CIRRELT
1971 – 2021

CIRRELT-2021-01

Node-Based Lagrangian Relaxations for Multicommodity Capacitated Fixed-Charge Network Design

Mohammad Rahim Akhavan Kazemzadeh
Tolga Bektaş
Teodor Gabriel Crainic
Antonio Frangioni
Bernard Gendron
Enrico Gorgone

January 2021

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

Node-Based Lagrangian Relaxations for Multicommodity Capacitated Fixed-Charge Network Design[ł]

Mohammad Rahim Akhavan Kazemzadeh[1,2], Tolga Bektaş[3], Teodor Gabriel Crainic[1,4,*], Antonio Frangioni[5], Bernard Gendron[1,2], Enrico Gorgone[6]

[1] Interuniversity Research Centre on Enterprise Networks, Logistics and Transportation (CIRRELT)

[2] Department of Computer Science and Operations Research, Université de Montréal, P.O. Box 6128, Station Centre-Ville, Montréal, Canada H3C 3J7

[3] University of Liverpool Management School, University of Liverpool, Liverpool, L69 7ZH, United Kingdom

[4] Department of Management and Technology, Université du Québec à Montréal, P.O. Box 8888, Station Centre-Ville, Montréal, Canada H3C 3P8

[5] Dipartimento di informatica, Università di Pisa

[6] Dipartimento di elettronica, informatica e sistemistica, Univerità della Calabria

Abstract. Classical Lagrangian relaxations for the multicommodity capacitated fixed-charge network design problem are the so-called flow and knapsack relaxations, where the resulting Lagrangian subproblems decompose by commodities and by arcs, respectively. We introduce node-based Lagrangian relaxations, where the resulting Lagrangian subproblem decomposes by nodes. We show that the Lagrangian dual bounds of these relaxations improve upon the linear programming relaxation bound, known to be equal to the Lagrangian dual bounds for the flow and knapsack relaxations. We also develop a Lagrangian matheuristic to compute upper bounds. The computational results on a set of benchmark instances show that the Lagrangian matheuristic is competitive with the state-of-the-art heuristics from the literature.

Keywords. Networks design, Lagrangian relaxation, column generation, matheuristic.

Acknowledgement: Partial funding for this work was provided by the Canadian Natural Sciences and Engineering Research Council (NSERC) through its Discovery Grants Program. The BTT code used in this paper has been partly developed under the auspices of the European Union's Horizon 2020 research and innovation program under grant agreement No 773897 *plan4res*. While working on this paper, Teodor Gabriel Crainic was Adjunct Professor, Department of Computer Science and Operations Research, Université de Montréal, Canada. The authors are grateful to Serge Bisaillon for his invaluable assistance with algorithm coding and numerical experiments.

[ł] Revised version of the CIRRELT-2019-21

---

* Corresponding author: TeodorGabriel.Crainic@cirrelt.ca

Dépôt légal – Bibliothèque et Archives nationales du Québec
Bibliothèque et Archives Canada, 2021

1 Introduction

In the *multicommodity capacitated fixed-charge network design problem (MCND)*, a subset of the arcs must be selected in order to route several commodities, each with a given origin-destination pair. The selection of an arc incurs a fixed cost that provides a predefined capacity available to all commodities. In addition, a variable cost is imposed to route each unit of commodity demand on each arc. The problem is to find a minimum cost solution (including decisions about design and routing) such that the demand for each commodity is met and the capacity on each arc is not exceeded.

The problem is typically formulated as a mixed-integer programming (MIP) model that includes two main sets of constraints: *flow conservation equations*, ensuring that the demand for each commodity is satisfied, and *capacity constraints*, ensuring that the flow of all commodities on each arc does not exceed its capacity. The usual Lagrangian relaxations for this formulation are the so-called flow (or shortest path) and knapsack relaxations, which are obtained, respectively, by relaxing capacity constraints and flow conservation equations (see Section 2 for an account of the relevant literature). For the flow relaxation, the resulting Lagrangian subproblem decomposes into a series of shortest path problems, one for each commodity. The Lagrangian subproblem for the knapsack relaxation is separable by arcs, reducing to a continuous knapsack problem for each arc. The flow and knapsack relaxations can therefore be seen, respectively, as *commodity-based* and *arc-based* decomposition approaches.

In this paper, we introduce three *node-based Lagrangian relaxations*, where the subproblem decomposes by nodes. We show that the Lagrangian dual bound of each of these relaxations improves upon the linear programming (LP) relaxation bound. As a result, the node-based relaxation bounds also improve upon the Lagrangian dual bounds of the flow and knapsack relaxations, since the latter are both equal to the LP relaxation bound. We compare theoretically the three node-based relaxations, establishing a hierarchy of lower bounds between them. We also compare them experimentally by solving the *Dantzig-Wolfe (DW) reformulation* of their Lagrangian duals by column generation. Our computational results on a set of benchmark instances show that the improvement provided by the strongest node-based relaxation over the LP relaxation is 1.8% on average, with a maximum of 20.5%.

We embed the proposed node-based relaxations within a *Lagrangian matheuristic* framework, where feasible solutions to the MCND are derived from the information obtained when solving the Lagrangian dual, as in a classical Lagrangian heuristic. The Lagrangian matheuristic also involves solving a large number of mathematical programming

models, typically defined through intensification and diversification mechanisms similar to those found in the literature on metaheuristics. On the same set of benchmark instances, we show that the resulting Lagrangian matheuristic outperforms most heuristics proposed in the literature and is competitive with the best ones. For instance, compared with a state-of-the-art heuristic (Gendron et al., 2018), our Lagrangian matheuristic improves the upper bound for 13 out of 25 hard instances, showing an improvement of 0.34% on average, with a maximum of 3.37%. An advantage of the Lagrangian matheuristic over several heuristic methods is that it also provides an effective lower bound on the optimal value and therefore a guarantee on the quality of the feasible solution.

The paper is organized as follows. Section 2 provides the background material: the classical MIP model, along with a short review of the relevant literature, are presented. In Section 3, we present the node-based Lagrangian relaxations and theoretically compare the strength of their Lagrangian dual bounds. Section 4 describes the Lagrangian matheuristic. In Section 5, we present the computational results, including the evaluation of the node-based relaxations and Lagrangian matheuristic. In Section 6, we summarize this work and we propose future research directions.

2 Background

The MCND is defined on a directed graph $G = (N, A)$, where N is the set of nodes and A is the set of arcs. For each node $i \in N$, we define the sets of forward and backward neighbours, N_i^+ and N_i^-, respectively. Each commodity $k \in K$ corresponds to an origin-destination pair such that $d^k > 0$ units of flow must travel between the origin $O(k)$ and the destination $D(k)$. The objective function to be minimized includes a cost $c_{ij}^k \geq 0$ for routing one unit of commodity $k \in K$ through arc $(i, j) \in A$ and a fixed cost $f_{ij} \geq 0$ for using arc $(i, j) \in A$, thus providing a capacity $u_{ij} \in (0, \sum_{k \in K} d^k]$ on the arc. A classical model for the MCND introduces two sets of variables: x_{ij}^k is the flow of commodity $k \in K$ on arc $(i, j) \in A$ (*flow variables*), while y_{ij} is 1, if arc $(i, j) \in A$ is used, and 0, otherwise (*design variables*). The model is written as follows:

$$Z^{ND} = \min \sum_{(i,j) \in A} \sum_{k \in K} c_{ij}^k x_{ij}^k + \sum_{(i,j) \in A} f_{ij} y_{ij} \tag{1}$$

$$\sum_{j \in N_i^+} x_{ij}^k - \sum_{j \in N_i^-} x_{ji}^k = b_i^k, \quad \forall i \in N, \forall k \in K, \tag{2}$$

$$\sum_{k \in K} x_{ij}^k \leq u_{ij} y_{ij}, \quad \forall (i,j) \in A, \tag{3}$$

$$x_{ij}^k \leq d^k y_{ij}, \quad \forall (i,j) \in A, \forall k \in K, \tag{4}$$

$$x_{ij}^k \geq 0, \quad \forall (i,j) \in A, \forall k \in K, \tag{5}$$

$$y_{ij} \in \{0,1\}, \quad \forall (i,j) \in A. \tag{6}$$

The objective, (1), is to minimize total routing and design costs. Constraints (2) are the usual flow conservation equations ensuring that the demand for each commodity is routed from the origin to the destination, where:

$$b_i^k = \begin{cases} +d^k, & \text{if } i = O(k), \\ -d^k, & \text{if } i = D(k), \\ 0, & \text{otherwise,} \end{cases} \qquad \forall i \in N, \forall k \in K.$$

Capacity constraints (3) ensure that the sum of the flows on each arc $(i,j) \in A$ does not exceed its capacity u_{ij}. These are also known as *linking constraints* because they ensure that no flow is allowed on arc $(i,j) \in A$, unless it is used and its fixed cost is paid. Constraints (4) are *strong linking constraints*. Although they are redundant for the MIP model, adding these inequalities significantly improves the LP relaxation lower bound (Gendron and Crainic, 1994).

As mentioned in the Introduction, two Lagrangian relaxations have been used to develop solution methods for the MCND: the flow and knapsack relaxations. The flow relaxation is obtained by relaxing (3)-(4) in a Lagrangian way. The Lagrangian subproblem separates into two components, one in y variables, trivially solvable by inspection, and the other in x variables that decomposes by commodities and can be solved as a shortest path problem (with nonnegative arc lengths) for each commodity $k \in K$. The knapsack relaxation is derived by relaxing equations (2) in a Lagrangian way. The Lagrangian subproblem decomposes by arcs, reducing to the computation of a continuous knapsack problem for each arc $(i,j) \in A$. We now review the literature on the MCND with a focus on Lagrangian-based algorithms.

An early line of research looked at the theoretical and experimental comparison of the two Lagrangian relaxations (Gendron and Crainic, 1994; Gendron et al., 1999). In particular, it was shown that the Lagrangian dual bounds for both relaxations are equal to the LP relaxation bound (this result was recently generalized to a wide class of network design models (Gendron, 2018)). To optimize the Lagrangian dual, subgradient

methods were used, but their limitations in terms of numerical stability and speed of convergence were soon identified. Their comparison with bundle methods (Crainic et al., 2001), for both flow and knapsack relaxations, show the advantages of the latter, as they converge in fewer iterations and are more robust relative to the parameter values, problem specifications, and different relaxation types. Nevertheless, subgradient methods are still being used, because they are easy to implement and relatively fast, so they can serve to initialize column generation and bundle methods. For the two classical relaxations of the MCND, state-of-the-art subgradient methods have been recently implemented and compared (Frangioni et al., 2017).

Lagrangian-based branch-and-bound (B&B) algorithms have been developed, based mostly on the knapsack relaxation, which was shown to be more effective than the flow relaxation in such a framework (Sellmann et al., 2002). Both exact and heuristic variants of B&B were developed and tested, using subgradient algorithms (Holmberg and Yuan, 2000; Sellmann et al., 2002), but also bundle methods (Kliewer and Timajev, 2005). These early contributions show how to exploit Lagrangian relaxation to derive variable fixing and simple valid inequalities that preserve the structure of the Lagrangian subproblem.

Bundle methods for both the flow and the knapsack relaxations were also used as part of a slope scaling heuristic for the MCND (Crainic et al., 2004). The idea of slope scaling is to iteratively solve a linear *multicommodity minimum cost network flow problem (MMCF)* and to use the flow distribution to adjust the linear approximation at the next iteration. In that work, a bundle method was periodically launched to find new Lagrange multipliers, which were then used to reinitialize the linearized costs and start another slope scaling process. We also use a slope scaling procedure in our Lagrangian matheuristic (see Section 4.2), but it interacts with the Lagrangian dual optimization through primal solutions, rather than through Lagrange multipliers. A similar approach has been recently applied to another network design problem (Gendron and Gouveia, 2016).

While most of the contributions on the MCND essentially consider the bundle method as a black-box non-differentiable optimization algorithm, it is well-known that it is tightly linked to DW decomposition, i.e., *column generation* approaches for solving DW reformulations (Amor et al., 2009). This connection has been used to derive master problem formulations for the flow relaxation of the MCND that exploit its structure, in particular the presence of a so-called "easy component" in the Lagrangian subproblem, i.e., the component in y variables that is solvable by inspection (Frangioni and Gorgone, 2014).

In spite of the tight relationships between Lagrangian relaxation and column genera-

tion, there have been very few works on column generation and branch-and-price (B&P) for the MCND (although column generation has been used to solve MMCF subproblems in some heuristic methods for the MCND (Katayama, 2015; Katayama et al., 2009; Yaghini et al., 2013)). Apart from the paper just cited on column generation based on the flow relaxation (Frangioni and Gorgone, 2014), a B&P(-and-cut) algorithm was developed and interpreted in terms of a Lagrangian relaxation of constraints (2) and (3) (Gendron and Larose, 2014), which can be seen as a variant of the knapsack relaxation. It is interesting to note that the master problem used in this B&P algorithm is the MIP model (1)-(6), the generated columns corresponding to the multicommodity flow variables.

Besides the Lagrangian-based solution methods, other exact and heuristic algorithms for the MCND have been proposed in the literature. Exact solution methods include Benders decomposition (Costa et al., 2009, 2012) and branch-and-cut (Chouman et al., 2016, 2017), while heuristics include tabu search (Crainic and Gendreau, 2002; Crainic et al., 2000, 2006; Ghamlouche et al., 2003), path relinking (Ghamlouche et al., 2004), scatter search (Crainic and Gendreau, 2007), simulated annealing (Paraskevopoulos et al., 2016; Yaghini et al., 2013), capacity scaling (Katayama, 2015; Katayama et al., 2009), local branching (Rodríguez-Martín and José Salazar-González, 2010), and matheuristics (Chouman and Crainic, 2010; Gendron et al., 2018; Hewitt et al., 2010).

In summary, the Lagrangian relaxations proposed so far in the literature for the MCND, to the best of our knowledge, are the traditional commodity-based (flow) and arc-based (knapsack) relaxations. Lagrangian-based algorithms, consequently, also use these two relaxations. A major contribution of this paper is to develop node-based Lagrangian relaxations, and to compare them theoretically and experimentally with the flow and knapsack relaxations, and also with one another.

3 Node-based Lagrangian relaxations

The model has to be slightly reformulated to derive the node-based relaxations. For this purpose, we introduce the following notation for each node $i \in N$:
$K_i^O = \{k \in K \,|\, i = O(k)\}$, the commodities for which i is the origin;
$K_i^D = \{k \in K \,|\, i = D(k)\}$, the commodities for which i is the destination;
$K_i^T = \{k \in K \,|\, i \neq O(k), D(k)\}$, the commodities for which i is a transshipment node.

We also consider the following basic properties: 1) for each commodity $k \in K$, it is well-known that the flow conservation equation at $i = D(k)$ (or at $i = O(k)$) is redundant, and 2) because the costs are nonnegative, for each arc $(i, j) \in A$, $x_{ij}^k = 0$ if $k \in K_j^O$ (or

$k \in K_i^D$). We then rewrite the flow conservation equations (2) as follows:

$$\sum_{j \in N_i^+} x_{ij}^k - \sum_{j \in N_i^-} x_{ji}^k = 0, \quad \forall i \in N, \forall k \in K_i^T, \tag{7}$$

$$\sum_{j \in N_i^+} x_{ij}^k = d^k, \quad \forall i \in N, \forall k \in K_i^O, \tag{8}$$

$$x_{ij}^k = 0, \quad \forall (i,j) \in A, \forall k \in K_j^O \cup K_i^D. \tag{9}$$

In order to derive a compact notation when comparing the different relaxations, we define the following sets: S^x is the set of flow vectors x that satisfy (7), while T^{xy} is the set of solutions that satisfy the other constraints, (3)-(6) and (8)-(9). Using this notation, we can rewrite the MIP model for the MCND in a compact form as follows:

$$Z^{ND} = \min \{cx + fy \,|\, x \in S^x, (x,y) \in T^{xy}\}.$$

In addition, for any set S, we denote by $\text{conv}(S)$ its convex hull and by $\overline{S}$ the polyhedron obtained by relaxing the integrality constraints in the definition of S. Using this notation, we can write the LP relaxation as:

$$Z^{LP} = \min \left\{cx + fy \,|\, x \in S^x, (x,y) \in \overline{T^{xy}}\right\}.$$

3.1 Facility location relaxation

We relax constraints (7) in a Lagrangian way by introducing π_i^k, $\forall i \in N, \forall k \in K_i^T$, as the Lagrange multipliers for each of these constraints. The following valid inequalities are also added to improve the relaxation:

$$\sum_{j \in N_i^+} x_{ij}^k \leq g_i^k, \quad \forall i \in N, \forall k \in K_i^T, \tag{10}$$

where $g_i^k = \min\{d^k, \sum_{j \in N_i^-} u_{ji}\}, \forall i \in N, \forall k \in K_i^T$. The resulting Lagrangian subproblem decomposes by nodes. The subproblem for each node $i \in N$ is then:

$$Z_i^{xy}(\pi) = \min \sum_{j \in N_i^+} \left(\sum_{k \in K} c_{ij}^k(\pi) x_{ij}^k + f_{ij} y_{ij} \right) \tag{11}$$

$$\sum_{j \in N_i^+} x_{ij}^k = d^k, \quad \forall k \in K_i^O, \tag{12}$$

$$\sum_{j \in N_i^+} x_{ij}^k \leq g_i^k, \quad \forall k \in K_i^T, \tag{13}$$

$$x_{ij}^k = 0, \quad \forall j \in N_i^+, \forall k \in K_i^D \cup K_j^O, \tag{14}$$

$$\sum_{k \in K} x_{ij}^k \leq u_{ij} y_{ij}, \quad \forall j \in N_i^+, \tag{15}$$

$$x_{ij}^k \leq d^k y_{ij}, \quad \forall j \in N_i^+, \forall k \in K, \tag{16}$$

$$x_{ij}^k \geq 0, \quad \forall j \in N_i^+, \forall k \in K, \tag{17}$$

$$y_{ij} \in \{0,1\}, \quad \forall j \in N_i^+, \tag{18}$$

where

$$c_{ij}^k(\pi) = \begin{cases} c_{ij}^k + \pi_i^k - \pi_j^k, & \text{if } k \in K_i^T \cap K_j^T, \\ c_{ij}^k + \pi_i^k, & \text{if } k \in K_i^T \setminus K_j^T, \\ c_{ij}^k - \pi_j^k, & \text{if } k \in K_j^T \setminus K_i^T, \\ c_{ij}^k, & \text{if } k \in K_i^O \cap K_j^D, \end{cases} \quad \forall j \in N_i^+, \forall k \in K.$$

This node-based relaxation is called the *facility location relaxation* because the resulting subproblem for each node $i \in N$ reduces to a *capacitated facility location problem* (CFLP), where $K_i^O \cup K_i^T$ and N_i^+ are the sets of customers and facilities, respectively. Indeed, the following transformations can be done to derive from (11)-(18) the classical MIP model for the CFLP:

1. Eliminate x_{ij}^k variables such that $c_{ij}^k(\pi) \geq 0$, $\forall j \in N_i^+, \forall k \in K_i^T$;
2. Strengthen constraints (16) for $k \in K_i^T$ by replacing d^k with g_i^k;
3. Add an artificial facility with a (very large) fixed cost of M and connected to each $k \in K_i^T$ by an arc with 0 variable flow cost;
4. Replace the x_{ij}^k variables by X_{ij}^k variables assuming values in the interval [0,1], i.e., $\forall j \in N_i^+$, $x_{ij}^k = d^k X_{ij}^k$, $\forall k \in K_i^O$, $x_{ij}^k = g_i^k X_{ij}^k$, $\forall k \in K_i^T$.
5. Add M to all costs associated with X variables.

The optimal solution to this CFLP instance is also optimal for (11)-(18), and the optimal value $Z_i^{xy}(\pi)$ is obtained by subtracting $M \times (|K_i^O| + |K_i^T| + \delta)$ from the optimal value of this CFLP instance, where $\delta = 1$, if the artificial facility is used, and 0, otherwise.

A lower bound on Z^{ND} is computed as follows: $Z^{FL}(\pi) = \sum_{i \in N} Z_i^{xy}(\pi)$. The best lower bound is obtained by solving the Lagrangian dual: $Z^{FL} = \max_\pi Z^{FL}(\pi)$. We denote by T_i^{xy} the set of solutions that satisfies constraints (12)-(18), for each node $i \in N$. Because the Lagrangian subproblem decomposes by nodes, we have, by Lagrangian duality theory (Geoffrion, 1974):

$$\begin{aligned} Z^{ND} &= \min \{cx + fy \,|\, x \in S^x, (x,y) \in T^{xy}\} \\ &= \min \left\{ cx + fy \,|\, x \in S^x, (x,y) \in \left(\bigtimes_{i \in N} T_i^{xy} \right) \right\} \\ &\geq \min \left\{ cx + fy \,|\, x \in S^x, (x,y) \in \left(\bigtimes_{i \in N} \text{conv}(T_i^{xy}) \right) \right\} \\ &= Z^{FL}. \end{aligned}$$

This last expression of Z^{FL} allows us to write down the DW reformulation of the Lagrangian dual associated with the facility location relaxation. To obtain this reformulation, we let Q_i be the index set of the extreme points of $\text{conv}(T_i^{xy})$ for $i \in N$, i.e., $(x(q), y(q))_{q \in Q_i}, i \in N$, are these extreme points. We also denote by $\theta(q)$ the variable representing the weight associated with the extreme point indexed by $q \in Q_i, i \in N$. The DW reformulation of the Lagrangian dual is then written as:

$$Z^{FL} = \min \sum_{i \in N} \sum_{q \in Q_i} \theta(q) \left(\sum_{j \in N_i^+} \sum_{k \in K} c_{ij}^k x_{ij}^k(q) + \sum_{j \in N_i^+} f_{ij} y_{ij}(q) \right) \tag{19}$$

$$\sum_{j \in N_i^+} \sum_{q \in Q_i} \theta(q) x_{ij}^k(q) - \sum_{j \in N_i^-} \sum_{q \in Q_j} \theta(q) x_{ji}^k(q) = 0, \quad \forall i \in N, \forall k \in K_i^T, \ (\pi_i^k) \tag{20}$$

$$\sum_{q \in Q_i} \theta(q) = 1, \quad \forall i \in N, \tag{21}$$

$$\theta(q) \geq 0, \quad \forall i \in N, q \in Q_i. \tag{22}$$

We propose to solve this large-scale LP model by column generation, the pricing problem in this case corresponding to the Lagrangian subproblem. At any column generation iteration, the solution to the Lagrangian subproblem provides a *candidate design solution* $\widetilde{y}$, corresponding to the 0-1 values of the design variables in that solution. In addition, the DW master problem solution $\overline{\theta}$ at any iteration also provides a candidate design solution (possibly fractional) defined as $\overline{y}_{ij} = \sum_{q \in Q_i} \overline{\theta}(q) y_{ij}(q), \ \forall (i,j) \in A$. These candidate design solutions are used as input to the slope scaling heuristic presented in Section 4.2.

Using Lagrangian duality theory, we can easily show the following property:

Proposition 1 *$Z^{FL} \geq Z^{LP}$ and the inequality can be strict.*

Proof. Since $\text{conv}(T_i^{xy}) \subseteq \overline{T_i^{xy}}$ for each $i \in N$, we have:

$$\begin{aligned} Z^{FL} &= \min\left\{cx + fy \,|\, x \in S^x, (x,y) \in \left(\bigtimes_{i \in N} \text{conv}(T_i^{xy})\right)\right\} \\ &\geq \min\left\{cx + fy \,|\, x \in S^x, (x,y) \in \left(\bigtimes_{i \in N} \overline{T_i^{xy}}\right)\right\} \\ &= \min\left\{cx + fy \,|\, x \in S^x, (x,y) \in \overline{T^{xy}}\right\} \\ &= Z^{LP} \end{aligned}$$

Since there are instances of the CFLP (even when capacities are arbitrarily large) for which all optimal LP relaxation solutions are fractional, the Lagrangian subproblem does not have the integrality property, i.e., the polyhedron $\overline{T^{xy}}$ is not integral, which implies that there are MCND instances where the inequality is strict. ■

3.2 Forward-backward facility location relaxation

The reformulation used to derive the facility location relaxation is obtained by replacing (2) with (7)-(9). In addition to this replacement, we use the Lagrangian decomposition technique (Guignard and Kim, 1987) by introducing copies of design and flow variables, respectively denoted z and v, defined by the following *copy constraints*:

$$z_{ij} - y_{ij} = 0, \quad \forall (i,j) \in A, \tag{23}$$
$$v_{ij}^k - x_{ij}^k = 0, \quad \forall (i,j) \in A, \forall k \in K. \tag{24}$$

The following redundant constraints are also added:

$$\sum_{j \in N_i^-} v_{ji}^k = d^k, \quad \forall i \in N, \forall k \in K_i^D, \tag{25}$$
$$v_{ji}^k = 0, \quad \forall (j,i) \in A, \forall k \in K_i^O \cup K_j^D, \tag{26}$$
$$\sum_{k \in K} v_{ji}^k \leq u_{ji} z_{ji}, \quad \forall (j,i) \in A, \tag{27}$$
$$v_{ji}^k \leq d^k z_{ji}, \quad \forall (j,i) \in A, \forall k \in K, \tag{28}$$
$$v_{ji}^k \geq 0, \quad \forall (j,i) \in A, \forall k \in K, \tag{29}$$
$$z_{ji} \in \{0,1\}, \quad \forall (j,i) \in A \tag{30}$$

Then, the copy constraints (23) and (24), as well as the flow conservation equations (7) are relaxed in a Lagrangian way. Similar to (10), the following valid inequalities are also added to improve the relaxation:

$$\sum_{j \in N_i^-} v_{ji}^k \leq h_i^k, \; \forall i \in N, \forall k \in K_i^T, \tag{31}$$

where $h_i^k = \min\{d^k, \sum_{j \in N_i^+} u_{ij}\}$, $\forall i \in N, \forall k \in K_i^T$. As before, we denote by π_i^k, $\forall i \in N, \forall k \in K_i^T$, the Lagrange multipliers associated with the flow conservation equations (7). We also denote by γ_{ij}, $\forall (i,j) \in A$, and ω_{ij}^k, $\forall (i,j) \in A$, $\forall k \in K$, the Lagrange multipliers associated with (23) and (24), respectively.

The resulting Lagrangian subproblem decomposes not only by nodes, but further, for each node, into two independent subproblems, one in variables (x, y) only, the other in variables (v, z) only. The first subproblem, in variables (x, y), can be written as follows, for each node $i \in N$:

$$Z_i^{xy}(\gamma, \omega, \pi) = \min \sum_{j \in N_i^+} \left(\sum_{k \in K} c_{ij}^k(\omega, \pi) x_{ij}^k + f_{ij}(\gamma) y_{ij} \right) \tag{32}$$

subject to (12)-(18), where $f_{ij}(\gamma) = f_{ij} - \gamma_{ij}$, $\forall j \in N_i^+$, and

$$c_{ij}^k(\omega, \pi) = \begin{cases} c_{ij}^k - \omega_{ij}^k + \pi_i^k - \pi_j^k, & \text{if } k \in K_i^T \cap K_j^T, \\ c_{ij}^k - \omega_{ij}^k + \pi_i^k, & \text{if } k \in K_i^T \setminus K_j^T, \\ c_{ij}^k - \omega_{ij}^k - \pi_j^k, & \text{if } k \in K_j^T \setminus K_i^T, \\ c_{ij}^k - \omega_{ij}^k, & \text{if } k \in K_i^O \cap K_j^D, \end{cases} \quad \forall j \in N_i^+, \forall k \in K.$$

The second subproblem, in variables (v, z), can be written as follows, for each node $i \in N$:

$$Z_i^{vz}(\gamma, \omega) = \min \sum_{j \in N_i^-} \left(\sum_{k \in K} c_{ji}^k(\omega) v_{ji}^k + f_{ji}(\gamma) z_{ji} \right) \tag{33}$$

$$\sum_{j \in N_i^-} v_{ji}^k = d^k, \quad \forall k \in K_i^D, \tag{34}$$

$$\sum_{j \in N_i^-} v_{ji}^k \leq h_i^k, \quad \forall k \in K_i^T, \tag{35}$$

$$v_{ji}^k = 0 \quad \forall j \in N_i^-, \forall k \in K_i^O \cup K_j^D, \tag{36}$$

$$\sum_{k \in K} v_{ji}^k \leq u_{ji} z_{ji}, \quad \forall j \in N_i^-, \tag{37}$$

$$v_{ji}^k \leq d^k z_{ji}, \quad \forall j \in N_i^-, \forall k \in K, \tag{38}$$

$$v_{ji}^k \geq 0, \quad \forall j \in N_i^-, \forall k \in K, \tag{39}$$

$$z_{ji} \in \{0, 1\}, \quad \forall j \in N_i^-, \tag{40}$$

where $f_{ji}(\gamma) = \gamma_{ji}$, $\forall j \in N_i^-$ and $c_{ji}^k = \omega_{ji}^k$, $\forall j \in N_i^-, \forall k \in K$. We denote by U_i^{vz} the set of solutions that satisfy constraints (34)-(40).

The two subproblems for each node $i \in N$ reduce to CFLPs. In the first subproblem, in variables (x, y), the set of customers is $K_i^O \cup K_i^T$, while the set of facilities is N_i^+ (forward neighbours set). In the second subproblem, in variables (v, z), the set of customers is $K_i^D \cup K_i^T$, while the set of facilities is N_i^- (backward neighbours set). This node-based relaxation is thus called the *forward-backward facility location relaxation.*

A lower bound on Z^{ND} is computed as follows: $Z^{FB}(\gamma, \omega, \pi) = \sum_{i \in N}(Z_i^{xy}(\gamma, \omega, \pi) + Z_i^{vz}(\gamma, \omega))$. To obtain the best possible lower bound, we have to solve the Lagrangian dual: $Z^{FB} = \max_{\gamma,\omega,\pi} Z^{FB}(\gamma, \omega, \pi)$. By Lagrangian duality theory, we have:

$$Z^{FB} = \min_{\substack{x \in S^x \\ z = y \\ v = x}} \left\{ cx + fy \,|\, (x, y) \in \left(\bigtimes_{i \in N} \text{conv}(T_i^{xy}) \right), (v, z) \in \left(\bigtimes_{i \in N} \text{conv}(U_i^{vz}) \right) \right\}.$$

We can model this Lagrangian dual with a DW reformulation. To this aim, we denote by R_i the index set of the extreme points of $\text{conv}(T_i^{xy}) \bigtimes \text{conv}(U_i^{vz})$ for $i \in N$, i.e., $(x(r), y(r), v(r), z(r))_{r \in R_i}, i \in N$, are these extreme points. We also denote by $\theta(r)$ the variable representing the weight associated with extreme point indexed by $r \in R_i, i \in N$. The DW reformulation of the Lagrangian dual is then:

$$Z^{FB} = \min \sum_{i \in N} \sum_{r \in R_i} \theta(r) \left(\sum_{j \in N_i^+} \sum_{k \in K} c_{ij}^k x_{ij}^k(r) + \sum_{j \in N_i^+} f_{ij} y_{ij}(r) \right) \tag{41}$$

$$\sum_{r \in R_j} \theta(r) z_{ij}(r) - \sum_{r \in R_i} \theta(r) y_{ij}(r) = 0, \quad \forall (i, j) \in A, \ (\gamma_{ij}) \tag{42}$$

$$\sum_{r \in R_j} \theta(r) v_{ij}^k(r) - \sum_{r \in R_i} \theta(r) x_{ij}^k(r) = 0, \quad \forall (i, j) \in A, \forall k \in K, \ (\omega_{ij}^k) \tag{43}$$

$$\sum_{j \in N_i^+} \sum_{q \in R_i} \theta(r) x_{ij}^k(r) - \sum_{j \in N_i^-} \sum_{r \in R_j} \theta(r) x_{ji}^k(r) = 0, \quad \forall i \in N, \forall k \in K_i^T, \ (\pi_i^k) \tag{44}$$

$$\sum_{r \in R_i} \theta(r) = 1, \quad \forall i \in N, \tag{45}$$

$$\theta(r) \geq 0, \quad \forall i \in N, r \in R_i. \tag{46}$$

Again, we propose to solve this LP model by column generation. At any column generation iteration, we can now derive three candidate design solutions: $\widetilde{y}$ and $\widetilde{z}$, corresponding to the 0-1 values of the design variables y and their copies z, respectively, in the optimal solution of the Lagrangian subproblem; $\overline{y}$, obtained from the DW master problem solution $\overline{\theta}$ and defined as $\overline{y}_{ij} = \sum_{r \in R_i} \overline{\theta}(r) y_{ij}(r)$, $\forall (i, j) \in A$.

By Lagrangian duality theory, we can show the following result:

Proposition 2 *$Z^{FB} \geq Z^{FL}$ and the inequality can be strict.*

Proof.

$$
\begin{aligned}
Z^{FB} &= \min_{\substack{x \in S^x \\ z=y \\ v=x}} \left\{ cx + fy \,|\, (x,y) \in \left(\bigtimes_{i \in N} \text{conv}(T_i^{xy}) \right), (v,z) \in \left(\bigtimes_{i \in N} \text{conv}(U_i^{vz}) \right) \right\} \\
&\geq \min_{\substack{x \in S^x \\ z=y \\ v=x}} \left\{ cx + fy \,|\, (x,y) \in \left(\bigtimes_{i \in N} \text{conv}(T_i^{xy}) \right), (v,z) \in \left(\bigtimes_{i \in N} \overline{U_i^{vz}} \right) \right\} \\
&= \min_{x \in S^x} \left\{ cx + fy \,|\, (x,y) \in \left(\bigtimes_{i \in N} (\text{conv}(T_i^{xy}) \cap \overline{U_i^{xy}}) \right) \right\} \\
&= \min_{x \in S^x} \left\{ cx + fy \,|\, (x,y) \in \left(\bigtimes_{i \in N} \text{conv}(T_i^{xy}) \right) \right\} \\
&= Z^{FL}
\end{aligned}
$$

Here, U_i^{xy} is the set of solutions that satisfy (34)-(40) with v replaced by x and z replaced by y. Because the CFLP does not have the integrality property, the polyhedron $\overline{U_i^{vz}}$ is not integral, i.e., we might have $\text{conv}(U_i^{vz}) \subset \overline{U_i^{vz}}$ for some $i \in N$, which implies that there are MCND instances where the inequality is strict. ■

Proposition 2 states that, in comparison with the facility location relaxation, the forward-backward facility location relaxation produces better lower bounds. There is a computational price to pay for such an improvement, since twice the number of CFLPs have to be solved at each column generation iteration ($2 \times |N|$ compared to $|N|$) and the number of Lagrange multipliers is increased from $\sum_{i \in N} |K_i^T|$ to $|A| + |A||K| + \sum_{i \in N} |K_i^T|$.

3.3 Multicommodity single-node flow relaxation

The forward-backward facility location relaxation is based on the reformulation obtained by replacing (2) with (7)-(9) and by adding (23)-(30). To define the third node-based relaxation, we use the same reformulation, except that the flow conservation equations (7) are replaced by:

$$
\sum_{j \in N_i^+} x_{ij}^k - \sum_{j \in N_i^-} v_{ji}^k = 0, \quad \forall i \in N, \forall k \in K_i^T. \tag{47}
$$

After applying Lagrangian relaxation on the copy constraints (23) and (24), the Lagrangian subproblem decomposes by nodes, using the fact that A can be partitioned into forward sets, $A = \cup_{i \in N} N_i^+$, or into backward sets, $A = \cup_{i \in N} N_i^-$. If we denote by γ_{ij}, $(i,j) \in A$, and ω_{ij}^k, $(i,j) \in A$, $k \in K$, the Lagrange multipliers associated with constraints (23) and (24), respectively, the Lagrangian subproblem for each node $i \in N$ can be written as:

$$Z_i^{xyvz}(\gamma,\omega) = \min \sum_{j \in N_i^+} \left(\sum_{k \in K} c_{ij}^k(\omega) x_{ij}^k + f_{ij}(\gamma) y_{ij} \right) + \sum_{j \in N_i^-} \left(\sum_{k \in K} c_{ji}^k(\omega) v_{ji}^k + f_{ji}(\gamma) z_{ji} \right) \tag{48}$$

subject to (12)-(18), (34)-(40) and

$$\sum_{j \in N_i^+} x_{ij}^k - \sum_{j \in N_i^-} v_{ji}^k = 0, \quad \forall k \in K_i^T, \tag{49}$$

where $f_{ij}(\gamma) = f_{ij} - \gamma_{ij}$, $j \in N_i^+$, $f_{ji}(\gamma) = \gamma_{ji}$, $j \in N_i^-$, $c_{ij}^k(\omega) = c_{ij}^k - \omega_{ij}^k$, $j \in N_i^+, k \in K$ and $c_{ji}^k(\omega) = \omega_{ji}^k$, $j \in N_i^-, k \in K$. This subproblem is a *multicommodity single-node fixed-charge network flow problem (MSNF)*. To the best of our knowledge, this problem has not been studied before, although its single-commodity variant has been the object of abundant work, both for characterizing its convex hull (see (Atamtürk, 2001) and the references therein) and for deriving efficient algorithms (at least for the special case with no transshipment, see (Klose, 2008) and the references therein). We call this relaxation the *multicommodity single-node flow relaxation*. We denote by V_i^{xyvz} the set of solutions that satisfy constraints (12)-(18), (34)-(40), and by W_i^{xv} the set of solutions to equations (49). The set of feasible solutions to the Lagrangian subproblem is thus $V_i^{xyvz} \cap W_i^{xv}$.

A lower bound on Z^{ND} is computed as follows: $Z^{SN}(\gamma,\omega) = \sum_{i \in N} Z_i^{xyvz}(\gamma,\omega)$. The best possible lower bound is obtained by solving the Lagrangian dual: $Z^{SN} = \max_{\gamma,\omega} Z^{SN}(\gamma,\omega)$. By Lagrangian duality theory, we have:

$$Z^{SN} = \min \left\{ cx + fy \,|\, z = y, v = x, (x,y,v,z) \in \left(\bigtimes_{i \in N} \text{conv}(V_i^{xyvz} \cap W_i^{xv}) \right) \right\}.$$

To derive the DW reformulation of the Lagrangian dual, we denote by P_i the index set of the extreme points of $\text{conv}(V_i^{xyvz} \cap W_i^{xv})$ for $i \in N$, i.e., $(x(p), y(p), v(p), z(p))_{p \in P_i}, i \in N$, are these extreme points. We also denote by $\theta(p)$ the variable representing the weight associated with extreme point indexed by $p \in P_i, i \in N$. The DW reformulation of the Lagrangian dual is then written as:

$$Z^{SN} = \min \sum_{i \in N} \sum_{p \in P_i} \theta(p) \left(\sum_{j \in N_i^+} \sum_{k \in K} c_{ij}^k x_{ij}^k(p) + \sum_{j \in N_i^+} f_{ij} y_{ij}(p) \right) \tag{50}$$

$$\sum_{p\in P_j}\theta(p)z_{ij}(p)-\sum_{p\in P_i}\theta(p)y_{ij}(p)=0,\quad \forall(i,j)\in A,\ (\gamma_{ij}) \tag{51}$$

$$\sum_{p\in P_j}\theta(p)v^k_{ij}(p)-\sum_{p\in P_i}\theta(p)x^k_{ij}(p)=0,\quad \forall(i,j)\in A,\forall k\in K,\ (\omega^k_{ij}) \tag{52}$$

$$\sum_{p\in P_i}\theta(p)=1,\quad \forall i\in N, \tag{53}$$

$$\theta(p)\geq 0,\quad \forall i\in N,\forall p\in P_i. \tag{54}$$

When solving this LP model by column generation, we can derive at any iteration three candidate design solutions: $\widetilde{y}$ and $\widetilde{z}$, corresponding to the 0-1 values of the design variables y and their copies z, respectively, in the optimal solution of the Lagrangian subproblem; $\overline{y}$, obtained from the DW master problem solution $\overline{\theta}$ and defined as $\overline{y}_{ij}=\sum_{p\in P_i}\overline{\theta}(p)y_{ij}(p)$, $\forall(i,j)\in A$.

Proposition 3 *$Z^{SN}\geq Z^{FB}$ and the inequality can be strict.*

Proof.

$$\begin{aligned}
Z^{SN} &= \min\left\{cx+fy\,|\,z=y,v=x,(x,y,v,z)\in\left(\bigtimes_{i\in N}\operatorname{conv}(V_i^{xyvz}\cap W_i^{xv})\right)\right\}\\
&\geq \min\left\{cx+fy\,|\,z=y,v=x,(x,y,v,z)\in\left(\bigtimes_{i\in N}(\operatorname{conv}(V_i^{xyvz})\cap\operatorname{conv}(W_i^{xv}))\right)\right\}\\
&= \min\left\{cx+fy\,|\,z=y,v=x,(x,y,v,z)\in\left(\bigtimes_{i\in N}(\operatorname{conv}(V_i^{xyvz})\cap W_i^{xv})\right)\right\}\\
&= \min\left\{cx+fy\,|\,z=y,v=x,(x,y,v,z)\in\left(\bigtimes_{i\in N}W_i^{xv}\right)\cap\left(\bigtimes_{i\in N}\operatorname{conv}(V_i^{xyvz})\right)\right\}\\
&= \min\left\{cx+fy\,|\,x\in S^x,z=y,v=x,(x,y,v,z)\in\left(\bigtimes_{i\in N}\operatorname{conv}(V_i^{xyvz})\right)\right\}\\
&= \min_{\substack{x\in S^x\\ z=y\\ v=x}}\left\{cx+fy\,|\,(x,y)\in\left(\bigtimes_{i\in N}\operatorname{conv}(T_i^{xy})\right),(v,z)\in\left(\bigtimes_{i\in N}\operatorname{conv}(U_i^{vz})\right)\right\}\\
&= Z^{FB}
\end{aligned}$$

Since solving independently the two CFLPs for each $i\in N$ (one over the set T_i^{xy}, the other over the set U_i^{vz}) does not guarantee that constraints (49) are satisfied, there are instances for which $\operatorname{conv}(V_i^{xyvz}\cap W_i^{xv})\subset(\operatorname{conv}(V_i^{xyvz})\cap\operatorname{conv}(W_i^{xv}))$, which implies that there are MCND instances for which $Z^{SN}>Z^{FB}$.

■

Proposition 3 states that, in comparison with the forward-backward location relaxation, the multicommodity single-node flow relaxation produces better lower bounds. However, the Lagrangian subproblem of the multicommodity single-node flow relaxation is more difficult to solve, as it does not decompose into $2 \times |N|$ independent CFLPs.

4 Lagrangian matheuristic

Traditional Lagrangian heuristics alternate between solving series of Lagrangian subproblems, each providing a partial solution that is given as input to a primal heuristic that restores the feasibility of this partial solution, often through a simple local search procedure. Our Lagrangian matheuristic does not differ fundamentally from this classical approach, but instead of simple local search, it uses state-of-the-art mathematical programming methods to restore feasibility, while also exploiting concepts from the metaheuristics literature, such as intensification and diversification. An advantage of this approach over many heuristics is that it works simultaneously on the dual and the primal sides to produce feasible solutions with a measurable quality.

On the dual side, the Lagrangian matheuristic solves by column generation the DW reformulation associated with one of the three node-based Lagrangian relaxations presented in Section 3. The choice of which of the three relaxations to select depends on the tradeoff between the quality of the lower bound and the time needed to compute that bound. In our computational results presented in Section 5, we compare the results obtained with the three relaxations on a set of benchmark instances. In addition to measuring the performance of the three relaxations, we also compare them to the classical flow and knapsack relaxations.

On the primal side, the Lagrangian matheuristic receives candidate design solutions derived from the column generation method, either from the Lagrangian subproblem solutions (as in a classical Lagrangian heuristic) or from the restricted master problem. In this last case, the candidate design solution can assume fractional values, as it is a convex combination of candidate design solutions derived from the Lagrangian subproblems. These candidate design solutions are used as input to a slope scaling procedure that derives feasible solutions to the MCND, as explained in Section 4.2.

In what follows, we assume the availability of a state-of-the-art LP code to solve the DW master problem at each column generation iteration. In addition, we assume the

availability of algorithms for the following problems:

- **CFLP.** An efficient algorithm for the CFLP is needed to solve the Lagrangian subproblems arising from the facility location and forward-backward facility location relaxations. For the column generation method to converge, an exact algorithm should be used, but it could be stopped before optimality is achieved provided it delivers at least one feasible solution (see Section 4.1). Several algorithms for the CFLP have been proposed in the literature (see (Fischetti et al., 2016) and the references therein). Given the relatively small size of the instances we have to solve (there are $O(10)$ facilities and $O(100)$ customers for each CFLP when the decomposition by nodes is applied on an MCND instance with $O(100)$ arcs and $O(100)$ commodities), a viable alternative is to use a state-of-the-art MIP solver. In our experiments, we choose this option and use CPLEX.
- **MSNF.** An exact algorithm for the MSNF is required to ensure convergence of the column generation method applied to the multicommodity single-node flow relaxation. We are not aware of any existing specialized algorithms for this problem, hence we use the state-of-the-art MIP solver CPLEX, which could be stopped before optimality is achieved (see Section 4.1).
- **MMCF.** The slope scaling heuristic described in Section 4.2 solves a linear MMCF at each iteration. Although there are several specialized decomposition methods for this problem (see (Babonneau et al., 2006) and the references therein), the MMCF instances derived from our benchmark are relatively easy to solve for general-purpose state-of-the-art LP solvers, so we use CPLEX in our experiments.
- **MCND.** We assume the availability of an algorithm to solve small- to medium-scale instances of the MCND derived from any large-scale instance by fixing the values of some design variables. Such an algorithm is typically used as a heuristic by limiting the computational effort. We assume it is easy to provide as input to this algorithm a feasible solution, along with upper and lower bounds on the optimal value of the original, large-scale, instance. In our experiments, we give the MIP model to CPLEX to ease the implementation. Other viable alternatives would be the exact methods described in Section 2.

The Lagrangian matheuristic consists of four main steps (see Section 5 for the time limits for each step used in our experiments):

1. **Solving the unrestricted MCND.** The original problem is solved for a limited time using CPLEX in order to find an initial feasible solution and to put aside the instances for which the optimal solutions are already found at this step.

2. **Solving the Lagrangian dual.** During this step, the column generation method is used to find lower bounds, while the slope scaling procedure is called on a regular basis to compute upper bounds (see Section 4.3 for details). The slope scaling heuristic receives as input the candidate design solutions computed by the column generation procedure. The slope scaling heuristic is also enhanced with intensification and diversification procedures (see Section 4.4).

3. **Solving restricted MCNDs.** Several restricted MCNDs are solved to improve the upper bound. A long-term memory of elite solutions, constantly updated during the calls to the slope scaling procedure, is used during this step, which can be seen as a form of intensification (see Section 4.4).

4. **Solving the bounded unrestricted MCND.** The original problem is solved again for a limited time, by providing as input the best feasible solution identified so far, as well as the best upper and lower bounds on the objective value.

4.1 Lagrangian dual optimization

We use column generation to solve the Lagrangian dual. The column generation method is stopped when it has converged or when a maximum number of iterations is achieved (1,000 in our experiments) or when a time limit is reached (including the time spent in the slope scaling heuristic). These parameters were calibrated in our experiments (see Section 5). At each iteration, the Lagrangian subproblem of the corresponding node-based relaxation has to be solved in order to obtain a lower bound and new columns to add to the DW master problem.

When solving the Lagrangian subproblems, we add the following knapsack inequalities to the CFLPs or the MSNFs associated to a given node $i \in N$, since these inequalities generally improve the performance of algorithms to solve these problems:

$$\sum_{j \in N_i^+} u_{ij} y_{ij} \geq \sum_{k \in K_i^O} d^k,$$

$$\sum_{j \in N_i^-} u_{ji} z_{ji} \geq \sum_{k \in K_i^D} d^k.$$

Even with these enhancements, solving the subproblems might be costly in terms of computational time. To accelerate the solution of the subproblems, we use the following approach. Assuming that each node-based subproblem (either a CFLP or an MSNF) is solved by B&B, we stop the B&B algorithm either when an optimality gap ϵ is attained

or when a limited number of nodes L is reached. During the course of the column generation method, the values of the two parameters are gradually modified, so that early termination of the B&B is favored during the first iterations, while the B&B is almost exact near the end. The parameters are initially set to $\epsilon = 10^{-2}$ and $L = 10$. Subsequently, every 5 column generation iterations, ϵ is decreased by multiplying it by 0.95, down to a limit of $\epsilon = 10^{-6}$, while L is multiplied by 10.

4.2 Slope scaling heuristic

To obtain feasible solutions to the MCND, we use a slope scaling procedure, where the following linear MMCF is solved at each iteration:

$$\min \sum_{(i,j)\in A} \sum_{k\in K} \overline{c}_{ij}^k x_{ij}^k \tag{55}$$

subject to (2), (5) and

$$\sum_{k\in K} x_{ij}^k \leq u_{ij}, \quad \forall (i,j) \in A. \tag{56}$$

Given an optimal solution $\overline{x}$ to this MMCF, a feasible solution $(\overline{x}, \overline{y})$ to the MCND is immediately derived:

$$\overline{y}_{ij} = \left\lceil \frac{\sum_{k\in K} \overline{x}_{ij}^k}{u_{ij}} \right\rceil, \quad (i,j) \in A.$$

Provided by the column generation method with a candidate design solution $\widehat{y}$, the slope scaling procedure starts with the following initial linearized costs, where M is a large positive value:

$$\overline{c}_{ij}^k = (c_{ij}^k + f_{ij}/u_{ij})(1 + M(1 - \widehat{y}_{ij})), \quad \forall (i,j) \in A, \forall k \in K. \tag{57}$$

When $\widehat{y}_{ij} = 1$, for some arc $(i,j) \in A$, the corresponding costs are linearized in such a way that, if the arc is used at its full capacity in the optimal solution $\overline{x}$ to the MMCF, then the exact costs, variable and fixed, would be accounted for in the feasible solution $(\overline{x}, \overline{y})$ to the MCND. If $\widehat{y}_{ij} = 0$, for some arc $(i,j) \in A$, then the linearized costs are set to large positive values to "discourage" any commmodity to be routed on arc (i,j). Note that, even if $\widehat{y}$ has fractional components, formula (57) can be used and has a similar interpretation: the commodities are "discouraged" (respectively, "encouraged") to be routed on the arcs with $\widehat{y}$ values close to 0 (respectively, close to 1).

After solving the MMCF at each iteration, a flow solution $\overline{x}$ is obtained and the linearized costs are updated using the following formula to trigger the next iteration:

$$\overline{c}_{ij}^k = \begin{cases} c_{ij}^k + f_{ij} / \sum_{k\in K} \overline{x}_{ij}^k, & \text{if } \overline{x}_{ij}^k > 0, \\ \overline{c}_{ij}^k, & \text{if } \overline{x}_{ij}^k = 0, \end{cases} \quad \forall (i,j) \in A, \forall k \in K. \tag{58}$$

If $\overline{x}^k_{ij} = 0$, the linear cost at the previous iteration is kept as is, since this cost was already large enough to incur no flow. When $\overline{x}^k_{ij} > 0$, this formula ensures that, at the next iteration, if the solution remains the same, the linear costs of $\overline{x}$ reflect the exact costs, variable and fixed, of $(\overline{x}, \overline{y})$, the corresponding feasible solution to the MCND, i.e., $\sum_{k\in K}\sum_{(i,j)\in A} \overline{c}^k_{ij}\overline{x}^k_{ij} = \sum_{k\in K}\sum_{(i,j)\in A} c^k_{ij}\overline{x}^k_{ij} + \sum_{(i,j)\in A} f_{ij}\overline{y}_{ij}$. The slope scaling procedure is stopped either when the same objective value is obtained for two successive iterations or when a predefined maximum number of iterations is attained (25 in our experiments).

4.3 Combining slope scaling and Lagrangian dual optimization

At each iteration of the column generation method, several candidate design solutions are generated. For the facility location relaxation, the Lagrangian subproblem produces one candidate design solution $\widetilde{y}$, while for the forward-backward facility location and multicommodity single-node flow relaxations, two candidate design solutions, $\widetilde{y}$ and $\widetilde{z}$, are generated. In addition, the solution of the DW master problem for all three relaxations provide a (fractional) candidate design solution $\overline{y}$. These candidate design solutions are given as input to the slope scaling procedure (each such candidate design solution is stored in a memory in order to avoid calling the procedure twice with the same input).

We use the following rules to decide when to call the slope scaling procedure in conjunction with the column generation method:

- Call the slope scaling procedure using the candidate design solutions that correspond to the best lower bound obtained at the end of the column generation method.
- Call the slope scaling procedure if the lower bound has improved "significantly" since the last time the upper bound was computed. The improvement is considered "significant" if $(Z_c - Z_l)/Z_l > \delta$, where δ is a parameter and Z_c and Z_l are, respectively, the lower bound computed at the current iteration and the lower bound obtained the last time the slope scaling procedure was called.
- Call the slope scaling procedure every n^{th} iteration of the column generation method (to avoid too early "freezing" of the upper bound in case δ is too large).

The parameters δ and n were calibrated for each Lagrangian relaxation and the values used in our experiments are given in Section 5.1.2.

4.4 Intensification and diversification

At the end of the slope scaling procedure, two intensification steps are performed. The first intensification selects the κ best feasible solutions obtained during this call to the procedure (in our experiments, we use $\kappa = 4$). For each such solution $(\overline{x}, \overline{y})$, a linear MMCF is defined by setting the costs to the following values, where M is a large positive value:

$$\overline{c}_{ij}^k = \begin{cases} c_{ij}^k, & \text{if } \overline{y}_{ij} > 0, \\ M, & \text{if } \overline{y}_{ij} = 0, \end{cases} \qquad \forall (i,j) \in A, \forall k \in K.$$

Note that $\overline{x}$ is feasible for this MMCF, but not necessarily optimal. This MMCF is solved in the hope of obtaining a better flow distribution for the same assignment of the design variables. The second intensification consists in solving a restricted MCND (for a limited time, set to 200 seconds in our experiments) using as guides the best solution obtained after the first intensification and the best solution found so far. This restricted MCND is obtained by fixing to 0 the design variables that assume value 0 in both of these solutions.

The remaining intensification and diversification mechanisms make use of a long-term memory of the best feasible solutions found so far, denoted $\mathcal{P}$. We limit the number of solutions (1,000 in our experiments) kept in this memory and update it all along the algorithm.

In the case where several successive calls (5 in our experiments) to the slope scaling procedure (including the two intensifications) could not improve the upper bound, a diversification step is performed. The idea of this diversification step is to "discourage" the selection of arcs that are frequently used in the solutions stored in $\mathcal{P}$. This diversification step consists of a small number (at most 10) of calls to the slope scaling procedure, each time using as input a candidate design solution $\widehat{y}$ corresponding to the best solution in $\mathcal{P}$, slightly modified by closing 10 arcs, randomly selected from the most frequently used arcs (in at least 90% of) the solutions stored in $\mathcal{P}$. The selected arcs are tagged to avoid choosing them again in a subsequent slope scaling call during the same diversification step.

After solving the Lagrangian dual, including the calls to the slope scaling procedure and the intensification/diversification steps just described, we enter an intensification phase where several restricted MCNDs are solved, each one being stopped when a predefined optimality gap is reached (0.3% in our experiments). To define the $t^{\text{th}}, t \geq 1$ restricted MCND, we select the $t + 1$ best solutions from $\mathcal{P}$. The design variables that assume value 0 in all these solutions are fixed to 0. Each restricted MCND is enhanced by providing as input the best feasible solution. We also add a constraint that bounds

from below the objective value, using the best lower bound computed by the column generation method. This intensification phase stops when it reaches a time limit or when the t^{th} best solution, to be used to define the next restricted MCND, is "too far" (in quality) from the overall best solution, i.e., when the gap between the two solutions is "too large" (we use 20% in our experiments).

5 Computational results

To implement the column generation method, we use BTT, version 3.44, a publicly available code that enables to implement subgradient, bundle and column generation methods for solving Lagrangian duals. In particular, the code solves the DW master problems either with quadratic or linear solvers, and it can construct aggregated (i.e., each Lagrangian subproblem provides a single subgradient) or disaggregated (i.e., the Lagrangian subproblem decomposes into several subproblems, each providing its part of a subgradient) DW master problems. For the three node-based relaxations, we solve disaggregated DW master problems, as described in Section 3, using CPLEX, which is called by BTT. We also use CPLEX to solve the different MIP models, i.e., CFLP, MSNF and MCND. We use the default parameters of CPLEX, except when solving the restricted MCNDs, where we deactivated cut generation to focus on the fast computation of feasible solutions.

We compare the three node-based relaxation with one another, as well as with the classical flow and knapsack relaxations. For the flow relaxation, we solve the (disaggregated) DW reformulation with "easy components" using BTT (Frangioni and Gorgone, 2014), which can be considered as state-of-the-art. For the knapsack relaxation, we use the bundle method implemented in BTT (with a limit of 10,000 iterations) that solves the (aggregated) DW reformulation with a "stabilizing" quadratic objective function (this variant was shown to be the most effective in preliminary tests). We denote by Z^{FW} and Z^{KN} the Lagrangian dual bound for the flow and knapsack relaxations, respectively. For both relaxations, we adapted in a straightforward way the Lagrangian matheuristic presented in Section 4.

We tested our algorithms on two sets of instances, the *benchmark* instances and the *large* instances. The *benchmark* instances comprise 43 of the most difficult C and C+ instances widely used in the literature (Crainic et al., 2001). These instances are selected because they serve as benchmark to compare the performance of heuristics for the MCND. For this set of instances, the number of nodes is between 20 and 100, the number of arcs varies from 100 to 700, while the number of commodities ranges from 10 to 400.

The 48 instances in the second set, initially generated to test advanced decomposition methods (Frangioni and Gorgone, 2014), include larger instances, as the number of nodes varies between 20 and 50, the number of arcs is between 300 and 1,200, while the number of commodities ranges between 100 and 800. To the best of our knowledge, no upper bounds were ever reported in the literature for these 48 *large* instances. All instances consist of general transshipment networks with one commodity per origin-destination pair and no parallel arcs. Each test instance is characterized by the number of nodes $|N|$, the number of arcs $|A|$, the number of commodities $|K|$, the degree of capacity tightness, with regard to the total demand, and the importance of the fixed costs, with respect to the variable costs. Each *benchmark* instance is identified with five entries of the form $|N|$,$|A|$,$|K|$, "F" or "V","T" or "L", where the first three represent the dimensions, while the last two indicate if fixed ("F") or variable ("V") costs are dominant and whether the instance is tightly ("T") or loosely ("L") capacitated. The *large* instances are numbered from 1 to 48. In order to easily identify them, we simply denote them with their number, following their dimensions, $|N|$,$|A|$,$|K|$. All instances can be found at `http://groups.di.unipi.it/optimize/Data/MMCF.html`, while the BTT code can be downloaded from `https://gitlab.com/frangio68/ndosolver_fioracle_project`.

We present next the results of our experiments on the *benchmark* instances in Section 5.1. Section 5.2 then focuses on the results of our experiments on the *large* instances.

5.1 Experiments on the *benchmark* instances

All experiments on the benchmark instances were performed on a computer with Intel Xeon-X5675 CPU 3.07GHz, using CPLEX version 12.6.3. We divide our analysis of the computational results into two parts. First, in Section 5.1.1, we focus on the effectiveness of the Lagrangian lower bounds, as well as on the efficiency of the column generation method to compute these bounds. Then, in Section 5.1.2, we analyze the performance of the Lagrangian matheuristic when used with the different relaxations and then compare its performance with the heuristics presented in the literature.

5.1.1 Lower bound computations

Table 1 presents a comparison of the lower bounds obtained by solving the Lagrangian duals of the different relaxations. The LP relaxation bound is also computed with CPLEX. In these experiments, the time limit is set to 2 hours and only the column generation (or bundle) method is performed (there is no call to the slope scaling heuristic). The

first column shows the characteristics of the instances. Column Z^{LP} is the LP relaxation bound, while the next columns show the gaps between the lower bound of the different relaxations and the LP relaxation bound, i.e., $100 \times (Z^{LP} - Z^{LR})/Z^{LP}$ where Z^{LR} is the lower bound of any one of the five Lagrangian relaxations (a negative value indicates a better lower bound than Z^{LP}).

Theoretically, the lower bounds obtained by the flow and knapsack relaxations are equal to the LP relaxation bound. Therefore, positive values in columns Z^{FW} and Z^{KN} indicate that the computations were stopped due to either the numerical accuracy to declare convergence, or the time and iteration limits. It is noteworthy that the facility location relaxation produces a better lower bound than Z^{LP} for all instances. It is not the case, however, for the forward-backward facility location and multicommodity single-node relaxations, for which the lower bounds are worse for 3 and 8 instances, respectively. For these large-scale instances with 400 commodities, the computations were stopped prematurely because of the time limit. Overall, the improvements over Z^{LP} obtained by Z^{FL}, Z^{FB} and Z^{SN} are significant: on average, 0.5%, 0.9%, and 1.8%, respectively, with maximum values of 4.8%, 7.7%, and 20.5%, respectively.

Table 2 shows the gaps between the Lagrangian relaxation lower bounds and the best known upper bounds reported in the literature, shown in column Z_U^{ND}. The remaining columns display the gaps, computed as $100 \times (Z_U^{ND} - Z_{LR})/Z_U^{ND}$, where Z^{LR} is the lower bound obtained by any of the five Lagrangian relaxations, respectively. The results show that, in comparison with the flow and knapsack relaxations, the node-based relaxations improve the gaps significantly, in particular the multicommodity single-node flow relaxation, with the exception of the instances with 400 commodities.

Table 3 shows statistics on the computational performance of the different relaxation methods. The first row presents the total computational times, on average over all instances. The fastest method by far is the flow relaxation. It is worth noting that both the flow and the knapsack relaxations are faster than CPLEX to compute the LP relaxation bound. Although the times are significant for the facility location relaxation, they are still reasonable (about four times slower than the LP relaxation computed by CPLEX). The results show, however, a very significant computational effort for the forward-backward and single-node flow relaxations, as they are both one order of magnitude slower than the facility location relaxation. The second row shows the number of master problem iterations for the different relaxations. The third and fourth rows present the fractions of the times dedicated to the solution of the Lagrangian subproblems and to the master problems, respectively. The table shows that a large proportion of the time is devoted to the solution of the master problems for all relaxations, except for the multicommodity single-node flow relaxation for which most of the time (65%) is spent in solving the

Instance	Z^{LP}	Z^{FW}	Z^{KN}	Z^{FL}	Z^{FB}	Z^{SN}
25,100,10VL	14610.0	0.000	0.000	-0.074	-0.133	-0.224
25,100,10FL	13017.5	0.000	0.000	-1.459	-3.595	-6.794
25,100,10FT	43454.7	0.001	0.000	-2.402	-3.647	-10.332
25,100,30VT	364390.0	0.000	0.000	-0.044	-0.074	-0.145
25,100,30FL	33543.5	0.010	0.000	-0.842	-4.291	-6.815
25,100,30FT	82418.5	0.000	0.000	-0.555	-0.605	-2.168
20,230,40VL	422853.0	0.000	0.000	-0.011	-0.201	-0.207
20,230,40VT	368819.0	0.000	0.000	-0.186	-0.400	-0.613
20,230,40FT	633466.0	0.000	0.000	-0.302	-0.619	-0.878
20,300,40VL	427947.0	0.000	0.000	0.000	-0.256	-0.338
20,300,40FL	575255.0	0.000	0.001	-0.004	-0.011	-1.047
20,300,40VT	460930.0	0.000	0.000	-0.175	-0.312	-0.501
20,300,40FT	596839.0	0.000	0.000	-0.143	-0.269	-0.792
20,230,200VL	91300.6	0.000	0.022	-0.213	-0.189	-0.079
20,230,200FL	132036.0	0.000	0.025	-0.326	-0.273	-0.358
20,230,200VT	95669.3	0.008	0.019	-0.113	-0.098	-0.134
20,230,200FT	131544.0	0.002	0.023	-0.160	-0.125	-0.138
20,300,200VL	73126.9	0.006	0.016	-0.094	-0.091	-0.133
20,300,200FL	110926.0	0.000	0.012	-0.105	-0.075	-0.303
20,300,200VT	74002.7	0.008	0.005	-0.050	-0.045	-0.199
20,300,200FT	103633.0	0.005	0.009	-0.327	-0.315	-0.185
100,400,10VL	27465.3	0.000	0.007	-0.063	-0.598	-1.890
100,400,10FL	19748.1	0.002	0.000	-3.871	-6.351	-11.112
100,400,10FT	48375.4	0.000	0.000	-4.767	-7.713	-20.518
100,400,30VT	380858.0	0.001	0.000	-0.068	-0.135	-0.521
100,400,30FL	45332.4	0.000	0.002	-1.412	-2.299	-5.048
100,400,30FT	117196.0	0.002	0.000	-1.974	-4.790	-10.023
30,520,100VL	53022.9	0.007	0.002	-0.291	-0.309	-0.732
30,520,100FL	90174.2	0.009	0.045	-0.097	-0.067	-0.282
30,520,100VT	51325.6	0.002	0.005	-0.090	-0.397	-0.632
30,520,100FT	94010.8	0.003	0.021	-0.273	-0.301	-0.398
30,700,100VL	47308.1	0.000	0.001	-0.052	-0.061	-0.302
30,700,100FL	58207.2	0.003	0.018	-0.089	-0.094	-0.360
30,700,100VT	45077.7	0.000	0.006	-0.301	-0.388	-0.568
30,700,100FT	53660.9	0.005	0.015	-0.226	-0.307	-0.407
30,520,400VL	111763.0	0.007	0.011	-0.106	-0.070	0.278
30,520,400FL	146680.0	0.005	0.011	-0.085	-0.030	0.647
30,520,400VT	114061.0	0.007	0.012	-0.071	-0.053	0.309
30,520,400FT	149751.0	0.003	0.017	-0.083	-0.031	0.399
30,700,400VL	96605.0	0.000	0.015	-0.080	0.035	1.517
30,700,400FL	130724.0	0.004	0.017	-0.099	0.082	2.454
30,700,400VT	94011.9	0.006	0.010	-0.097	-0.018	1.581
30,700,400FT	127572.0	0.008	0.011	-0.056	0.023	1.397
	Average:	0.003	0.008	-0.508	-0.919	-1.781
	Maximum:	0.010	0.045	0.000	0.082	2.454
	Minimum:	0.000	0.000	-4.767	-7.713	-20.518

Table 1: Comparison between lower bounds - *benchmark* instances

Instance	Z_U^{ND}	Z^{FW}	Z^{KN}	Z^{FL}	Z^{FB}	Z^{SN}
25,100,10VL	14712.0	0.69	0.69	0.62	0.56	0.47
25,100,10FL	14941.0	12.87	12.87	11.60	9.74	6.95
25,100,10FT	49899.0	12.92	12.91	10.82	9.74	3.92
25,100,30VT	365272.0	0.24	0.24	0.20	0.17	0.10
25,100,30FL	37060.0	9.50	9.49	8.73	5.60	3.32
25,100,30FT	85530.0	3.64	3.64	3.10	3.06	1.55
20,230,40VL	423848.0	0.23	0.23	0.22	0.03	0.03
20,230,40VT	371475.0	0.71	0.72	0.53	0.32	0.11
20,230,40FT	643036.0	1.49	1.49	1.19	0.88	0.62
20,300,40VL	429398.0	0.34	0.34	0.34	0.08	0.00
20,300,40FL	586077.0	1.85	1.85	1.84	1.84	0.82
20,300,40VT	464509.0	0.77	0.77	0.60	0.46	0.27
20,300,40FT	604198.0	1.22	1.22	1.08	0.95	0.44
20,230,200VL	94213.0	3.09	3.11	2.88	2.91	3.01
20,230,200FL	137642.0	4.07	4.10	3.76	3.81	3.73
20,230,200VT	97914.0	2.30	2.31	2.18	2.20	2.16
20,230,200FT	135866.0	3.18	3.20	3.03	3.06	3.05
20,300,200VL	74811.0	2.26	2.27	2.16	2.16	2.12
20,300,200FL	115539.0	3.99	4.00	3.89	3.92	3.70
20,300,200VT	74991.0	1.33	1.32	1.27	1.27	1.12
20,300,200FT	107102.0	3.24	3.25	2.92	2.93	3.06
100,400,10VL	28423.0	3.37	3.38	3.31	2.79	1.54
100,400,10FL	23949.0	17.54	17.54	14.35	12.30	8.38
100,400,10FT	63753.0	24.12	24.12	20.50	18.27	8.55
100,400,30VT	384802.0	1.03	1.02	0.96	0.89	0.51
100,400,30FL	49018.0	7.52	7.52	6.21	5.39	2.85
100,400,30FT	136250.0	13.99	13.98	12.29	9.86	5.36
30,520,100VL	53958.0	1.74	1.74	1.45	1.43	1.01
30,520,100FL	93967.0	4.04	4.08	3.94	3.97	3.77
30,520,100VT	52046.0	1.39	1.39	1.30	0.99	0.76
30,520,100FT	97107.0	3.19	3.21	2.92	2.90	2.80
30,700,100VL	47603.0	0.62	0.62	0.57	0.56	0.32
30,700,100FL	59958.0	2.92	2.94	2.83	2.83	2.57
30,700,100VT	45871.5	1.73	1.74	1.44	1.35	1.17
30,700,100FT	54904.0	2.27	2.28	2.04	1.96	1.87
30,520,400VL	112774.4	0.90	0.91	0.79	0.83	1.17
30,520,400FL	149335.4	1.78	1.79	1.69	1.75	2.41
30,520,400VT	114640.0	0.51	0.52	0.43	0.45	0.81
30,520,400FT	152510.0	1.81	1.83	1.73	1.78	2.20
30,700,400VL	97875.0	1.30	1.31	1.22	1.33	2.80
30,700,400FL	134589.8	2.88	2.89	2.78	2.95	5.26
30,700,400VT	95249.6	1.31	1.31	1.20	1.28	2.86
30,700,400FT	129909.6	1.81	1.81	1.74	1.82	3.17
	Average:	3.90	3.91	3.46	3.10	2.39
	Maximum:	24.12	24.12	20.50	18.27	8.55
	Minimum:	0.23	0.23	0.20	0.03	0.00

Table 2: Comparison between lower bounds and best upper bound - *benchmark* instances

	Z^{LP}	Z^{FW}	Z^{KN}	Z^{FL}	Z^{FB}	Z^{SN}
Total time (sec.)	170.25	7.40	125.56	699.74	4073.34	4677.71
Number of iterations	—	20	5866	284	373	316
Lagrangian subproblem time (%)	—	5	18	28	10	65
Master problem time (%)	—	95	82	72	90	35

Table 3: Computational performance of Lagrangian relaxations - *benchmark* instances

Lagrangian subproblems.

5.1.2 Upper bound computations

We use a two-phase parameter calibration process to find suitable values for the parameters. In the first phase, we tune the parameters associated with a particular relaxation and the column generation method. In the second phase, we calibrate the slope scaling heuristic parameters in two steps. The first step is to calibrate n and δ, the parameters used to initiate a call to the slope scaling procedure (see Section 4.3). Since the total number of column generation iterations varies significantly for different relaxations, these two parameters are tuned for each relaxation separately. The values used in our experiments are shown in Table 4. The second step is to calibrate the other parameters, for which the same value is calibrated for all the relaxations. For both calibration phases, we select 30% of the instances randomly. To find the best values of the parameters, we fix all the parameters, change one parameter at each time, and select the best value. After this calibration process, the following time limits are used for each the four main steps of the Lagrangian matheuristic (the other parameter values selected after the calibration phase were already given in Section 4):

1. *Solving the unrestricted MCND.* The original problem is solved using CPLEX for a maximum of 30 minutes.

2. *Solving the Lagrangian dual.* The column generation method (including calls to the slope scaling heuristic) is performed for a maximum of 4,200 seconds.

3. *Solving restricted MCNDs.* This intensification step stops when it reaches a time limit of 4,200 seconds.

4. *Solving the bounded unrestricted MCND.* The original problem is solved again using CPLEX (provided with the best feasible solution and bounds) for a maximum of 4,200 seconds, but not exceeding 3 hours for all four steps.

Parameter	Z^{FW}	Z^{KN}	Z^{FL}	Z^{FB}	Z^{SN}
n	1	5	10	5	20
δ	-	3	0.5	0.5	2

Table 4: Values of parameters to call slope scaling heuristic

Table 5 compares the upper bounds computed by the Lagrangian matheuristic for each of the five different relaxations. Note that the instances for which CPLEX finds an optimal solution in less than 30 minutes are not presented in the table. This leaves 25 instances out of the 43 original ones. The results of the ILP heuristic (Gendron et al., 2018) (that can be considered as state-of-the-art) are used as a basis of comparison. Column Z_{ILP} presents the upper bounds obtained with this method. The other columns show the gaps between Z^R_{LMH}, the upper bound obtained by the Lagrangian matheuristic used with relaxation R, and Z_{ILP}, computed as $100 \times (Z^R_{LMH} - Z_{ILP})/Z_{ILP}$. The results show that all variants of the Lagrangian matheuristic are competitive with ILP, producing better upper bounds on average. Among the different Lagrangian relaxations, the facility location relaxation obtains slightly better upper bounds on average than the others.

Table 6 displays the proportion of the time spent in each step of the Lagrangian matheuristic for the different relaxations. We divide the time spent by the column generation method into three parts: Lagrangian subproblems, master problems and slope scaling heuristic. The other phases are the ones described above: solving the unrestricted MCND; solving restricted MCNDs; solving the bounded unrestricted MCND.

Table 7 presents a comparison between the Lagrangian matheuristic based on the facility location relaxation and the state-of-the-art heuristics proposed in the literature. Column `LMH` presents the upper bounds computed by the Lagrangian matheuristic. The other columns correspond to the gaps (in %) between the upper bound Z_{LMH} in column `LMH` and the upper bound Z_H reported in the literature (computed as $100 \times (Z_{LMH} - Z_H)/Z_{LMH}$) for the following methods:

- `CTS`: Cycle-based Tabu Search (2003) (Ghamlouche et al., 2003);
- `PR`: Path Relinking (2004) (Ghamlouche et al., 2004);
- `MCA`: Multilevel Cooperative Tabu Search (2006) (Crainic et al., 2006);
- `CSH`: Capacity Scaling Heuristic (2009) (Katayama et al., 2009);
- `IPS`: IP Search (2010) (Hewitt et al., 2010);

Instance	Z_{ILP}	Z_{LMH}^{FW}	Z_{LMH}^{KN}	Z_{LMH}^{FL}	Z_{LMH}^{FB}	Z_{LMH}^{SN}
20,230,200VL	94213.0	0.044	0.000	0.087	0.044	0.014
20,230,200FL	138169.0	-0.293	-0.037	-0.037	-0.037	-0.037
20,230,200VT	97914.0	0.000	0.000	0.000	0.000	0.000
20,230,200FT	136513.0	-0.097	-0.023	-0.476	0.234	0.232
20,300,200VL	74971.0	-0.213	-0.188	0.167	-0.011	0.000
20,300,200FL	116375.0	-0.570	-0.077	-0.605	-0.611	-0.414
20,300,200VT	74991.0	0.000	0.000	0.000	0.000	0.000
20,300,200FT	107298.0	0.256	-0.183	0.431	0.501	0.429
100,400,10FT	65247.0	-2.273	-2.273	-2.273	-2.290	-2.290
100,400,30FL	49018.0	0.000	0.000	0.198	0.000	0.000
100,400,30FT	139177.0	-1.093	-1.206	-1.390	-0.669	-0.421
30,520,100VL	53958.0	0.000	0.000	0.000	0.000	0.000
30,520,100FL	94066.0	0.327	-0.035	-0.086	0.453	0.934
30,520,100VT	52046.0	0.004	0.004	0.000	0.000	0.000
30,520,100FT	97404.0	-0.314	-0.314	0.355	0.538	0.633
30,700,100FL	60049.0	0.050	0.008	0.012	0.078	0.150
30,700,100VT	45908.0	-0.080	-0.080	-0.039	-0.072	-0.072
30,520,400VL	112974.0	-0.112	-0.102	0.201	0.321	-0.081
30,520,400FL	149945.0	0.476	0.524	-0.001	0.022	-0.148
30,520,400VT	114798.0	-0.097	0.097	-0.050	-0.129	-0.138
30,520,400FT	153856.0	-0.308	0.074	-0.572	-0.445	0.036
30,700,400VL	98385.0	-0.299	-0.299	-0.394	-0.421	-0.171
30,700,400FL	139663.0	-2.691	-2.184	-3.264	-3.185	-3.097
30,700,400VT	95733.0	-0.237	-0.167	-0.288	-0.295	-0.270
30,700,400FT	131141.0	-0.494	-0.357	-0.253	-0.377	-0.042
	Average:	-0.321	-0.273	**-0.331**	-0.254	-0.190

Table 5: Comparison of upper bounds for different variants of Lagrangian matheuristic - *benchmark* instances

	Unrestricted MCND	Subproblem	Master problem	Slope scaling	Restricted MCNDs	Bounded unrestricted MCND
Z_{LMH}^{FW}	16.893	0.006	0.091	23.343	37.524	22.144
Z_{LMH}^{KN}	16.363	0.003	0.003	40.240	36.488	6.903
Z_{LMH}^{FL}	16.401	0.195	0.043	41.320	34.560	7.481
Z_{LMH}^{FB}	14.742	0.280	0.713	44.300	34.057	5.907
Z_{LMH}^{SN}	15.048	2.551	1.231	42.789	31.832	6.550

Table 6: Time analysis for different variants of Lagrangian matheuristic - *benchmark* instances

- `LocalB`: Local Branching (2010) (Rodríguez-Martín and José Salazar-González, 2010);
- `SACG`: Simulated Annealing/Column Generation (2013) (Yaghini et al., 2013) tested with time limits of 600 seconds (`SACG1`) and 18000 seconds (`SACG2`);
- `CEA`: Cycle-Based Evolutionary Algorithm (2016) (Paraskevopoulos et al., 2016);
- `CCL`: Combined Capacity Scaling/Local Branching (2015) (Katayama, 2015) tested with two different parameter settings (`CCL`) and (`CCL2`), which correspond, respectively, to columns "10-1000" and "20-2000" in Table 2 (Katayama, 2015);
- `ILP`: Iterative Linear Programming (2018) (Gendron et al., 2018).

The results show that the Lagrangian matheuristic provides better upper bounds on average than all the heuristics previously proposed in the literature, except `CCL` and `CCL2`. However, the Lagrangian matheuristic produces almost the same upper bounds (with an average gap of 0.24% and a maximum gap of 0.72%) in much less computational times, as we see in Table 8, which reports the CPU time spent by each method to reach its best feasible solution. The times are normalized based on the CPU type and the number of cores, using data from `www.cpubenchmark.net`. For any heuristic H that required a CPU time T_H on a computer C_H with U_H cores, the normalized CPU times $\overline{T}_H$ are computed with the formula

$$\overline{T}_H = T_H \times (P(C_H)/P(C_{\texttt{LMH}})) \times U_H,$$

where $P(C_H)$ and $P(C_{\texttt{LMH}})$ are the Passmark CPU scores of the computers used to run heuristic H and our `LMH` algorithm (Dongarra, 2014). Note that the Passmark CPU scores are not available for `CTS` and `PR` methods, and the CPU times of `MCA` are not reported in Crainic et al. (2006). Therefore, the computational times for these methods are not presented in Table 8. The results show that the Lagrangian matheuristic is competitive with the heuristics in the literature. For instance, it requires about 6 times the effort of `ILP`, but it produces solutions that are 0.34% better on average. Also, it is 5.4 times faster than `CCL2` to generate solutions that are only 0.24% away on average.

5.2 Experiments on *large* instances

All experiments on the *large* instances were performed on a more powerful computer with Intel Xeon E5-2637 CPU 3.50GHz, using CPLEX version 12.10. The parameter values are the same as those used in our experiments on the *benchmark* instances, with the

Instance	LMH	CTS	PR	MCA	CSH	IPS	LocalB	SACG	SACG2	CEA	CCL	CCL2	ILP
20,230,200VL	94295.0	-4.98	-6.48	-4.55	0.05	-0.85	-1.06	-1.06	0.01	-0.18	0.09	0.09	0.09
20,230,200FL	138118.0	-6.09	-7.15	-3.64	0.34	-2.27	-3.86	-1.19	0.20	-0.61	0.34	0.34	-0.04
20,230,200VT	97914.0	-6.98	-6.92	-4.20	-0.06	-1.53	-0.12	-0.07	0.00	-0.30	0.00	0.00	0.00
20,230,200FT	135863.0	-8.48	-8.60	-3.92	-0.20	-3.25	-3.88	-3.61	-0.89	-0.93	-0.12	0.00	-0.48
20,300,200VL	75096.5	-7.62	-4.11	-4.15	0.24	-0.30	-1.70	-1.21	0.26	-0.24	0.38	0.38	0.17
20,300,200FL	115671.0	-6.64	-6.75	-5.43	-0.10	-1.62	-3.00	-1.96	-0.66	-0.98	-0.07	0.11	-0.61
20,300,200VT	74991.0	-6.17	-5.17	-3.01	-0.41	-1.61	-1.57	-0.57	0.00	-0.60	0.00	0.00	0.00
20,300,200FT	107760.0	-6.24	-5.40	-3.17	-0.09	-2.40	-1.90	-1.81	-0.81	0.20	0.41	0.61	0.43
100,400,10FT	63764.0	-5.10	-2.37	-3.95	-15.37	-3.33	-5.63	-2.21	-2.21	-2.82	-0.69	0.02	-2.33
100,400,30FL	49115.0	-4.96	-4.50	-2.73	-5.78	-1.17	-1.54	-1.07	-0.27	-0.71	0.20	0.20	0.20
100,400,30FT	137243.0	-5.76	-3.00	-6.18	-5.15	-3.00	-3.20	-3.13	-2.75	-1.67	-0.66	0.72	-1.41
30,520,100VL	53958.0	-1.85	-1.75	-3.33	-0.24	-0.29	-0.13	-0.05	-0.05	-0.26	0.00	0.00	0.00
30,520,100FL	93985.0	-5.96	-8.59	-6.21	-0.87	-0.43	-2.42	-0.34	-0.09	-0.68	0.02	0.02	-0.09
30,520,100VT	52046.0	-1.80	-1.87	-2.82	-0.45	-0.25	-0.16	-0.66	-0.39	-0.26	0.00	0.00	0.00
30,520,100FT	97750.0	-7.95	-8.57	-4.84	-1.11	-1.16	-3.43	-2.49	-0.81	-0.11	0.37	0.66	0.35
30,700,100FL	60056.0	-4.02	-5.05	-6.16	-0.23	-1.07	-0.36	-0.87	-0.56	-0.80	0.16	0.16	0.01
30,700,100VT	45890.0	-2.47	-2.87	-3.41	-0.61	-0.34	-0.03	-0.56	-0.14	-0.42	0.04	0.04	-0.04
30,520,400VL	113201.0	-6.58	-5.49	-2.18	0.31	-0.74	-1.03	-0.88	-0.46	0.01	0.38	0.38	0.20
30,520,400FL	149943.0	-7.44	-8.78	-4.44	0.33	-2.85	-5.19	-3.93	-0.71	-0.80	0.35	0.41	0.00
30,520,400VT	114741.0	-5.97	-4.73	-5.44	0.09	-0.16	-0.43	-0.76	-0.73	-0.83	0.09	0.09	-0.05
30,520,400FT	152976.0	-9.78	-6.99	-4.73	0.15	-1.07	-10.19	-4.58	NA	-0.95	0.26	0.30	-0.58
30,700,400VL	97997.0	-8.96	-7.26	-4.73	0.03	-0.74	-4.73	-3.81	-1.35	-0.75	0.12	0.12	-0.40
30,700,400FL	135104.0	-10.25	-7.34	-6.58	0.03	-12.93	-6.58	-18.81	0.83	-1.49	0.36	0.38	-3.37
30,700,400VT	95457.4	-6.51	-6.03	-3.92	0.16	-0.74	-3.92	-1.66	-0.08	-0.70	0.22	0.22	-0.29
30,700,400FT	130809.0	-9.15	-7.80	-5.70	0.51	-0.63	-5.70	-8.17	-0.51	-1.24	0.69	0.69	-0.25
Average:		-6.31	-5.74	-4.38	-1.14	-1.79	-2.87	-2.62	-0.51	-0.73	0.12	0.24	-0.34
Maximum:		-1.80	-1.75	-2.18	0.51	-0.16	-0.03	-0.05	0.83	0.20	0.69	0.72	0.43
Minimum:		-10.25	-8.78	-6.58	-15.37	-12.93	-10.19	-18.81	-2.75	-2.82	-0.69	0.00	-3.37
# of improvements / 25:		25	25	25	14	25	25	25	18	23	4	0	13
# of draws / 25:		0	0	0	0	0	0	0	2	0	4	5	5
# of non-improvements / 25:		0	0	0	11	0	0	0	4	2	17	20	7

Table 7: Comparison between Lagrangian matheuristic and state-of-the-art heuristics - *benchmark* instances

	LMH	CSH	IPS	LocalB	SACG	SACG2	CEA	CCL	CCL2	ILP
CPU	Intel Xeon X5675 @ 3.07GHz	Intel Pentium D 940 @ 3.20GHz	Intel Xeon X3350 @ 2.66GHz	Intel Core2 Duo E4600 @ 2.40GHz	Intel Core2 Duo E6850 @ 3.00GHz	Intel Core2 Duo E6850 @ 3.00GHz	Intel Xeon E5507 @ 2.27GHz	Intel Core i7 4770 @ 3.40GHz	Intel Core i7 4770 @ 3.40GHz	Intel Core i7 4900MQ @ 2.80GHz
Passmark CPU	8507	710	3876	1383.00	1951	1951	3144	9792	9792	9060
Cores	1	2	8	2	2	2	1	4	4	1
Instance										
20,230,200VL	7663.46	73.80	2996.19	195.09	275.21	1128.36	1391.54	14090.26	30106.00	2569.34
20,230,200FL	9843.41	276.76	2518.69	195.09	275.21	504.55	1498.34	20299.49	40973.31	333.19
20,230,200VT	1135.54	87.38	2992.54	195.09	275.21	1078.36	1219.72	9948.77	13328.26	255.11
20,230,200FT	10248.30	324.46	568.62	195.09	275.21	572.89	1156.48	20893.44	52461.27	1009.01
20,300,200VL	7728.23	58.04	2923.29	195.09	275.21	1359.53	1199.58	23241.12	42161.20	340.70
20,300,200FL	10800.00	215.31	2500.47	195.09	275.21	479.78	1519.56	35769.17	39517.46	2728.90
20,300,200VT	7836.48	71.56	1414.26	195.09	275.21	1844.82	1318.36	9935.42	11787.69	150.37
20,300,200FT	4210.58	287.39	1443.42	195.09	275.21	1140.28	2891.25	17487.70	37935.45	1337.67
100,400,10FT	8387.01	9.28	2963.38	195.09	77.52	77.52	1397.78	12594.81	59757.56	1148.93
100,400,30FL	1547.10	103.74	3229.47	195.09	275.21	429.78	1479.49	3975.73	12169.84	783.01
100,400,30FT	10743.70	25.69	3236.76	195.09	275.21	553.63	4717.18	50774.29	112500.15	649.49
30,520,100VL	1151.77	4.49	794.61	195.09	275.21	629.31	1967.45	3096.33	7684.88	1933.90
30,520,100FL	8343.49	67.85	823.77	195.09	275.21	425.66	2304.02	15039.19	30308.58	1289.21
30,520,100VT	505.40	6.26	1658.47	195.09	275.21	4186.38	4759.50	9226.83	18713.34	2934.08
30,520,100FT	10794.70	33.33	2970.67	195.09	275.21	462.35	1359.34	22107.11	83229.81	1675.25
30,700,100FL	10259.90	6.26	1658.47	195.09	275.21	4186.38	4759.50	9226.83	18713.34	1650.13
30,700,100VT	9576.96	33.33	2970.67	195.09	275.21	462.35	1359.34	22107.11	83229.81	872.87
30,520,400VL	10143.10	19.11	2700.94	195.09	275.21	1831.05	4562.51	13298.33	32047.59	842.29
30,520,400FL	10800.00	7.73	1352.29	195.09	275.21	1926.92	1774.42	25595.25	37311.12	462.49
30,520,400VT	8531.79	16.16	1410.61	195.09	275.21	3139.21	4225.09	20515.89	36425.73	2561.76
30,520,400FT	10760.40	94.81	1436.13	195.09	275.21	5918.82	5664.97	27242.18	48228.16	1637.73
30,700,400VL	9651.91	435.73	2733.75	195.09	275.21	4463.43	4558.08	15593.07	83088.00	2890.34
30,700,400FL	10009.70	38.39	2263.54	195.09	275.21	8256.26	1564.90	17154.36	41117.42	1135.52
30,700,400VT	10800.00	279.41	1698.57	195.09	275.21	4286.83	2385.07	23705.69	69290.57	941.52
30,700,400FT	9972.70	79.25	809.19	195.09	275.21	8256.26	2345.23	21763.63	46253.42	1567.50
Average:	8058	106	2083	195	267	2304	2535	18587	43534	1348

Table 8: Normalized times for Lagrangian matheuristic and state-of-the-art heuristics - *benchmark* instances

exception of the time limits used in the different steps of the Lagrangian matheuristic. To set these time limits, we divide the *large* instances into two groups: the 24 instances with less than 400 commodities and the 24 instances with at least 400 commodities. For the first group of instances, we use the same time limits as for the *benchmark* instances, while for the second group of instances, we use the following time limits for each of the four steps of the Lagrangian matheuristic:

1. *Solving the unrestricted MCND.* This step is skipped for instances in the second group, since even solving the LP relaxation with CPLEX for such instances requires prohibitive computational times (see below).
2. *Solving the Lagrangian dual.* The column generation method (including calls to the slope scaling heuristic) is performed for a maximum of 6 hours.
3. *Solving restricted MCNDs.* This intensification step stops when it reaches a time limit of 5 hours.
4. *Solving the bounded unrestricted MCND.* The original problem is solved using CPLEX (giving as input the best solution and bounds found so far) for a maximum of 1 hour.

Table 9 presents a comparison of the lower bounds obtained by solving the Lagrangian duals of the different relaxations in the second step of the Lagrangian matheuristic. The LP relaxation bound is also computed with CPLEX for a time limit of 20,000 seconds. The LP relaxations of most instances could be solved within this time limit, except the four instances with 1,200 arcs and 800 commodities. For these instances, we show in column Z^{LP} the value $\max\{Z^{FW}, Z^{KN}\}$. The different columns are given the same interpretation as in Table 1.

The results confirm the effectiveness of the facility location relaxation, as it produces a better lower bound than Z^{LP} for most instances (37 out of 48), showing an improvement, on average, of 0.11% over Z^{LP}. By comparison, the forward-backward facility location and multicommodity single-node relaxations cannot improve, on average, over Z^{LP}, obtaining lower bounds worse than Z^{LP} for 24 and 23 instances, respectively. Larger time limits would be needed to obtain bound improvements, especially for instances with at least 400 commodities. Note that the flow and the knapsack relaxations are significantly faster than CPLEX to compute the LP relaxation bound, since they require, respectively, less than 70 seconds and 300 seconds, on average, while CPLEX needs more than 9,000 seconds, on average, to compute the LP relaxation.

Instance	Z^{LP}	Z^{FW}	Z^{KN}	Z^{FL}	Z^{FB}	Z^{SN}
20,300,100,1	2079230.0	0.001	0.035	-0.176	-0.172	-0.221
20,300,100,2	1651290.0	0.000	0.002	-0.068	-0.074	-0.303
20,300,100,3	2113650.0	0.000	0.007	-0.138	-0.136	-0.256
20,300,100,4	2753660.0	0.001	0.006	-0.092	-0.080	-0.348
20,300,200,5	119224.0	0.008	0.001	-0.371	-0.320	-0.199
20,300,200,6	77581.2	0.005	0.029	-0.679	-0.581	-0.409
20,300,200,7	112974.0	0.001	0.007	-0.681	-0.577	-0.406
20,300,200,8	79512.6	0.003	0.015	-0.600	-0.514	-0.400
20,300,400,9	354065.0	0.005	0.002	-0.219	-0.028	-0.244
20,300,400,10	572316.0	0.008	0.002	-0.187	0.034	-0.049
20,300,400,11	460452.0	0.001	0.000	-0.076	0.005	-0.015
20,300,400,12	796276.0	0.002	0.000	-0.098	0.037	-0.045
20,300,800,13	342150.0	0.003	0.004	-0.147	0.003	-0.028
20,300,800,14	403160.0	0.001	0.004	-0.122	0.042	-0.003
20,300,800,15	368977.0	0.222	0.220	0.099	0.237	0.218
20,300,800,16	577289.0	0.272	0.270	0.129	0.288	0.276
30,600,100,17	382161.0	0.000	0.000	-0.177	-0.192	-0.279
30,600,100,18	151875.0	0.005	0.001	-0.527	-0.515	-0.413
30,600,100,19	332958.0	0.000	0.000	-0.136	-0.092	-0.140
30,600,100,20	144411.0	0.006	0.003	-0.258	-0.213	-0.530
30,600,200,21	286318.0	0.007	0.000	-0.168	0.218	0.049
30,600,200,22	503650.0	0.010	0.000	-0.139	0.087	0.123
30,600,200,23	401435.0	0.008	0.000	-0.249	-0.065	-0.127
30,600,200,24	740939.0	0.009	0.000	-0.242	-0.200	0.024
30,600,400,25	1625830.0	0.007	0.263	-0.063	-0.060	0.019
30,600,400,26	2425060.0	0.006	0.115	-0.085	-0.082	-0.187
30,600,400,27	1501140.0	0.005	0.404	-0.041	-0.033	0.350
30,600,400,28	2179030.0	0.006	0.203	-0.037	-0.034	-0.048
30,600,800,29	924347.0	0.004	0.005	-0.185	0.018	0.435
30,600,800,30	826673.0	0.003	0.003	-0.066	0.083	0.275
30,600,800,31	549252.0	0.000	0.016	-0.135	-0.026	0.341
30,600,800,32	889032.0	0.002	0.004	-0.136	0.084	0.337
50,1200,100,33	1922620.0	0.008	0.000	-0.629	-0.615	-1.163
50,1200,100,34	964793.0	0.000	0.011	-0.578	-0.579	-1.495
50,1200,100,35	2296930.0	0.004	0.000	-0.655	-0.653	-1.057
50,1200,100,36	1117300.0	0.007	0.000	-0.627	-0.632	-1.266
50,1200,200,37	1066490.0	0.002	0.001	0.625	1.326	1.733
50,1200,200,38	167544.0	0.010	0.004	-0.199	0.618	1.102
50,1200,200,39	858612.0	0.007	0.001	0.392	1.504	1.865
50,1200,200,40	92767.1	0.001	0.062	-0.043	1.144	2.164
50,1200,400,41	259050.0	0.000	0.018	-0.067	0.251	0.315
50,1200,400,42	1442420.0	0.002	0.002	0.608	0.837	1.391
50,1200,400,43	849448.0	0.009	0.004	0.093	0.574	0.879
50,1200,400,44	1799980.0	0.006	0.002	0.579	0.679	1.532
50,1200,800,45	1002460.0	0.000	0.002	0.277	0.990	2.628
50,1200,800,46	556912.0	0.000	0.014	0.180	0.532	1.598
50,1200,800,47	1237300.0	0.000	0.001	0.215	0.805	1.568
50,1200,800,48	2121800.0	0.004	0.000	0.454	1.300	2.022
	Average:	0.014	0.036	-0.113	0.109	0.242
	Maximum:	0.272	0.404	0.625	1.504	2.628
	Minimum:	0.000	0.000	-0.681	-0.653	-1.495

Table 9: Comparison between lower bounds - *large* instances

Instance	Z_{LMH}	Z^{FW}_{LMH}	Z^{KN}_{LMH}	Z^{FL}_{LMH}	Z^{FB}_{LMH}	Z^{SN}_{LMH}
20,300,100,1	2122240.0	0.000	0.000	0.000	0.000	0.000
20,300,100,2	1664690.0	0.000	0.000	0.000	0.000	0.000
20,300,100,3	2139960.0	0.000	0.000	0.000	0.000	0.000
20,300,100,4	2840340.0	0.045	0.000	0.000	0.000	0.000
20,300,200,5	125280.0	0.144	0.234	0.000	0.098	0.222
20,300,200,6	81842.0	0.127	0.000	0.992	1.084	0.310
20,300,200,7	119601.0	0.000	0.000	0.201	0.391	0.000
20,300,200,8	83528.0	0.307	0.000	0.091	0.127	0.204
20,300,400,9	373047.0	0.319	0.103	0.000	0.917	0.173
20,300,400,10	599919.0	0.619	0.000	0.016	0.277	0.336
20,300,400,11	473093.0	0.096	0.465	0.000	0.174	0.122
20,300,400,12	824995.0	0.234	0.808	0.000	0.136	0.293
20,300,800,13	352025.0	0.387	0.066	0.185	0.000	0.508
20,300,800,14	413726.0	0.291	0.622	0.000	0.244	0.389
20,300,800,15	377037.0	0.906	0.000	0.153	0.374	0.319
20,300,800,16	593729.0	0.334	0.929	0.304	0.000	0.683
30,600,100,17	404645.0	0.259	0.000	0.747	1.018	0.922
30,600,100,18	158213.0	0.000	0.222	0.134	0.132	0.324
30,600,100,19	360534.0	0.891	0.001	0.134	0.000	0.079
30,600,100,20	149582.0	0.158	0.000	0.262	0.262	0.262
30,600,200,21	307998.0	0.713	0.979	0.440	0.445	0.000
30,600,200,22	541640.0	1.569	0.713	0.000	0.763	1.431
30,600,200,23	428292.0	0.306	0.247	0.617	0.539	0.000
30,600,200,24	789781.0	0.255	0.000	0.578	1.143	2.242
30,600,400,25	1671240.0	0.218	0.000	0.079	0.023	0.199
30,600,400,26	2491800.0	0.845	1.017	0.000	0.178	0.370
30,600,400,27	1545650.0	0.000	0.388	0.177	0.209	1.193
30,600,400,28	2240590.0	0.378	0.012	0.365	0.000	0.472
30,600,800,29	958991.0	1.113	1.620	0.000	0.012	0.554
30,600,800,30	851300.0	0.977	0.435	0.391	0.000	1.371
30,600,800,31	563363.0	0.538	1.193	0.000	0.749	0.595
30,600,800,32	913800.0	2.491	1.758	0.000	0.485	1.752
50,1200,100,33	2080910.0	0.000	1.085	1.500	2.338	1.071
50,1200,100,34	1056360.0	0.235	0.041	0.000	0.057	0.866
50,1200,100,35	2482000.0	1.070	0.000	0.237	0.060	1.418
50,1200,100,36	1207520.0	0.965	0.000	1.035	1.133	1.314
50,1200,200,37	1182460.0	3.905	0.000	0.274	1.691	0.217
50,1200,200,38	180607.0	0.038	0.477	0.409	0.035	0.000
50,1200,200,39	964185.0	0.289	0.000	0.465	1.963	1.426
50,1200,200,40	98770.1	1.069	0.047	0.288	0.000	1.306
50,1200,400,41	275946.0	0.000	1.061	1.152	1.152	0.602
50,1200,400,42	1556500.0	1.758	4.218	0.249	0.000	1.626
50,1200,400,43	908157.0	1.738	1.881	0.958	0.000	2.755
50,1200,400,44	1960060.0	0.170	2.874	0.000	0.000	0.378
50,1200,800,45	1070100.0	0.381	0.000	0.549	0.549	0.513
50,1200,800,46	581820.0	0.000	1.216	1.144	1.144	1.134
50,1200,800,47	1299950.0	0.364	0.318	0.000	0.000	0.000
50,1200,800,48	2231220.0	0.000	0.000	0.579	0.649	0.231
	Average:	0.552	0.522	0.306	0.428	0.629

Table 10: Comparison of upper bounds for different variants of Lagrangian matheuristic - *large* instances

Table 10 compares the upper bounds found by the Lagrangian matheuristic for each of the five different relaxations. Column Z_{LMH} shows the best upper bound obtained with the five variants of the Lagrangian matheuristic. The other columns show the gaps between Z^R_{LMH}, the upper bound obtained by the Lagrangian matheuristic used with relaxation R, and Z_{LMH}, computed as $100 \times (Z^R_{LMH} - Z_{LMH})/Z_{LMH}$. The results show that all variants of the Lagrangian matheuristic are competitive between one another, with the facility location relaxation producing slightly better upper bounds on average than the others. The best upper bound for each instance found in this way also allows us to compute a gap with respect to the lower bounds. For the facility location relaxation, we obtain a gap of 4.9% on average between the lower bound and the best upper bound, which is quite effective for problem instances of this size.

6 Conclusion

We have introduced three node-based Lagrangian relaxations for the MCND. We have shown through theoretical and experimental results that these relaxations provide better lower bounds than the classical flow and knapsack relaxations. The three relaxations define a hierarchy of lower bounds: the multicommodity single-node flow relaxation dominates the forward-backward facility location relaxation, which itself dominates the facility location relaxation. This last relaxation, in spite of being the weakest of the three, provides a good tradeoff between the quality of the lower bound and the computational efficiency of the column generation method to solve the Lagrangian dual. We have developed and tested a Lagrangian matheuristic to compute effective upper bounds. Our computational experiments on a set of benchmark instances show that the Lagrangian matheuristic based on the facility location relaxation is competitive with the state-of-the-art heuristics for the MCND.

The hierarchical nature of the three node-based Lagrangian relaxations paves the way to a dual-ascent method that would perform them sequentially, the optimal Lagrange multipliers of one relaxation being provided as input to the stronger relaxation. Such a method could serve as a basis for an exact algorithm, the development of which is a challenging research avenue. In particular, an exact algorithm could integrate specialized algorithms for the CFLP and the MSNF, to solve the Lagrangian subproblems, as well as for the MMCF and for restricted MCNDs of small- to medium-scale, to derive feasible solutions in an efficient way.

Acknowledgments

Partial funding for this work was provided by the Canadian Natural Sciences and Engineering Research Council (NSERC) through its Discovery Grants Program. The BTT code used in this paper has been partly developed under the auspices of the European Union's Horizon 2020 research and innovation program under grant agreement No 773897 *plan4res*. While working on this paper, Teodor Gabriel Crainic was Adjunct Professor, Department of Computer Science and Operations Research, Université de Montréal, Canada. The authors are grateful to Serge Bisaillon for his invaluable assistance with algorithm coding and numerical experiments.

References

Amor, H. M. B., Desrosiers, J., and Frangioni, A. (2009). On the choice of explicit stabilizing terms in column generation. *Discrete Applied Mathematics*, 157(6):1167 – 1184.

Atamtürk, A. (2001). Flow pack facets of the single node fixed-charge flow polytope. *Operations Research Letters*, 29(3):107 – 114.

Babonneau, F., du Merle, O., and Vial, J.-P. (2006). Solving large-scale linear multicommodity flow problems with an active set strategy and proximal-accpm. *Operations Research*, 54(1):184–197.

Chouman, M. and Crainic, T. G. (2010). A MIP-tabu search hybrid framework for multicommodity capacitated fixed-charge network design. *Publication CIRRELT 2010-31*. Interuniversity Research Centre on Enterprise Networks, Logistics and Transportation.

Chouman, M., Crainic, T. G., and Gendron, B. (2016). Commodity representations and cut-set-based inequalities for multicommodity capacitated fixed-charge network design. *Transportation Science*, 51(2):650–667.

Chouman, M., Crainic, T. G., and Gendron, B. (2017). The impact of filtering in a branch-and-cut algorithm for multicommodity capacitated fixed charge network design. *EURO Journal on Computational Optimization*, 6(2):1–42.

Costa, A. M., Cordeau, J.-F., and Gendron, B. (2009). Benders, metric and cutset inequalities for multicommodity capacitated network design. *Computational Optimization and Applications*, 42(3):371–392.

Costa, A. M., Cordeau, J.-F., Gendron, B., and Laporte, G. (2012). Accelerating Benders decomposition with heuristicmaster problem solutions. *Pesquisa Operacional*, 32(1):03–20.

Crainic, T. G., Frangioni, A., and Gendron, B. (2001). Bundle-based relaxation methods for multicommodity capacitated fixed charge network design. *Discrete Applied Mathematics*, 112(1):73–99.

Crainic, T. G. and Gendreau, M. (2002). Cooperative parallel tabu search for capacitated network design. *Journal of Heuristics*, 8(6):601–627.

Crainic, T. G. and Gendreau, M. (2007). A scatter search heuristic for the fixed-charge capacitated network design problem. In Doerner, K., Gendreau, M., Greistorfer, P., Gutjahr, W. J., Hartl, R. F., and Reismann, M., editors, *Metaheuristics - progress in complex systems optimization*, pages 25–40. Springer, Dordrecht.

Crainic, T. G., Gendreau, M., and Farvolden, J. M. (2000). A simplex-based tabu search method for capacitated network design. *INFORMS Journal on Computing*, 12(3):223–236.

Crainic, T. G., Gendron, B., and Hernu, G. (2004). A slope scaling/Lagrangean perturbation heuristic with long-term memory for multicommodity capacitated fixed-charge network design. *Journal of Heuristics*, 10(5):525–545.

Crainic, T. G., Li, Y., and Toulouse, M. (2006). A first multilevel cooperative algorithm for capacitated multicommodity network design. *Computers & Operations Research*, 33(9):2602–2622.

Dongarra, J. J. (2014). Performance of various computers using standard linear equations software. Technical report, University of Manchester.

Fischetti, M., Ljubić, I., and Sinnl, M. (2016). Benders decomposition without separability: A computational study for capacitated facility location problems. *European Journal of Operational Research*, 253(3):557 – 569.

Frangioni, A., Gendron, B., and Gorgone, E. (2017). On the computational efficiency of subgradient methods: a case study with lagrangian bounds. *Mathematical Programming Computation*, pages 1–32.

Frangioni, A. and Gorgone, E. (2014). Bundle methods for sum-functions with "easy" components: applications to multicommodity network design. *Mathematical Programming*, 145(1-2):133–161.

Gendron, B. (2018). Revisiting Lagrangian relaxation for network design. *Discrete Applied Mathematics*.

Gendron, B. and Crainic, T. G. (1994). Relaxations for multicommodity capacitated network design problems. *Publication CRT-965*. Centre for Research on Transportation.

Gendron, B., Crainic, T. G., and Frangioni, A. (1999). Multicommodity capacitated network design. In Soriano, P. and Sansò, B., editors, *Telecommunications network planning*, pages 1–19. Kluwer Academics Publisher, Dordrecht.

Gendron, B. and Gouveia, L. (2016). Reformulations by discretization for piecewise linear integer multicommodity network flow problems. *Transportation Science*, 51(2):629–649.

Gendron, B., Hanafi, S., and Todosijević, R. (2018). Matheuristics based on iterative linear programming and slope scaling for multicommodity capacitated fixed charge network design. *European Journal of Operational Research*, 268(1):70–81.

Gendron, B. and Larose, M. (2014). Branch-and-price-and-cut for large-scale multicommodity capacitated fixed-charge network design. *EURO Journal on Computational Optimization*, 2(1-2):55–75.

Geoffrion, A. (1974). Lagrangean relaxation for integer programming. *Mathematical Programming Study*, 2:82–114.

Ghamlouche, I., Crainic, T. G., and Gendreau, M. (2003). Cycle-based neighbourhoods for fixed-charge capacitated multicommodity network design. *Operations Research*, 51(4):655–667.

Ghamlouche, I., Crainic, T. G., and Gendreau, M. (2004). Path relinking, cycle-based neighbourhoods and capacitated multicommodity network design. *Annals of Operations Research*, 131(1):109–133.

Guignard, M. and Kim, S. (1987). Lagrangean decomposition: A model yielding stronger lagrangean bounds. *Mathematical Programming*, 39(2):215–228.

Hewitt, M., Nemhauser, G. L., and Savelsbergh, M. W. (2010). Combining exact and heuristic approaches for the capacitated fixed-charge network flow problem. *INFORMS Journal on Computing*, 22(2):314–325.

Holmberg, K. and Yuan, D. (2000). A Lagrangian heuristic based branch-and-bound approach for the capacitated network design problem. *Operations Research*, 48(3):461–481.

Katayama, N. (2015). A combined capacity scaling and local branching approach for capacitated multi-commodity network design problem. *Far East Journal of Applied Mathematics*, 92(1):1.

Katayama, N., Chen, M., and Kubo, M. (2009). A capacity scaling heuristic for the multicommodity capacitated network design problem. *Journal of Computational and Applied Mathematics*, 232(1):90–101.

Kliewer, G. and Timajev, L. (2005). Relax-and-cut for capacitated network design. In *Proceedings of Algorithms-ESA 2005: 13th Annual European Symposium on Algorithms*, pages 47–58. Lecture Notes in Computer Science 3369.

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

Paraskevopoulos, D. C., Bektaş, T., Crainic, T. G., and Potts, C. N. (2016). A cycle-based evolutionary algorithm for the fixed-charge capacitated multi-commodity network design problem. *European Journal of Operational Research*, 253(2):265–279.

Rodríguez-Martín, I. and José Salazar-González, J. (2010). A local branching heuristic for the capacitated fixed-charge network design problem. *Computers & Operations Research*, 37(3):575–581.

Sellmann, M., Kliewer, G., and Koberstein, A. (2002). Lagrangian cardinality cuts and variable fixing for capacitated network design. In *Proceedings of Algorithms-ESA 2002: 10th Annual European Symposium on Algorithms*, pages 845–858. Lecture Notes in Computer Science 2461.

Yaghini, M., Rahbar, M., and Karimi, M. (2013). A hybrid simulated annealing and column generation approach for capacitated multicommodity network design. *Journal of the Operational Research Society*, 64(7):1010–1020.