
10,000	0.9	24.5	$115.5^{(8)}$	760.2

Table 6: Percentage of instances with an optimal solution having a partial node by group

n	Group 1	Group 2, $\beta = 5$	Group 2, $\beta = 10$	Group 2, $\beta = 100$	Group 2, $\beta = 1{,}000$	Group 3	Group 4
500	60	100	97	80	19	90	0
1,000	68	96	92	67	7	58	0
5,000	69	84	76	19	0	9	0
10,000	77	80	66	1	0	1	0
25,000	71	63	47	0	0	-	-

contained a partial node when $n = 500$, this frequency dropped to only 1% for $n = 10{,}000$. This fact is coherent with the reduced capacities which come with high values of n for the generation method of Group 3 test problems. Moreover, as F_r increases, the structure of the SSFCTP gets closer to that of a Min-KP, since the contribution to the objective of the fixed costs becomes proportionally more important. Hence, our approximation of the SSFCTP by a KP becomes more precise as F_r increases and bounds from step H2 are therefore strengthened. This leads to the observed reduction in computing time.

7.3 Analysis of the Instance Generation Methods

To support the analysis of the four classes of problems that were tested, we computed the proportion of data instances for which the optimal solution contains a partial node on an extended collection of instances. Each entry of Table 6, except for Group 3, has been computed on a set of $N = 100$ new instances using common random numbers (CRN) to minimize the variance in results due to the data generation process and therefore better isolate the influence of the parameters on the problem structure. For Group 3, the same 120 instances that were used in the previous section for each value of n has been preserved to cover the range of f-ratios and b-ratios that were considered by Klose [1] and Herer et al. [2].

This measure is an important indicator of the structure of the SSFCTP. Indeed, it appears that, for some groups of instances, the optimal solution never includes a partial node. For such problems, the distinction between the fixed cost f_j and the transportation cost $c_j b_j$ of a supplier completely disappears, since the total cost u_j is systematically paid on each selected node. More importantly, this means that solving a single Min-KP, given by model P3(λ), with $\lambda = 0$, always suffices to find the optimal solution for these instances. Notably, the optimal solution to each of the 440 instances of Group 4 that we generated did not contain a partial node. Furthermore, when using KTA, `P3Search` required the resolution of one single KP per instance and the bounds of step H2 sufficed in each case to prove the optimality of the resulting solution. Thus, not a single computation has been executed during steps H3, H4, F and E for problems of Group 4. We can therefore conclude that the supposedly hard instances considered by Klose [1] can now be regarded as relatively simple instances of the Min-KP.

As expected, the generation method of Group 1 test problems seems to preserve the same structure for every problem size, with approximately 70% of the optimal solutions containing

Table 7: Average CPU time (ms) per step of KTA for Group 1

n	H1	H2	H3	H4	F	E	Total
500	0.0	0.1	0.0	0.0	0.0	0.0	0.1
1,000	0.0	0.2	0.0	0.0	0.0	0.0	0.3
5,000	0.1	0.7	0.0	0.3	0.0	0.0	1.1
10,000	0.2	1.3	0.1	0.6	0.0	0.0	2.2
25,000	0.5	4.0	0.2	1.4	0.0	0.0	6.0
Average	0.2	1.3	0.1	0.5	0.0	0.0	1.9

a partial node. This property is desirable, since it makes it possible to test the influence of problem size on the performance of algorithms without introducing a bias in the analysis that would be due to an underlying transformation of the optimal solution properties.

In this regard, the generation method of Group 2 is complementary to that of Group 1. Its parameters jointly transform the problem structure from a pure SSFCTP for which the optimal solution always requires a partial node and is therefore quite distinct from a KP, to a problem that closely resembles the KP and the subset-sum problem as n and β increase. Together, these two classes of problems permit a robust analysis of the performance of algorithms on a range of SSFCTPs that aims to cover most of the important problem structures that are likely to emerge from future application contexts.

7.4 Performance Analysis of KTA

Tables 7 and 8 present the average computation time required by each step of the algorithm to solve the instances of Groups 1 and 2 that were used in Section 7.2. The majority of the execution time, respectively 65% for Group 1 and 72% for Group 2, has been spent in H2, the knapsack transformation step. Overall, H4, the strong linear relaxation step, was the second most time-consuming, followed by H3, the dominance relation step and H1, the elementary bounds phase. The filtering and exact phases together required less than 1% of the total computing time for both groups.

These computation times reflect quite directly the relative contribution of each step to the whole resolution process. A notable fact is that, for each of the 3,670 data instances we considered, the optimal solution was identified by the knapsack transformation method of step H2. In other words, the equality $Z_{\mathrm{UB}}^{\mathrm{P3}} = Z$ held for each problem and the exact phase could never improve this bound. Although it is easy to build very small artificial instances such that $Z_{\mathrm{UB}}^{\mathrm{P3}} > Z$, this never happens on our test instances. In practice, the only purpose of the subsequent steps, including the exact phase, is therefore to prove the optimality of the solution identified in step H2.

Tables 9 and 10 show the remaining percentage of partial nodes candidates after steps H2, H3 and H4. For Group 1 instances, the observed percentages after each step are relatively stable from one size of problem to another. The knapsack transformation step filters approximately

Table 8: Average CPU time (ms) per step of KTA for Group 2

n	β	H1	H2	H3	H4	F	E	Total
500	5	0.0	0.1	0.0	0.0	0.0	0.0	0.3
	10	0.0	0.2	0.0	0.0	0.0	0.0	0.3
	100	0.0	0.4	0.0	0.0	0.0	0.0	0.5
	1,000	0.0	1.5	0.0	0.0	0.0	0.0	1.6
1,000	5	0.0	0.2	0.1	0.1	0.0	0.0	0.4
	10	0.0	0.2	0.0	0.1	0.0	0.0	0.4
	100	0.0	0.8	0.0	0.0	0.0	0.0	0.9
	1,000	0.0	1.4	0.0	0.0	0.0	0.0	1.4
5,000	5	0.1	0.9	0.3	0.4	0.0	0.0	1.7
	10	0.1	1.0	0.2	0.4	0.0	0.0	1.7
	100	0.1	1.7	0.0	0.0	0.0	0.0	1.9
	1,000	0.1	1.7	0.0	0.0	0.0	0.0	1.7
10,000	5	0.2	1.5	0.8	0.7	0.1	0.0	3.2
	10	0.2	1.9	0.5	0.8	0.0	0.0	3.4
	100	0.2	2.2	0.0	0.0	0.0	0.0	2.4
	1,000	0.2	3.2	0.0	0.0	0.0	0.0	3.4
25,000	5	0.5	3.9	2.4	2.0	0.2	0.1	9.0
	10	0.6	3.5	0.8	1.3	0.0	0.0	6.2
	100	0.5	6.2	0.0	0.0	0.0	0.0	6.7
	1,000	0.5	5.5	0.0	0.0	0.0	0.0	6.0
Average		0.2	1.9	0.3	0.3	0.0	0.0	2.7

Table 9: Percentage of nodes that can be partial after each filtering step for Group 1

n	H2	H3	H4
500	0.64	0.32	0.00
1,000	3.86	0.66	0.00
5,000	2.70	0.16	0.00
10,000	3.09	0.10	0.00
25,000	2.48	0.06	0.00
Average	2.55	0.26	0.00

97.5% of the candidates on average. From there, the dominance step divides the cardinality of P by approximately 10 and the strong linear relaxation eliminates virtually all the remaining candidates. The exact phase is therefore almost never necessary for the instances of Group 1. For Group 2 test problems, as the correlation parameter increases, the average percentage of candidates remaining after H2 goes from more than 25% for $\beta = 5$ to less than 0.1% for $\beta =$ 1,000. This result illustrates that this step of the algorithm is especially adapted to these last instances of the SSFCTP, which are by construction very similar to KPs or subset-sum problems. Fortunately, for small values of β, the bounds from H3 and H4 compensate for the weakness of those of H2. The percentage of potential partial nodes that have to be considered during the exact phase in consequently of only 0.03% on average for the correlated data instances.

Tables 11 and 12 give some details about the knapsack subproblems that are solved in both the heuristic and the exact phases. The KPs that are considered in the heuristic phase are solved within `P3Search` during step H2 and correspond to model P3(λ) for different reimbursement rates λ. They are therefore quite different in structure from the ones of the exact phase, which are solved during step E to obtain the optimal solutions to the subproblems (P1|p is partial) using model P4$_p$ for the remaining candidates $p \in P$. In the latter case, the filtering procedure described in step F significantly reduces the number of nodes that have to be considered in the KPs. Indeed, for both Group 1 and Group 2 test problems, for every problem size, the average number of items to be considered per knapsack subproblem was of approximately 40, which corresponds to 8% of the suppliers when $n = 500$ and less than 0.002% of them when $n = 25{,}000$. In addition, the exact phase was required for only 1 of the 40 instances of the first group of problems and for 35 of the 200 instances of the second one, for an average of less than 0.2 KP solved per instance. During the heuristic phase, 1.6 KP was required per instance for both groups on average.

In total, KTA required the resolution of fewer than 2 KPs per instance on average for both Group 1 and Group 2 test problems. Since the resolution of these subproblems represents the majority of the computation time of KTA, we can conclude that our new method, at least for the types of instances that were considered in this article, fills the gap between the state of the art on KPs and SSFCTPs.

Table 10: Percentage of nodes that can be partial after each filtering step for Group 2

n	β	H2	H3	H4
500	5	29.08	10.84	0.16
	10	17.40	7.94	0.24
	100	2.62	2.38	0.08
	1,000	0.34	0.32	0.04
1,000	5	27.86	8.55	0.05
	10	15.94	6.49	0.00
	100	0.59	0.57	0.01
	1,000	0.00	0.00	0.00
5,000	5	25.60	4.79	0.00
	10	13.27	3.55	0.00
	100	0.05	0.05	0.00
	1,000	0.00	0.00	0.00
10,000	5	24.95	3.67	0.00
	10	14.42	3.04	0.00
	100	0.00	0.00	0.00
	1,000	0.00	0.00	0.00
25,000	5	26.00	2.93	0.00
	10	7.44	1.26	0.00
	100	0.00	0.00	0.00
	1,000	0.00	0.00	0.00
Average		10.28	2.82	0.03

Table 11: KPs solved for Group 1

	Average number of KPs per instance			Properties of exact phase's KPs	
n	Heuristic phase	Exact phase	Total	Instances requiring the exact phase	Average number of items per KP in the exact phase
500	1.4	0.0	1.4	0	-
1,000	1.7	0.0	1.7	0	-
5,000	1.6	0.0	1.6	0	-
10,000	1.7	0.1	1.8	1	40.0
25,000	1.7	0.0	1.7	0	-
Average	1.6	0.0	1.6	0.2	40.0

Table 12: KPs solved for Group 2

		Average number of KPs per instance			Properties of exact phase's KPs	
n	β	Heuristic phase	Exact phase	Total	Instances requiring the exact phase	Average number of items per KP in the exact phase
500	5	2.4	0.8	3.2	5	27.9
	10	2.1	1.3	3.4	7	27.4
	100	1.8	0.4	2.2	4	44.0
	1,000	1.2	0.2	1.4	1	70.0
1,000	5	2.0	0.5	2.5	5	21.6
	10	2.0	0.0	2.0	0	-
	100	1.5	0.1	1.6	1	56.0
	1,000	1.0	0.0	1.0	0	-
5,000	5	2.0	0.2	2.2	2	37.5
	10	1.9	0.1	2.0	1	29.0
	100	1.1	0.1	1.2	1	56.0
	1,000	1.0	0.0	1.0	0	-
10,000	5	1.8	0.3	2.1	3	37.7
	10	1.9	0.3	2.2	2	36.3
	100	1.0	0.0	1.0	0	-
	1,000	1.0	0.0	1.0	0	-
25,000	5	1.9	0.3	2.2	3	39.3
	10	1.6	0.0	1.6	0	-
	100	1.0	0.0	1.0	0	-
	1,000	1.0	0.0	1.0	0	-
Average		1.6	0.2	1.8	1.8	33.6*

* Weighted by the number of KPs solved in the exact phase for each type of instances

8 Conclusion

We introduced a binary non-linear programming reformulation of the single-sink fixed-charge transportation problem. Using the knapsack problem that is obtained by a transformation of this new model, we developed several bounds, some of which are subsequently improved by a dominance relation between potential partial nodes and a strong linear relaxation to produce an efficient and robust heuristic method. A new filtering procedure allows us to complete the exact algorithm by iteratively fixing a partial node and solving a reduced-size knapsack problem. As shown on a large set of instances, the knapsack transformation algorithm completely outperforms the existing algorithms from the literature. In particular, a reduction of several orders of magnitude in the resolution time of the state-of-the-art methods occurs for large and highly structured instances of the SSFCTP.

Acknowledgments

This research was supported by the Natural Sciences and Engineering Research Council of Canada through the Discovery Grants 2017-06054 and 2021-00028. We thank A. Klose for making available the source code of his algorithms.

References

[1] Andreas Klose. Algorithms for solving the single-sink fixed-charge transportation problem. *Computers & Operations Research*, 35(6):2079–2092, 2008.

[2] Yale T. Herer, Meir J. Rosenblatt, and Ilan Hefter. Fast algorithms for single-sink fixed charge transportation problems with applications to manufacturing and transportation. *Transportation Science*, 30:276–290, 1996.

[3] Simon Görtz and Andreas Klose. The single-sink fixed-charge transportation problem: Applications and solution methods. In Hans-Otto Günther, Dirk C. Mattfeld, and Leena Suhl, editors, *Management logistischer Netzwerke*, pages 383–406, Heidelberg, 2007. Physica-Verlag HD.

[4] Aristide Mingozzi and Roberto Roberti. An exact algorithm for the fixed charge transportation problem based on matching source and sink patterns. *Transportation Science*, 52(2):229–238, 2018.

[5] Tue R. L. Christensen, Kim Allan Andersen, and Andreas Klose. Solving the single-sink, fixed-charge, multiple-choice transportation problem by dynamic programming. *Transportation Science*, 47(3):428–438, 2013.

[6] Mohammad Rahim Akhavan Kazemzadeh, Tolga Bektaş, Teodor Gabriel Crainic, Antonio Frangioni, Bernard Gendron, and Enrico Gorgone. Node-based lagrangian relaxations for multicommodity capacitated fixed-charge network design. *Discrete Applied Mathematics*, 308(C):255–275, 2022.

[7] Josef Haberl. Exact algorithm for solving a special fixed-charge linear programming problem. *Journal of Optimization Theory and Applications*, 69:489–529, 1991.

[8] Bahram Alidaee and Gary A. Kochenberger. A note on a simple dynamic programming approach to the single-sink, fixed-charge transportation problem. *Transportation Science*, 39(1):140–143, 2005.

[9] Silvano Martello and Paolo Toth. An upper bound for the zero-one knapsack problem and a branch and bound algorithm. *European Journal of Operational Research*, 1(3):169–175, 1977.

[10] Silvano Martello and Paolo Toth. Upper bounds and algorithms for hard 0-1 knapsack problems. *Operations Research*, 45(5):768–778, 1997.

[11] David Pisinger. A minimal algorithm for the 0-1 knapsack problem. *Operations Research*, 45(5):758–767, 1997.

[12] Balasubramanian Ram and Sanjiv Sarin. An algorithm for the 0-1 equality knapsack problem. *Journal of the Operational Research Society*, 39(11):1045–1049, 1988.

[13] David Pisinger. A minimal algorithm for the bounded knapsack problem. *INFORMS Journal on Computing*, 12(1):75–82, 2000.

[14] Silvano Martello and Paolo Toth. An exact algorithm for the two-constraint 0–1 knapsack problem. *Operations Research*, 51(5):826–835, 2003.

[15] Meir J. Rosenblatt, Yale T. Herer, and Ilan Hefter. Note. an acquisition policy for a single item multi-supplier system. *Management Science*, 44(11-part-2):S96–S100, 1998.

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

[17] Simon Görtz and Andreas Klose. Analysis of some greedy algorithms for the single-sink fixed-charge transportation problem. *Journal of Heuristics*, 15(4):331–349, 2009.

[18] Silvano Martello and Paolo Toth. *Knapsack Problems: Algorithms and Computer Implementations*. John Wiley & Sons, Inc., USA, 1990.

[19] Silvano Martello, David Pisinger, and Paolo Toth. Dynamic programming and strong bounds for the 0-1 knapsack problem. *Management Science*, 45(3):414–424, 1999.

[20] David Pisinger. Where are the hard knapsack problems? *Computers & Operations Research*, 32(9):2271–2284, 2005.

Appendix 1

Proof of Proposition 2.5 : First, $(\boldsymbol{x}^1, \boldsymbol{y}^1)$ respects constraint (2), since

$$\sum_{j=1}^{n} x_j^1 = \sum_{j=1}^{n} \Big(b_j y_j^2 - \big(\sum_{i=1}^{n} b_i y_i^2 - D\big) z_j^2 \Big) \overset{(8)}{=} \sum_{j=1}^{n} b_j y_j^2 - \Big(\sum_{j=1}^{n} b_j y_j^2 - D \Big) = D.$$

Now, let p be the only node such that $z_p^2 = 1$. Since $x_j^1 = b_j y_j^2 = b_j y_j^1 \ \forall j \neq p$, constraint (3) is respected for these nodes. Also, $y_p^1 = 1$ and $x_p^1 = b_p - (\sum_{i=1}^{n} b_i y_i^2 - D) \in^{(6),(7)} \{1, \ldots, b_p\}$. Hence, constraint (3) is respected. Finally, constraint (4) is trivially respected by construction of solution $(\boldsymbol{x}^1, \boldsymbol{y}^1)$. Consequently, $(\boldsymbol{x}^1, \boldsymbol{y}^1)$ is a feasible solution to P1.

The following shows that the objective values $Z^1(\boldsymbol{x}^1, \boldsymbol{y}^1)$ and $Z^2(\boldsymbol{y}^2, \boldsymbol{z}^2)$ are the same.

$$\begin{aligned}
Z^1(\boldsymbol{x}^1, \boldsymbol{y}^1) &= \sum_{j=1}^{n} (c_j x_j^1 + f_j y_j^1) && , \textit{ by } (1) \\
&= \sum_{j=1}^{n} \Big(c_j \Big(b_j y_j^2 - \big(\sum_{i=1}^{n} b_i y_i^2 - D\big) z_j^2 \Big) + f_j y_j^2 \Big), && \textit{by defn. of } (\boldsymbol{x}^1, \boldsymbol{y}^1) \\
&= \sum_{j=1}^{n} (c_j b_j + f_j) y_j^2 - c_p \Big(\sum_{j=1}^{n} b_j y_j^2 - D \Big) && , \textit{ by defn. of } x_p^1 \\
&= \sum_{j=1}^{n} (c_j b_j + f_j) y_j^2 - \Big(\sum_{j=1}^{n} b_j y_j^2 - D \Big) \sum_{j=1}^{n} c_j z_j^2 && , \textit{ by defn. of } p \\
&= Z^2(\boldsymbol{y}^2, \boldsymbol{z}^2)
\end{aligned}$$

□

Proof of Proposition 2.6 : By Proposition 2.5, a feasible solution to P1 is associated with any feasible solution to P2 and their objective value is the same. Hence, $Z^1 \leq Z^2$.

Now, let $(\boldsymbol{x}^*, \boldsymbol{y}^*)$ be an optimal solution to P1 in which at most one of the nodes that are used in the solution can be partial and in which no fixed cost is uselessly paid, i.e. $y_j^* \leq x_j^* \ \forall j \in \{1, \ldots, n\}$. Such an optimal solution to P1 exists by Proposition 2.4 and because setting $y_i = 1$ for a node i such that $x_i = 0$ cannot improve the solution since $f_j \geq 0 \ \forall j$ by hypothesis.

Let p be the partial node of $(\boldsymbol{x}^*, \boldsymbol{y}^*)$ if it exists and an arbitrary node j such that $y_j^* = 1$ otherwise. We will now consider the solution $(\boldsymbol{y}^2, \boldsymbol{z}^2) \in \{0,1\}^n \times \{0,1\}^n$ such that $z_j^2 = \mathbb{1}(j = p)$ and $y_j^2 = y_j^*$ for all $j \in \{1, \ldots, n\}$. $(\boldsymbol{y}^2, \boldsymbol{z}^2)$ is a feasible solution to P2. First, it respects constraint (6), since

$$\sum_{j=1}^{n} b_j y_j^2 = \sum_{j=1}^{n} b_j y_j^* \overset{(3)}{\geq} \sum_{j=1}^{n} x_j^* \overset{(2)}{=} D \tag{48}$$

Also, since $(\boldsymbol{x}^*, \boldsymbol{y}^*)$ is a feasible solution to P1 in which p is the only node that may be partial, the following equality holds.

$$\sum_{j=1}^{n} x_j^* = \sum_{\substack{j=1 \\ j \neq p}}^{n} b_j y_j^* + x_p^* \overset{(2)}{=} D \tag{49}$$

From there, $(\boldsymbol{y}^2, \boldsymbol{z}^2)$ also respects constraint (7), because

$$\begin{aligned}
\sum_{j=1}^{n} b_j y_j^2 - D &= \sum_{j=1}^{n} b_j y_j^* - D \\
&= \sum_{\substack{j=1 \\ j\neq p}}^{n} b_j y_j^* - D + b_p \\
&= \Big(\sum_{\substack{j=1 \\ j\neq p}}^{n} b_j y_j^* + x_p^*\Big) - D + b_p - x_p^* \\
&\overset{(49)}{=} b_p - x_p^* \\
&\leq b_p - 1 \qquad\qquad \text{, since } x_p^* \geq y_p^* = 1 \text{ by hypothesis} \\
&= \sum_{j=1}^{n} b_j z_j^2 - 1
\end{aligned}$$

Also, (8) is respected, since $\sum_{j=1}^{n} z_j^2 = \sum_{j=1}^{n} \mathbb{1}(j = p) = 1$. Finally, (9), as well as (10), directly follow from the construction of $(\boldsymbol{y}^2, \boldsymbol{z}^2)$. Consequently, $(\boldsymbol{y}^2, \boldsymbol{z}^2)$ is a feasible solution to P2.

Applying Proposition 2.5 to it leads to the feasible solution $(\boldsymbol{x}^1, \boldsymbol{y}^1)$ to P1, where $y_j^1 = y_j^2 = y_j^*$ and

$$x_j^1 = b_j y_j^2 - \Big(\sum_{i=1}^{n} b_i y_i^2 - D\Big) z_j^2 = \begin{cases} b_j y_j^* = x_j^* & \text{for } j = 1, \ldots, p-1, p+1, \ldots, n, \\ b_j - \big(\sum_{\substack{i=1 \\ i\neq p}}^{n} b_i y_i^* - D\big) \overset{(49)}{=} x_j^* & \text{for } j = p. \end{cases}$$

Hence, $(\boldsymbol{x}^1, \boldsymbol{y}^1) = (\boldsymbol{x}^*, \boldsymbol{y}^*)$. By Proposition 2.5, we conclude that $Z^2 \leq Z^1$. Therefore, $Z^1 = Z^2$. □

Proof of Proposition 2.8 : Constraint (11) enforces that the partial node p, in any solution to P2, if it exists, is the node with the maximal unit cost c_j among suppliers shipping a positive number of units. By Proposition 2.4, if an optimal solution to P1 containing a partial node p exists, then $y_j = 0$ for each node $j \in \{1, \ldots, n\}$ such that $c_j > c_p$. Hence, constraint (11) does not make infeasible any optimal solution to P2 that leads to an optimal solution to P1 through Proposition 2.5.

If an optimal solution $(\boldsymbol{y}^2, \boldsymbol{z}^2)$ to P2 contains a partial node i such that $c_i = c_j$ and $b_i < b_j$, then, using the $\sum_{j=1}^{n} b_j y_j^2$ excess units on j instead of i (i.e. fixing $y_i = 0$ and $y_j = 1$) would lead to another feasible solution with the same objective value that respects (11). Consequently, in case of equality on the maximal unit cost c_j among the nodes that are used in a solution to P2, the candidate with the greatest capacity b_j can systematically be selected.

Note that the case ($c_i = c_j$ and $b_i = b_j$ and $i < j$) results from an arbitrary choice. Its last condition could have been replaced by $i > j$ in Definition 2.7. It is however necessary to include

one or another of these conditions for $\prec$ to be a connected binary relation and thus a strict total order. □

Proof of Bijection $\tilde{f}_{2,1}$: First, let us demonstrate that $\tilde{f}_{2,1}$ is injective. Let $(\boldsymbol{y}^{2A}, \boldsymbol{z}^{2A}), (\boldsymbol{y}^{2B}, \boldsymbol{z}^{2B}) \in \tilde{S}^2$ be solutions such that $\tilde{f}_{2,1}(\boldsymbol{y}^{2A}, \boldsymbol{z}^{2A}) = (\boldsymbol{x}^{1A}, \boldsymbol{y}^{1A})$ and $\tilde{f}_{2,1}(\boldsymbol{y}^{2B}, \boldsymbol{z}^{2B}) = (\boldsymbol{x}^{1B}, \boldsymbol{y}^{1B})$.

Suppose that $(\boldsymbol{x}^{1A}, \boldsymbol{y}^{1A}) = (\boldsymbol{x}^{1B}, \boldsymbol{y}^{1B})$. This implies that $y_j^{1A} = y_j^{1B}$ for all $j \in \{1, \dots, n\}$. By definition of $\tilde{f}_{2,1}$, this means that $y_j^{2A} = y_j^{2B}$ for each supplier j.

Now, by definition of set $\tilde{S}^2$, $(\boldsymbol{y}^{2A}, \boldsymbol{z}^{2A})$ and $(\boldsymbol{y}^{2B}, \boldsymbol{z}^{2B})$ both respect constraint (11). Consequently, $z_j^{2A} = y_j^{2A} \mathbb{1}(j \succ i \ \forall\ i \neq j : y_i^{2A} = 1)$ and $z_j^{2B} = \mathbb{1}(j \succ i \ \forall\ i \neq j : y_i^{2B} = 1)$. Thus, the fact that $y_i^{2A} = y_i^{2B}\ \forall\ i$ implies that $z_j^{2A} = z_j^{2B}$ for all $j \in \{1, \dots, n\}$.

Hence, $(\boldsymbol{y}^{2A}, \boldsymbol{z}^{2A}) = (\boldsymbol{y}^{2B}, \boldsymbol{z}^{2B})$. Therefore, $\tilde{f}_{2,1}$ is injective.

Now, to demonstrate that $\tilde{f}_{2,1}$ is surjective, it suffices to refer to the proof of Proposition (2.6) given above. We have shown that, for an optimal solution $(\boldsymbol{x}^*, \boldsymbol{y}^*)$ to P1 in which at most one node is partial, a feasible solution to $(\boldsymbol{y}^2, \boldsymbol{z}^2) \in \tilde{S}^2$ can be build such that $\tilde{f}_{2,1}(\boldsymbol{y}^2, \boldsymbol{z}^2) = (\boldsymbol{x}^*, \boldsymbol{y}^*)$. Nowhere in the proof the optimality of $(\boldsymbol{x}^*, \boldsymbol{y}^*)$ was used. The same proof in thus applicable to any feasible solution to P1 that admit at most one partial supplier. Since each element of $\tilde{S}^1$ contains at most one partial node by definition, which is selected as the maximal element of the strictly totally ordered set $(Y^1, \prec)$, where $Y^1 = \left\{j \in \{1, \dots, n\} : y_j^1 = 1\right\}$, setting $y_j^2 = y_j^1$ for all $j \in \{1, \dots, n\}$ and fixing the value of the decision variables z_j^2 according to equation (11) leads to a feasible solution $(\boldsymbol{y}^2, \boldsymbol{z}^2) \in \tilde{S}^2$ such that $\tilde{f}_{2,1}(\boldsymbol{y}^2, \boldsymbol{z}^2) = (\boldsymbol{x}^1, \boldsymbol{y}^1)$ for any solution $(\boldsymbol{x}^1, \boldsymbol{y}^1) \in \tilde{S}^1$, by the proof of Proposition (2.6).

Hence, $\tilde{f}_{2,1}$ is injective and surjective, thus bijective. □

Proof of Bijection $\tilde{f}_{3,2}$: First, let us demonstrate that $\tilde{f}_{3,2}$ is injective. Let $(\boldsymbol{y}^{3A}), (\boldsymbol{y}^{3B}) \in \tilde{S}^3$ be solutions such that $\tilde{f}_{3,2}(\boldsymbol{y}^{3A}) = (\boldsymbol{y}^{2A}, \boldsymbol{z}^{2A})$ and $\tilde{f}_{3,2}(\boldsymbol{y}^{3B}) = (\boldsymbol{y}^{2B}, \boldsymbol{z}^{2B})$.

Suppose that $(\boldsymbol{y}^{2A}, \boldsymbol{z}^{2A}) = (\boldsymbol{y}^{2B}, \boldsymbol{z}^{2B})$. This implies that $y_j^{2A} = y_j^{2B}$ for all $j \in \{1, \dots, n\}$. By definition of $\tilde{f}_{3,2}$, this means that $y_j^{3A} = y_j^{3B}$ for each supplier j. Hence, $(\boldsymbol{y}^{3A}) = (\boldsymbol{y}^{3B})$.

Therefore, $\tilde{f}_{3,2}$ is injective.

Now, let us demonstrate that $\tilde{f}_{3,2}$ is surjective. We consider a solution $(\boldsymbol{y}^2, \boldsymbol{z}^2) \in \tilde{S}^2$. Since P3 is obtained from P2 by removing the decision variables y_j as well as a subset of constraints, setting $y_j^3 = y_j^2\ \forall j \in \{1, \dots, n\}$ leads to a feasible solution $\boldsymbol{y}^3 \in S^3$. Furthermore, the fact that $\boldsymbol{y}^3$ is an element of the subset $\tilde{S}^3$ follows from constraint (7), which must be respected by the vector $\boldsymbol{y}^2$ and therefore by $\boldsymbol{y}^3$.

Let us consider solution $(\boldsymbol{y}^*, \boldsymbol{z}^*) = \tilde{f}_{3,2}(\boldsymbol{y}^3)$. The definition of $\tilde{f}_{3,2}$ directly implies that $\boldsymbol{y}^* = \boldsymbol{y}^2$. Furthermore, since supplier p^3, defined in 3.1, corresponds to the maximal element of the set of nodes that are used in solution $\boldsymbol{y}^3$ and consequently in solution $\boldsymbol{y}^*$ according to the strict total order $\prec$, then setting $z_j^* = \mathbb{1}(j = p^3)$ for all $j \in \{1, \dots, n\}$ is equivalent to setting $z_j^* = y_j^* \mathbb{1}(j \succ i\ \ \forall i \neq j : y_i^* = 1)$. Consequently, since (11) is respected by $(\boldsymbol{y}^2, \boldsymbol{z}^2)$ by definition of $\tilde{S}^2$, then $\boldsymbol{z}^* = \boldsymbol{z}^2$. Thus, $\tilde{f}_{3,2}(\boldsymbol{y}^3) = (\boldsymbol{y}^2, \boldsymbol{z}^2)$. Therefore, $\tilde{f}_{3,2}$ is also surjective. □

Proof of Bijection $\tilde{f}_{3,1}$: Function $\tilde{f}_{3,1}$ being the composition of two bijective functions, it is also bijective. □

Proof of Proposition 3.2 : We have shown that $\tilde{f}_{2,1}$ is a bijective function from the set of feasible solutions to P2 respecting constraint (11) to a subset of feasible solutions to P1. This function is the same as the one given in Proposition 2.5, but defined on the domain $\tilde{S}^2 \subseteq S^2$ instead of S^2.

Hence, Proposition 2.5 implies that $Z^2(\boldsymbol{y}^2, \boldsymbol{z}^2) = Z^1\big(\tilde{f}_{2,1}(\boldsymbol{y}^2, \boldsymbol{z}^2)\big)$ for any solution $(\boldsymbol{y}^2, \boldsymbol{z}^2) \in \tilde{S}^2$.

By Proposition 2.8, there exists a solution $(\boldsymbol{y}^*, \boldsymbol{z}^*) \in \tilde{S}^2$ such that $Z^2(\boldsymbol{y}^*, \boldsymbol{z}^*) = Z^2$. Thus, $Z^1\big(\tilde{f}_{2,1}(\boldsymbol{y}^*, \boldsymbol{z}^*)\big) = Z^2 \overset{(2.6)}{=} Z^1$.

Hence, $\tilde{f}_{2,1}(\boldsymbol{y}^*, \boldsymbol{z}^*) \in \tilde{S}^1$ is an optimal solution to P1. □

Proof of Proposition 3.3 : Suppose that a solution $\boldsymbol{y}^3$ to P3(λ) includes a supplier $i \in \{1, \ldots, n\}$ such that $y_i^3 = 1$ and $\sum_{j=1}^n b_j y_j^3 - b_i \geq D$, where $\lambda < e_1$. Let us consider another solution $\boldsymbol{y}^*$ defined as follows:

$$y_j^* = \begin{cases} 0 & \text{for } j = i \\ y_j^3 & \text{for } j = 1, \ldots, i-1, i+1, \ldots, n. \end{cases}$$

It follows from the hypotheses above that $\boldsymbol{y}^*$ respects constraints (18) and (19). The objective value of this solution is given by

$$\begin{aligned}
Z^3(\lambda, \boldsymbol{y}^*) &\overset{(17)}{=} \lambda D + \sum_{j=1}^n \big((c_j - \lambda)b_j + f_j\big) y_j^* \\
&= \lambda D + \sum_{j=1}^n \big((c_j - \lambda)b_j + f_j\big) y_j^3 - \big((c_i - \lambda)b_i + f_i\big) \\
&= Z^3(\lambda, \boldsymbol{y}^3) - \big((c_i - \lambda)b_i + f_i\big) \\
&= Z^3(\lambda, \boldsymbol{y}^3) - b_i(e_i - \lambda) \\
&\leq Z^3(\lambda, \boldsymbol{y}^3) - b_i(e_1 - \lambda) && \text{, since } e_1 \leq e_j \ \forall j \\
&< Z^3(\lambda, \boldsymbol{y}^3) && \text{, since } \lambda < e_1
\end{aligned}$$

Thus, $\boldsymbol{y}^*$ is a feasible solution to P3(λ) having a lower objective value than $\boldsymbol{y}^3$. Hence, a solution to P3 that includes a supplier $i \in \{1, \ldots, n\}$ such that $y_i^3 = 1$ and $\sum_{j=1}^n b_j y_j^3 - b_i \geq D$ cannot be optimal if $\lambda < e_1$. In particular, this is true for $i = p^3$. Hence, an optimal solution to P3(λ) is necessarily an element of the subset $\tilde{S}^3$ when $\lambda < e_1$. □

Proof of Proposition 3.4 : Let us consider a solution $\boldsymbol{y}^3 \in \tilde{S}^3$. Applying the bijective function $\tilde{f}_{3,2}$ to this solution leads to the solution $(\boldsymbol{y}^2, \boldsymbol{z}^2) = \tilde{f}_{3,2}(\boldsymbol{y}^3) \in \tilde{S}^2$, with objective value

$$\begin{aligned}
Z^2(\boldsymbol{y}^2, \boldsymbol{z}^2) &\overset{(5)}{=} \sum_{j=1}^{n}(c_j b_j + f_j)y_j^2 - \Big(\sum_{j=1}^{n} b_j y_j^2 - D\Big)\sum_{j=1}^{n} c_j z_j^2 \\
&= \sum_{j=1}^{n}(c_j b_j + f_j)y_j^3 - \Big(\sum_{j=1}^{n} b_j y_j^3 - D\Big)\sum_{j=1}^{n} c_j z_j^2 && \text{, by defn. of } \tilde{f}_{3,2} \\
&= \sum_{j=1}^{n}(c_j b_j + f_j)y_j^3 - \Big(\sum_{j=1}^{n} b_j y_j^3 - D\Big)c_{p^3} && \text{, by defn. of } p^3 \\
&= \sum_{j=1}^{n}(c_j b_j + f_j)y_j^3 - \Big(\sum_{j=1}^{n} b_j y_j^3 - D\Big)\big(\lambda - (\lambda - c_{p^3})\big) \\
&\overset{(16)}{=} Z^3\big(\lambda, (\boldsymbol{y}^3)\big) + \Big(\sum_{j=1}^{n} b_j y_j^3 - D\Big)(\lambda - c_{p^3})
\end{aligned}$$

Furthermore, as explained in the proof of Proposition 3.2, $Z^1\big(\tilde{f}_{2,1}(\boldsymbol{y}^2, \boldsymbol{z}^2)\big) = Z^2(\boldsymbol{y}^2, \boldsymbol{z}^2)$ for any solution $(\boldsymbol{y}^2, \boldsymbol{z}^2) \in \tilde{S}^2$. Hence,

$$Z^1\big(\tilde{f}_{3,1}(\boldsymbol{y}^3)\big) = Z^1\Big(\tilde{f}_{2,1}\big(\tilde{f}_{3,2}(\boldsymbol{y}^3)\big)\Big) = Z^1\big(\tilde{f}_{2,1}(\boldsymbol{y}^2, \boldsymbol{z}^2)\big) = Z^3(\lambda, \boldsymbol{y}^3) + \Big(\sum_{j=1}^{n} b_j y_j^3 - D\Big)(\lambda - c_{p^3}).$$

□

Proof of Proposition 3.5 : Let $(\boldsymbol{y}^2, \boldsymbol{z}^2)$ be a feasible solution to P2 and let us define $\mu = \max_{j\in\{1,\dots,n\}}\{c_j z_j^2\}$. Since $(\boldsymbol{y}^2, \boldsymbol{z}^2) \in S^2$, the vector $\boldsymbol{y}^3 \in \{0,1\}^n$ respecting $y_j^3 = y_j^2 \ \forall\ j \in \{1, \dots, n\}$ is a feasible solution to P3. Using a multiplier $\lambda \geq \mu$, the objective value of $\boldsymbol{y}^3$ is then given by

$$\begin{aligned}
Z^3(\lambda, \boldsymbol{y}^3) &\overset{(16)}{=} \sum_{j=1}^{n}(c_j b_j + f_j)y_j^3 - \Big(\sum_{j=1}^{n} b_j y_j^3 - D\Big)\lambda \\
&= \sum_{j=1}^{n}(c_j b_j + f_j)y_j^2 - \Big(\sum_{j=1}^{n} b_j y_j^2 - D\Big)\lambda \\
&\leq \sum_{j=1}^{n}(c_j b_j + f_j)y_j^2 - \Big(\sum_{j=1}^{n} b_j y_j^2 - D\Big)\mu && \text{, by constraint (6) and since } \lambda \geq \mu \\
&\leq \sum_{j=1}^{n}(c_j b_j + f_j)y_j^2 - \Big(\sum_{j=1}^{n} b_j y_j^2 - D\Big)\sum_{j=1}^{n} c_j z_j^2 && \text{, by (8) and the defn. of } \mu \\
&\overset{(5)}{=} Z^2(\boldsymbol{y}^2, \boldsymbol{z}^2)
\end{aligned}$$

From there, we can demonstrate the second part of the proposition. Let us consider a solution $(\boldsymbol{x}^1, \boldsymbol{y}^1) \in \tilde{S}^1$. The bijective function $\tilde{f}_{3,1}$ guarantees the existence of a solution $\boldsymbol{y}^3 \in \tilde{S}^3$ such

that $\tilde{f}_{3,1}(\boldsymbol{y}^3) = (\boldsymbol{x}^1, \boldsymbol{y}^1)$. By Proposition 3.4, the equation

$$Z^1(\boldsymbol{x}^1, \boldsymbol{y}^1) = Z^3(\lambda, \boldsymbol{y}^3) + \Big(\sum_{j=1}^{n} b_j y_j^3 - D\Big)(\lambda - c_{p^3}) \tag{50}$$

holds for any multiplier $\lambda \geq 0$. If $(\boldsymbol{x}^1, \boldsymbol{y}^1)$ does not contain a partial node, then constraint (18) holds with equality and the second term on the right-hand side is equal to zero. Hence, $Z^3(\lambda, \boldsymbol{y}^3) = Z^1(\boldsymbol{x}^1, \boldsymbol{y}^1)$.

If $(\boldsymbol{x}^1, \boldsymbol{y}^1)$ contains a partial node, it corresponds to p^3 by construction of $\tilde{S}^1$. Hence, the fact that $(\boldsymbol{x}^1, \boldsymbol{y}^1)$ contains a partial node with a variable cost inferior or equal to λ is equivalent to the inequality $\lambda \geq c_{p^3}$. Furthermore, the demand constraint (18) implies that $(\sum_{j=1}^{n} b_j y_j^3 - D) \geq 0$. Thus (50) lead to,

$$Z^3(\lambda, \boldsymbol{y}^3) = Z^1(\boldsymbol{x}^1, \boldsymbol{y}^1) - \Big(\sum_{j=1}^{n} b_j y_j^3 - D\Big)(\lambda - c_{p^3}) \leq Z^1(\boldsymbol{x}^1, \boldsymbol{y}^1)$$

In both cases, feasible solutions $\boldsymbol{y}^3 \in \tilde{S}^3$ to P3 such that $Z^3(\lambda, \boldsymbol{y}^3) \leq Z^1(\boldsymbol{x}^1, \boldsymbol{y}^1)$ can therefore be found. □

Proof of Proposition 3.6 : $\boldsymbol{y}^3$ is a feasible solution to P3, with objective value $Z^3(\lambda_1)$. Hence, replacing λ_1 by λ_2 in the objective function (16) does not affect the feasibility of $\boldsymbol{y}^3$ and leads to the objective value given in the proposition, i.e.

$$\begin{aligned} Z^3(\lambda_2, \boldsymbol{y}^3) &= \sum_{j=1}^{n} (c_j b_j + f_j) y_j^3 - \Big(\sum_{j=1}^{n} b_j y_j^3 - D\Big)\lambda_2 \\ &= \sum_{j=1}^{n} (c_j b_j + f_j) y_j^3 - \Big(\sum_{j=1}^{n} b_j y_j^3 - D\Big)\big(\lambda_1 + (\lambda_2 - \lambda_1)\big) \\ &\overset{(16)}{=} Z^3(\lambda_1) - \Big(\sum_{j=1}^{n} b_j y_j^3 - D\Big)(\lambda_2 - \lambda_1) \end{aligned}$$

□

Proof of Proposition 3.7 : Let $g(\lambda)$ be the equation of the line passing by two points $\big(\lambda_0, Z^3(\lambda_0)\big)$ and $\big(\lambda_1, Z^3(\lambda_1)\big)$ in the plane. This line is given by $g(\lambda) = Z^3(\lambda_0) - (\lambda - \lambda_0)\dfrac{Z^3(\lambda_0) - Z^3(\lambda_1)}{\lambda_1 - \lambda_0}$. We want to show that $g(\lambda)$ is a lower bound on $Z^3(\lambda)\ \forall\ \lambda \in [\lambda_0, \lambda_1]$.

Since $g(\lambda_0) = Z^3(\lambda_0)$ and $g(\lambda_1) = Z^3(\lambda_1)$, the result is trivial at the limits of the interval $[\lambda_0, \lambda_1]$. Now, let us consider the case where $\lambda_0 < \lambda < \lambda_1$. Let $\boldsymbol{y}^\lambda$ be an optimal solution to P3(λ), with objective value $Z^3(\lambda)$.

Since $Z^3(\lambda_0)$ is the optimal value of P3(λ_0), the upper bound on $Z^3(\lambda_0)$ that can be obtained by Proposition 3.6 using $\boldsymbol{y}^\lambda$ is necessarily greater than or equal to $Z^3(\lambda_0)$. Furthermore,

constraint (18) implies that $(\sum_{j=1}^{n} b_j y_j^{\lambda} - D) \geq 0$. Hence, we have

$$Z^3(\lambda_0) \leq Z^3(\lambda) - \Big(\sum_{j=1}^{n} b_j y_j^{\lambda} - D\Big)(\lambda_0 - \lambda) \tag{51}$$

$$\iff \qquad \frac{Z^3(\lambda_0) - Z^3(\lambda)}{(\lambda - \lambda_0)} \leq \Big(\sum_{j=1}^{n} b_j y_j^{\lambda} - D\Big) \tag{52}$$

Similarly, since $Z^3(\lambda_1)$ is the optimal value of P3(λ_1), the upper bound on this value that is given by Proposition 3.6 must be greater than or equal to $Z^3(\lambda_1)$. Formally,

$$Z^3(\lambda_1) \leq Z^3(\lambda) - \Big(\sum_{j=1}^{n} b_j y_j^{\lambda} - D\Big)(\lambda_1 - \lambda) \tag{53}$$

$$\iff \qquad \frac{Z^3(\lambda) - Z^3(\lambda_1)}{(\lambda_1 - \lambda)} \geq \Big(\sum_{j=1}^{n} b_j y_j^{\lambda} - D\Big) \tag{54}$$

Inequalities (52) and (54) directly lead to the expected lower bound on $Z^3(\lambda)$.

$$\begin{aligned}
&\frac{Z^3(\lambda_0) - Z^3(\lambda)}{(\lambda - \lambda_0)} \leq \frac{Z^3(\lambda) - Z^3(\lambda_1)}{(\lambda_1 - \lambda)} \\
\iff \qquad & Z^3(\lambda) \geq \Big(\frac{\lambda - \lambda_1}{\lambda_0 - \lambda_1}\Big) Z^3(\lambda_0) - \Big(\frac{\lambda - \lambda_0}{\lambda_0 - \lambda_1}\Big) Z^3(\lambda_1) \\
\iff \qquad & Z^3(\lambda) \geq Z^3(\lambda_0) - (\lambda - \lambda_0)\frac{Z^3(\lambda_0) - Z^3(\lambda_1)}{\lambda_1 - \lambda_0} \\
\iff \qquad & Z^3(\lambda) \geq g(\lambda)
\end{aligned}$$

□

Proof of Proposition 3.9 : Proposition 3.2 states that there exists an optimal solution $(\boldsymbol{x}^*, \boldsymbol{y}^*)$ to P1 such that $(\boldsymbol{x}^*, \boldsymbol{y}^*) \in \tilde{S}^1$. Now, suppose that a solution $(\boldsymbol{x}^1, \boldsymbol{y}^1) \in \tilde{S}^1$ contains a partial node p supplying x_p^1 units. We will show that if the bound $LB_{x_p}^{\mathrm{Dom}}(q)$ associated with a node $q \succ p$ respecting $f_q \leq f_p$ is greater than or equal to x_p^1, then another solution $(\boldsymbol{x}', \boldsymbol{y}') \in \tilde{S}^1$ such that $Z^1(\boldsymbol{x}', \boldsymbol{y}') \leq Z^1(\boldsymbol{x}^1, \boldsymbol{y}^1)$ in which p is not used exists.

Since $q \succ p$, the fact that solution $(\boldsymbol{x}^1, \boldsymbol{y}^1) \in \tilde{S}^1$ partially uses node p implies that $x_q^1 = y_q^1 = 0$. From there, the modified solution can be built as follows:

$$x'_j = \begin{cases} x_j^1 & \forall j \in \{1, \ldots, n\} : j \neq p \text{ and } j \neq q \\ 0 & \text{for } j = p, \\ x_p^1 & \text{for } j = q. \end{cases}$$

$$y'_j = \begin{cases} y_j^1 & \forall j \in \{1, \ldots, n\} : j \neq p \text{ and } j \neq q \\ 0 & \text{for } j = p, \\ 1 & \text{for } j = q. \end{cases}$$

This modification only implies to use x_p^1 units on node q instead of node p, which becomes unused. Since $b_q \geq LB_{x_p}^{\text{Dom}}(q)$ by definition of the dominance relation, and since $LB_{x_p}^{\text{Dom}}(q) \geq x_p^1$ by hypothesis, $(\boldsymbol{x}', \boldsymbol{y}')$ remains a feasible solution to P2. Also, the fact that $(\boldsymbol{x}^1, \boldsymbol{y}^1)$ is an element of $\tilde{S}^1$ implies that p was the only partial node used in this solution, and that it was the maximal element among the used nodes according to the strict total order $\prec$. Now, q is the only node that can be partial in $(\boldsymbol{x}', \boldsymbol{y}')$ by construction and the other used nodes remain the same. Hence, since $q \succ p$ by hypothesis, $(\boldsymbol{x}', \boldsymbol{y}')$ respects the definition of $\tilde{S}^1$.

From there, it suffices to show that $Z^1(\boldsymbol{x}', \boldsymbol{y}') \leq Z^1(\boldsymbol{x}^1, \boldsymbol{y}^1)$ to conclude that $(\boldsymbol{x}^1, \boldsymbol{y}^1)$ does not have to be considered to identify an optimal solution $(\boldsymbol{x}^*, \boldsymbol{y}^*) \in \tilde{S}^1$ to P1. The changes made to $(\boldsymbol{x}^1, \boldsymbol{y}^1)$ modify the objective value $Z^1(\boldsymbol{x}^1, \boldsymbol{y}^1)$ by $f_q - f_p + x_p^1(c_q - c_p)$. Let us demonstrate that this value is nonpositive.

First, if this node q is such that $c_q = c_p$, then $f_q - f_p + x_p^1(c_q - c_p) = f_q - f_p$. Since $f_q \leq f_p$ by hypothesis, this value is nonpositive and $Z^1(\boldsymbol{x}', \boldsymbol{y}') \leq Z^1(\boldsymbol{x}^1, \boldsymbol{y}^1)$.

In the opposite case, $c_q > c_p$, since $q \succ p$ and $c_q \neq c_p$. By definition of the dominance relation, $LB_{x_p}^{\text{Dom}}(q) \leq \frac{f_p - f_q}{c_q - c_p}$ and $x_p^1 \leq LB_{x_p}^{\text{Dom}}(q)$ by hypothesis. From there, $f_q - f_p + x_p^1(c_q - c_p) \leq f_q - f_p + \frac{f_p - f_q}{c_q - c_p}(c_q - c_p) = 0$.

Consequently, in both possible cases,

$$Z^1(\boldsymbol{x}', \boldsymbol{y}') = Z^1(\boldsymbol{x}^1, \boldsymbol{y}^1) + f_q - f_p + x_p^1(c_q - c_p) \leq Z^1(\boldsymbol{x}^1, \boldsymbol{y}^1)$$

When considering solutions to P1 that are elements of $\tilde{S}^1$ and in which a node p is partially used, the global lower bound $LB_{x_p}^{\text{Dom}}$ therefore corresponds to the smallest integer number of shipments x_p for which the previous arguments do not prove the existence of a different solution $(\boldsymbol{x}', \boldsymbol{y}') \in \tilde{S}^1$ having an objective value at least as good as $Z^1(\boldsymbol{x}^1, \boldsymbol{y}^1)$ in which p is not used. □

Proof of Proposition 6.1 : Since the constraints of both problems $\hat{\text{P}}1$ and P1 are identical, any feasible solution to $\hat{\text{P}}1$ is also a feasible solution to P1 and vice versa.

Hence, suppose that $(\hat{\boldsymbol{x}}^*, \hat{\boldsymbol{y}}^*)$ is an optimal solution to the modified problem $\hat{\text{P}}1$. Its objective value is given by

$$\begin{aligned}
\hat{Z}^1 &= \sum_{j=1}^{n} (\hat{c}_j \hat{x}_j^* + f_j \hat{y}_j^*) \\
&= \sum_{j=1}^{n} \left((c_j - \theta) \hat{x}_j^* + f_j \hat{y}_j^* \right) && \text{, by defn. of the unit costs } \hat{c}_j \\
&= \sum_{j=1}^{n} (c_j \hat{x}_j^* + f_j \hat{y}_j^*) - \theta \sum_{j=1}^{n} \hat{x}_j^* \\
&= \sum_{j=1}^{n} (c_j \hat{x}_j^* + f_j \hat{y}_j^*) - \theta D && \text{, since } (\hat{\boldsymbol{x}}^*, \hat{\boldsymbol{y}}^*) \text{ respects constraint (2)}
\end{aligned}$$

For the original problem P1, the objective value of $(\hat{\boldsymbol{x}}^*, \hat{\boldsymbol{y}}^*)$ is

$$\begin{aligned} Z^1(\hat{\boldsymbol{x}}^*, \hat{\boldsymbol{y}}^*) &= \sum_{j=1}^{n} (c_j \hat{x}_j^* + f_j \hat{y}_j^*) \\ &= \hat{Z}^1 + \theta D \end{aligned}$$

$Z^1(\hat{\boldsymbol{x}}^*, \hat{\boldsymbol{y}}^*)$ being the objective value of a feasible solution to P1, it is a lower bound on the optimal value Z^1 to P1. Thus, $Z^1 \leq \hat{Z}^1 + \theta D$.

Now, let $(\boldsymbol{x}^*, \boldsymbol{y}^*)$ be an optimal solution to the original problem P1. Its objective value is given by

$$\begin{aligned} Z^1 &= \sum_{j=1}^{n} (c_j x_j^* + f_j y_j^*) \\ &= \sum_{j=1}^{n} \left((\hat{c}_j + \theta) x_j^* + f_j y_j^* \right) && \text{, by defn. of the unit costs } \hat{c}_j \\ &= \sum_{j=1}^{n} (\hat{c}_j x_j^* + f_j y_j^*) + \theta \sum_{j=1}^{n} x_j^* \\ &= \sum_{j=1}^{n} (\hat{c}_j x_j^* + f_j y_j^*) + \theta D && \text{, since } (\boldsymbol{x}^*, \boldsymbol{y}^*) \text{ respects constraint (2)} \end{aligned}$$

For the modified problem $\hat{\text{P}}1$, the objective value of $(\boldsymbol{x}^*, \boldsymbol{y}^*)$ is

$$\begin{aligned} \hat{Z}^1(\boldsymbol{x}^*, \boldsymbol{y}^*) &= \sum_{j=1}^{n} (\hat{c}_j x_j^* + f_j y_j^*) \\ &= Z^1 - \theta D \end{aligned}$$

$\hat{Z}^1(\boldsymbol{x}^*, \boldsymbol{y}^*)$ being the objective value of a feasible solution to $\hat{\text{P}}1$, it is a lower bound on the optimal value $\hat{Z}^1$ to $\hat{\text{P}}1$. Thus, $\hat{Z}^1 \leq Z^1 - \theta D$.

We have shown that $Z^1 \leq \hat{Z}^1 + D\theta$ and that $\hat{Z}^1 \leq Z^1 - D\theta$. Therefore, we conclude that $\hat{Z}^1 = Z^1 - D\theta$. □

Proof of Proposition 6.2 : Let $(\boldsymbol{x}, \boldsymbol{y})$ be a feasible solution to P1 such that $y_j = 0$ for a node j having a negative fixed cost $f_j < 0$. Let $(\tilde{\boldsymbol{x}}, \tilde{\boldsymbol{y}})$ be the same solution, except that $\tilde{y}_j = 1$ for the aforementioned node j. This modification does not affect the feasibility of this solution and leads to the objective value $Z^1(\tilde{\boldsymbol{x}}, \tilde{\boldsymbol{y}}) = Z^1(\boldsymbol{x}, \boldsymbol{y}) + f_j < Z^1(\boldsymbol{x}, \boldsymbol{y})$. Hence, for a solution $(\boldsymbol{x}^1, \boldsymbol{y}^1)$ to P1 to be optimal, it is necessary that $\boldsymbol{y}^1 = 1$ for each node $j \in \{1, \ldots, n\}$ such that $f_j < 0$. Consequently, this reduction does not prevent the identification of an optimal solution and therefore does not affect the optimal value. □

Appendix 2

Explanation of the time complexity of steps H4 and F : Step H4 requires to solve model (29)-(32) for each partial node $p \in P$. If $Z^{\text{LP}}_{\text{LB}(p)} < Z_{\text{UB}}$, the strong linear relaxation bounds $LB^{\text{LP}}_{x_p}$ and $UB^{\text{LP}}_{x_p}$ on the usage of the partial node p are then computed.

One could use a naive implementation in which this process is executed independently for each partial node $p \in P$. Each time, solving the linear program (29)-(32) optimally would be done in $\mathcal{O}(n)$. Using the optimal solution to the strong linear relaxation, $LB^{\text{LP}}_{x_p}$ and $UB^{\text{LP}}_{x_p}$ could then be computed in $\mathcal{O}(n)$. Indeed, starting from the optimal solution to (29)-(32), $LB^{\text{LP}}_{x_p}$ can be calculated by sequentially going through nodes $j \in \{s^p, s^p + 1, \dots\}$, verifying if they respect constraint (32) and, if it is the case, computing how many units can be used on j instead of p without the value of the current solution to (29)-(32) to exceed the incumbent value Z_{UB}. When x_p reaches LB_{x_p}, the cost of the current solution to (29)-(32) reaches Z_{UB} or node n is reached, the computation stops. Computing $UB^{\text{LP}}_{x_p}$ can be done very similarly. The only difference is that, instead of visiting the unused nodes $\{s^p, s^p + 1, \dots\}$ to decrease the usage of node p, the complete nodes $\{s^p, s^p - 1, \dots\}$ are crossed sequentially and x_p is increased. Using this approach, the complexity of these computations would therefore be linear in the number of nodes n and would be repeated for each partial node $p \in P$. The total complexity of step H4 would then be $\mathcal{O}(|P_{\text{H4}}| \cdot n)$.

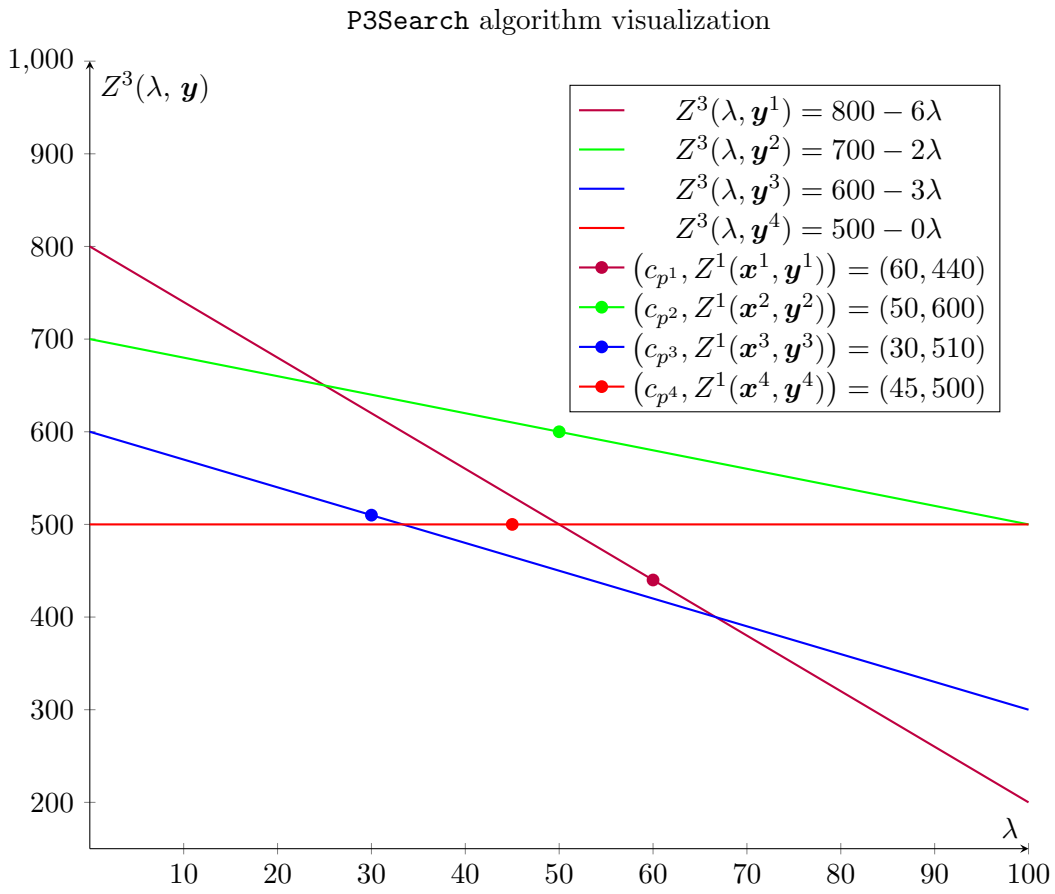
To improve this theoretical bound, we use a more clever implementation that relies on appropriate data structures. First, we build a max-heap containing pointers to nodes $j \in \{1, \dots, s\}$, namely the nodes that are included in the optimal solution to the classical linear relaxation (12)-(14). In the resulting complete binary tree, the children of an internal node that contains a pointer to supplier j point to suppliers i such that $i \prec j$. Building this heap is done in $\mathcal{O}\big(n \log(n)\big)$. Then, we build a linked list containing pointers to the n nodes of the problem. The nodes in the linked list respect the ordering used in the article i.e. $e_1 \leq e_2 \leq \dots \leq e_n$. This step is performed in $\mathcal{O}(n)$, since we suppose that the nodes were previously sorted in non-decreasing order of linearized cost. Finally, in $\mathcal{O}(n)$, we build an array of size $|P_{\text{H4}}|$ containing pointers to each potential partial node $p \in P$. The elements of this array are then sorted according to the strict total order $\prec$, in $\mathcal{O}\big(|P_{\text{H4}}| \cdot \log(|P_{\text{H4}}|)\big)$.

After initializing these data structures, $Z^{\text{LP}}_{\text{LB}(p)}$ and, if necessary, $LB^{\text{LP}}_{x_p}$ and $UB^{\text{LP}}_{x_p}$, are computed for each $p \in P$, starting by the maximal element of the strictly totally ordered set $(P, \prec)$ and iterating backwards through the previously built array. For each new partial node p, we start by removing from the heap each node that can no longer be used in a solution that is an element of $\tilde{S}^1$. To do so, it suffices to delete the root node j as long as $j \succ p$. Indeed, since $p \succ q$ for each remaining node $q \in P$, any root $j \succ p$ is such that $j \succ q$ and will thus not be used in the optimal solution to the remaining problems (29)-(32), by constraint (32). After removing these nodes from the current solution, we use our linked list to add the nodes j of minimal linearized cost e_j that follow the previous split node and that respect $j \prec p$ to the solution until the demand is satisfied. When iterating over a node j such that $j \succ p$, we delete it from the linked list in $\mathcal{O}(1)$. Each node that is added to the solution is inserted into the heap. At this point, the current solution $\boldsymbol{x}$ corresponds to an optimal solution to the problem that would be obtained by relaxing constraint (31) from model (29)-(32). To respect constraint (31) and obtain the optimal solution that is defined is Section 3.4.1, the number of units that is used

on node p must be adjusted. This is done by trading units between node p and other nodes, over which we iterate by starting on the current split node and moving to its neighbours in the linked list. At most b_p units have to be moved between p and other suppliers. Hence, in addition to p, no more than $\min\{n, \frac{b_p}{\min\{b_j\}}\} \in \mathcal{O}\big(\min\{n, \frac{\max\{b_j\}}{\min\{b_j\}}\}\big)$ nodes will be involved in these movements of units. If $Z^{\mathrm{LP}}_{\mathrm{LB}(p)} < Z_{\mathrm{UB}}$, computing $LB^{\mathrm{LP}}_{x_p}$ and $UB^{\mathrm{LP}}_{x_p}$ requires similar adjustment operations, with the same complexity. In total, the complexity of step H4 is therefore bounded by $\mathcal{O}\big(n\log(n) + |P_{\mathrm{H4}}| \cdot \min\{n, \frac{\max\{b_j\}}{\min\{b_j\}}\}\big)$.

The complexity of step F can be justified similarly. First, the linear relaxation (36)-(38) is solved in $\mathcal{O}(n)$. Then, the strict total order filtering requires to verify whether $Z^{\mathrm{LP}(q)} + b_j(e_{s^q} - e_j)$ exceeds Z_{UB} for each node $j \in Q$ such that $j < s^q$ and, symmetrically, to compare $Z^{\mathrm{LP}(q)} + b_j(e_j - e_{s^q})$ with the incumbent value for each node $j \in Q$ such that $j > s^q$. This step also requires $\mathcal{O}(n)$ operations in total. For each node j that could not be fixed by the strict total order filtering, the strong linear filtering is applied. Computing k^0_j (or k^1_j if $j > s^q$) requires to move at most b_j units from node j to nodes $i \in \{s^q, s^q+1, \ldots, t^j\} \cap Q$ (or from nodes $i \in \{s^q, s^q-1, \ldots, r^j\} \cap Q$ to node j if $j > s^q$). Hence, in each case, at most $\min\{n, \frac{\max\{b_j\}}{\min\{b_j\}}\}$ nodes need to be visited. The time complexity of step F is thus bounded by $\mathcal{O}\big(n \cdot \min\{n, \frac{\max\{b_j\}}{\min\{b_j\}}\}\big)$.

Appendix 3



Suppose that $\boldsymbol{y}^1, \boldsymbol{y}^2, \boldsymbol{y}^3$ and $\boldsymbol{y}^4$ are four feasible solutions to P3. For each of these solutions $\boldsymbol{y}^i$, by (16), the objective value respects $Z^3(\lambda, \boldsymbol{y}^i) = \sum_{j=1}^{n}(c_j b_j + f_j)y_j^i - (\sum_{j=1}^{n} b_j y_j^i - D)\lambda$. Let assume that $\boldsymbol{y}^i \in \tilde{S}^3$ for each $i \in \{1, 2, 3, 4\}$, so that $\tilde{f}_{3,1}$ leads to a feasible solution $(\boldsymbol{x}^i, \boldsymbol{y}^i) \in \tilde{S}^1$, with objective value $Z^1(\boldsymbol{x}^i, \boldsymbol{y}^i) = Z^3(\lambda, \boldsymbol{y}^i) + (\sum_{j=1}^{n} b_j y_j^i - D)(\lambda - c_{p^i})$ in each case if $\boldsymbol{y}^i$ is discovered.

In the visualization above, each line represents a feasible solution $\boldsymbol{y}^i$ to P3. The y-intercept of the line associated with solution $\boldsymbol{y}^i$ corresponds to the sum of the total costs $\sum_{j=1}^{n}(c_j b_j + f_j)y_j^i$ of the nodes that are used in it. The negative slope $D - \sum_{j=1}^{n} b_j y_j^i$ represents the excess supply associated with $\boldsymbol{y}^i$. When solving P3(λ), $\boldsymbol{y}^\lambda$ is selected as the solution having the minimal value at the λ abscissa-position. For each solution $\boldsymbol{y}^i$, the point $\left(c_{p^i}, Z^1\left(\boldsymbol{x}^i, \boldsymbol{y}^i\right)\right)$ gives the unit cost of the maximal element p^i, as defined in 3.1, and the objective value of the corresponding feasible solution to P1 given by $\tilde{f}_{3,1}(\boldsymbol{y}^i)$.

Suppose that $\min\left\{e_s,\ \max_{j\in\{1,\ldots,n\}}(c_j)\right\} = 80$. `P3Search` will be executed as follows:

$$L \leftarrow \{\}$$
$$\lambda \leftarrow 80$$

$$\boldsymbol{y}^{\lambda} \leftarrow \boldsymbol{y}^{1}$$
$$L \leftarrow \{80\}$$
$$Z_{\text{UB}}^{\text{P3(80)}} \leftarrow 440$$
$$Z_{\text{UB}}^{\text{P3}} \leftarrow 440$$
$$\lambda \leftarrow 60$$

$$\boldsymbol{y}^{\lambda} \leftarrow \boldsymbol{y}^{3}$$
$$L \leftarrow \{80, 60\}$$
$$Z_{\text{UB}}^{\text{P3(60)}} \leftarrow 510 > Z_{\text{UB}}^{\text{P3}}$$
$$\lambda \leftarrow 30$$

$$\boldsymbol{y}^{\lambda} \leftarrow \boldsymbol{y}^{4}$$
$$L \leftarrow \{80, 60, 30\}$$
$$Z_{\text{UB}}^{\text{P3(45)}} \leftarrow 500 > Z_{\text{UB}}^{\text{P3}}$$
$$\lambda \leftarrow 45$$
$$\sum_{j=1}^{n} b_j y_j^{\lambda} = D$$

At the end of the execution, the solutions $\boldsymbol{y}^1$, $\boldsymbol{y}^3$ and $\boldsymbol{y}^4$ have been discovered. The set of multipliers L for which the optimal value $Z^3(\lambda)$ is known is $\{80, 60, 30\}$ and the incumbent solution to P1 is $(\boldsymbol{x}^1, \boldsymbol{y}^1)$, with an objective value of $Z^1(\boldsymbol{x}^1, \boldsymbol{y}^1) = 440$.

This example also illustrates the influence of the initial multiplier λ on the quality of the bounds that can be obtained through `P3Search`. For example, if λ was initially set to 50 instead of 80, solutions $\boldsymbol{y}^3$ and $\boldsymbol{y}^4$ would be the only ones to be discovered and the global upper bound $Z_{\text{UB}}^{\text{P3}}$ would therefore be equal to 500 instead of 440.