CIRRELT-2019-38

A Lagrangian Heuristic for the Multicommodity Capacitated Location Problem with Balancing

Anis Kadri
Oumar Koné
Bernard Gendron

September 2019

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

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

A Lagrangian Heuristic for the Multicommodity Capacitated Location Problem with Balancing Requirements

Anis Kadri[1, *], Oumar Koné[2], Bernard Gendron[1]

[1] Interuniversity Research Centre on Enterprise Networks, Logistics and Transportation (CIRRELT), and Department of Computer Science and Operations Research, Université de Montréal

[2] Laboratoire de mathématiques et informatique, UFR des Sciences Fondamentales et Appliquées, Université Nangui Abrogoua, Côte d'ivoire

Abstract. We consider the multicommodity capacitated location problem with balancing requirements. This problem has applications in transportation and was introduced in the context of planning the movements of containers between ports and terminals, identified as depots in the problem formulation. We propose a Lagrangian relaxation based on a reformulation of the problem that allows the subproblem to be decomposed by depot, giving rise to a series of continuous knapsack subproblems. We optimize the Lagrangian dual by a bundle method and we develop a Lagrangian heuristic based on slope scaling. Numerical results are presented on a set of large-scale benchmark instances. We obtain good quality solutions within short computing times using the Lagrangian heuristic approach.

Keywords: Multicommodity capacitated location with balancing requirements, Lagrangian relaxation, Slope scaling, Bundle method.

Acknowledgments. Funding for this project has been provided by the Natural Sciences and Engineering Research Council of Canada (NSERC), through its Discovery Grant program. We also gratefully acknowledge the support of Fonds de recherche du Québec through their infrastructure grants.

---

* Corresponding author: Anis.Kadri@cirrelt.ca

Dépôt légal – Bibliothèque et Archives nationales du Québec
Bibliothèque et Archives Canada, 2019

1. Introduction

This paper discusses the multicommodity capacitated location problem with balancing requirements (*MCLB*) introduced in Crainic et al. (1989) in the context of planning the movements of containers between ports and terminals. This problem is formulated as a mixed-integer programming (*MIP*) model. It aims to locate the depots (ports and terminals) to meet the demands of customers for several commodities (types of containers), while minimizing the total cost, which includes the cost of opening and operating depots, the cost of transportation between customers and depots, and the cost of the inter-depot movements. The *MCLB* is NP-hard, as it generalizes the well known NP-hard uncapacitated facility location problem (Krarup and Pruzan 1983). Different exact and heuristic approaches have been proposed for the version of the problem without capacity (Crainic and Delorme 1993, Crainic et al. 1993a,b, Gendron and Crainic 1995, 1997, Gendron et al. 1999). A more challenging capacitated version, where each depot has a fixed and finite capacity, is presented in Gendron et al. (2003a) and Gendron et al. (2003b). In Gendron et al. (2003a), the authors proposed a tabu search heuristic combined with a slope scaling procedure for solving the *MCLB*, while in Gendron et al. (2003b), the authors developed a parallel hybrid heuristic.

In this paper, we propose a Lagrangian relaxation approach, based on the bundle method (Frangioni 2005, Bonnans et al. 2006), to compute tight lower bounds on the optimal value of the *MCLB*. We show that the corresponding Lagrangian dual provides the same bound as the linear programming (LP) relaxation. We propose a Lagrangian heuristic based on slope scaling (see Gendron and Gouveia (2017) for a similar approach) that finds good quality solutions in short computing times. We test our method on large-scale instances that cannot be solved in reasonable times by state-of-the-art MIP solvers. On this set of instances, we show that the bundle method is faster than a state-of-the-art LP solver, while providing accurate lower bounds. In addition, the slope scaling method is robust and much faster than solving the root node by a state-of-the-art MIP solver. In contrast with the approaches in Gendron et al. (2003a) and Gendron et al. (2003b) that only focus on finding feasible solutions, we take into account the lower bound as well, thus producing provably effective feasible solutions.

The remainder of the paper is structured as follows. Section 2 focuses on the presentation and the mathematical formulation of the problem. Section 3 presents the application of Lagrangian relaxation to this problem and the formulation of the Lagrangian subproblem. In Section 4, the Lagrangian heuristic based on slope scaling is presented. In Section 5, we present our experiments and results. Finally, conclusions and future work are discussed in Section 6.

2. Problem formulation

We consider a directed network $G=(N,A)$, where N is the set of nodes and A is the set of arcs. In our problem, N represents customers and depots, while A is the set of transportation links that connect either two depots or a customer and a depot. The commodities moving through the network are represented by set K. We distinguish three subsets of the set of nodes:

- *Origins* O: Each origin i supplies a quantity $o_i^k \geq 0$ of commodity k.
- *Destinations* D: Each destination i requests a quantity $d_i^k \geq 0$ of commodity k.
- *Depots* T: Each depot is a transshipment node for the commodities transported between the origins and the destinations.

We assume that the total supply equals the total demand for each commodity:

$$\sum_{i\in O} o_i^k = \sum_{i\in D} d_i^k \equiv M^k,\ k\in K.$$

The set of arcs can be partitioned into three subsets:

- *Origin-to-depot arcs* $A_{OT}=\{(i,j)\in A : i\in O, j\in T\}$;
- *Depot-to-depot arcs* $A_{TT}=\{(i,j)\in A : i\in T, j\in T\}$;
- *Depot-to-destination arcs* $A_{TD}=\{(i,j)\in A : i\in T, j\in D\}$.

The problem consists in finding an optimal distribution of flows moving through the network to satisfy the supplies at origins and the demands at destinations. The objective considers the costs of transporting the commodities between depots, as well as between depots and customers, and the fixed costs of opening the depots. For each unit of commodity $k\in K$, x_{ij}^k, moving from node $i\in N$ to node $j\in N$, is associated a transportation cost $c_{ij}^k \geq 0$. For each depot $j\in T$ having a fixed capacity $q_j>0$, a fixed cost $f_j\geq 0$ is incurred if the depot is opened. The volume of one unit of commodity $k\in K$ is denoted $v^k>0$. A binary location variable y_j takes value 1 if depot $j\in T$ is opened, and value 0, otherwise. The multicommodity capacitated location problem with balancing requirements ($MCLB$) is then formulated as follows:

$$\min \sum_{j\in T} f_j y_j + \sum_{k\in K}\left(\sum_{(i,j)\in A_{OT}} c_{ij}^k x_{ij}^k + \sum_{(l,j)\in A_{TT}} c_{lj}^k x_{lj}^k + \sum_{(j,i)\in A_{TD}} c_{ji}^k x_{ji}^k\right) \tag{1}$$

$$\sum_{j\in T_i^+} x_{ij}^k = o_i^k, \qquad i\in O, k\in K, \tag{2}$$

$$\sum_{j\in T_i^-} x_{ji}^k = d_i^k, \qquad i\in D, k\in K, \tag{3}$$

$$\sum_{i\in O_j} x_{ij}^k + \sum_{l\in T_j^-} x_{lj}^k - \sum_{l\in T_j^+} x_{jl}^k - \sum_{i\in D_j} x_{ji}^k = 0, \qquad j\in T, k\in K, \tag{4}$$

$$\sum_{k\in K} v^k \left(\sum_{i\in O_j} x_{ij}^k + \sum_{l\in T_j^-} x_{lj}^k \right) \le q_j y_j, \qquad j\in T, \tag{5}$$

$$\sum_{i\in O_j} x_{ij}^k + \sum_{l\in T_j^-} x_{lj}^k \le M^k y_j, \qquad j\in T, k\in K, \tag{6}$$

$$x_{ij}^k \le o_i^k y_j, \qquad j\in T, i\in O_j, k\in K, \tag{7}$$

$$x_{ji}^k \le d_i^k y_j, \qquad j\in T, i\in D_j, k\in K, \tag{8}$$

$$x_{ij}^k \ge 0, \qquad (i,j)\in A, k\in K, \tag{9}$$

$$y_j \in \{0,1\}, \qquad j\in T, \tag{10}$$

where $O_j = \{i\in O : (i,j)\in A\}$, $D_j = \{i\in D : (j,i)\in A\}$, $j\in T$, $T_i^+ = \{j\in T : (i,j)\in A\}$ and $T_i^- = \{j\in T : (j,i)\in A\}$, $i\in N$.

The objective function (1) minimizes the total cost, i.e., the transportation costs and the fixed costs of opening the depots. Constraints (2) and (3) ensure the satisfaction of supplies and demands. Constraints (4) ensure that the sum of the incoming flow is equal to the sum of the outgoing flow for each depot and each commodity. Constraints (2), (3) and (4) are the *flow conservation equations*. Constraints (5) ensure that the capacity at each depot is not exceeded, while at the same time forbidding any flow to circulate through a closed depot. The same is achieved by constraints (6), which are therefore redundant, but are added to improve the *LP* relaxation bound. Note that constraints (6) do not appear in the formulation of the *MCLB* presented in Gendron et al. (2003a). Finally, constraints (7) and (8) are linking constraints that forbid customer-depot and depot-customer flows going through a closed depot. These constraints are also redundant, but improve the *LP* relaxation bound.

This model includes as special cases several problems considered in the literature. The uncapacitated version of the problem is simply obtained by setting $q_j \ge \sum_{k\in K} v^k M^k$, $j\in T$, in which case constraints (5) can be removed. If, in addition, the interdepot costs satisfy the triangle inequality, i.e., $c_{lj}^k \le c_{lh}^k + c_{hj}^k$, $(l,j)\in A_{TT}$, $h\in T_l^+ \bigcap T_j^-$, $k\in K$, constraints (6) can be removed as well, and we obtain the formulation introduced in Crainic et al. (1989). The case where each commodity $k\in K$ has a single origin $O(k)$ and a single destination $D(k)$ with a demand $w^k > 0$ is obtained by setting

$$o_i^k = \begin{cases} w^k, & \text{if } i = O(k), \\ 0, & \text{otherwise,} \end{cases}$$

$$d_i^k = \begin{cases} w^k, & \text{if } i = D(k), \\ 0, & \text{otherwise.} \end{cases}$$

In that case, if we assume the triangle inequality holds, no more than two intermediate depots can be used on any path connecting $O(k)$ to $D(k)$ for each $k\in K$. The variant of the capacitated hub location problem introduced in Campbell (1994) can thus be formulated as a special case of our model for the ($MCLB$), the depots representing the hubs. Indeed, Campbell (1994) considers a problem where the capacity at each hub limits the flow coming into the hub from the origins and from other hubs, in the same way as in constraints (5). In addition, the problem defined in Campbell (1994) allows several paths to be used to satisfy the demand w^k between $O(k)$ and $D(k)$, as in our arc-based model, which is otherwise quite different than the path-based formulation of Campbell (1994). Note that many other variants of the capacitated hub location problem have since been proposed, see, e.g., Marin (2005) and Contreras et al. (2012).

3. Lagrangian relaxation

In this section, we present a Lagrangian relaxation method that provides a lower bound on the optimal value of the *MCLB*. We propose to relax the flow conservation equations. The advantage of this approach is that it yields a simple Lagrangian subproblem that decomposes by depot and can be efficiently solved. However, if we perform this Lagrangian relaxation on our model (1)-(9), only the incoming interdepot flow variables are considered for each depot. To overcome this issue, one could add the equivalent of constraints (5) and (6) that make use of outgoing interdepot flow variables, but then we would obtain a Lagrangian subproblem that is no more decomposable by depot because the inter-depot flow variables would be shared by two depots. In order to obtain a Lagrangian subproblem that is decomposable by depot, we add copy constraints on the inter-depot flow variables, as in Lagrangian decomposition (Guignard and Kim 1987):

$$w_{jl}^k = x_{jl}^k, \quad j \in T, l \in T_j^+, k \in K. \tag{11}$$

The duplicated variables are simply bounded as follows:

$$\sum_{i \in D_j} x_{ji}^k + \sum_{l \in T_j^+} w_{jl}^k \leq M^k y_j, \quad j \in T, k \in K, \tag{12}$$

$$w_{jl}^k \geq 0, \quad j \in T, l \in T_j^+, k \in K. \tag{13}$$

We also add the redundant capacity constraints:

$$\sum_{k \in K} v^k \left(\sum_{i \in D_j} x_{ji}^k + \sum_{l \in T_j^+} w_{jl}^k \right) \leq q_j y_j, \quad j \in T. \tag{14}$$

We obtain the following reformulation of the *MCLB*:

$$\min \sum_{j \in T} f_j y_j + \sum_{k \in K} \left(\sum_{(i,j) \in A_{OT}} c_{ij}^k x_{ij}^k + \sum_{(l,j) \in A_{TT}} c_{lj}^k x_{lj}^k + \sum_{(j,i) \in A_{TD}} c_{ji}^k x_{ji}^k \right) \tag{15}$$

subject to constraints (2)-(14).

The Lagrangian relaxation consists in relaxing the copy constraints (11) by associating with them a vector of Lagrange multipliers γ and the flow conservation equations (2), (3) and (4) by associating with each of these constraints a vector of Lagrange multipliers δ. We obtain the following Lagrangian subproblem:

$$Z(\delta,\gamma) = \min \sum_{k\in K}\Bigg(\sum_{(i,j)\in A_{OT}} (c_{ij}^k - \delta_i^k + \delta_j^k)x_{ij}^k + \sum_{(l,j)\in A_{TT}} (c_{lj}^k - \delta_l^k + \delta_j^k + \gamma_{lj}^k)x_{lj}^k + \sum_{(j,i)\in A_{TD}} (c_{ji}^k - \delta_j^k + \delta_i^k)x_{ji}^k - \sum_{(j,l)\in A_{TT}} \gamma_{jl}^k w_{jl}^k - \sum_{i\in D}\delta_i^k d_i^k + \sum_{i\in O}\delta_i^k o_i^k \Bigg) + \sum_{j\in T} f_j y_j \quad (16)$$

subject to constraints (5)-(10) and (12)-(14).

The objective function (16) of the Lagrangian subproblem can also be written as follows:

$$Z(\delta,\gamma) = \min \sum_{j\in T}\Bigg\{ \sum_{k\in K}\Bigg(\sum_{i\in O_j}(c_{ij}^k - \delta_i^k + \delta_j^k)x_{ij}^k + \sum_{l\in T_j^-}(c_{lj}^k - \delta_l^k + \delta_j^k + \gamma_{lj}^k)x_{lj}^k + \sum_{i\in D_j}(c_{ji}^k - \delta_j^k + \delta_i^k)x_{ji}^k - \sum_{l\in T_j^+}\gamma_{jl}^k w_{jl}^k \Bigg) + f_j y_j \Bigg\} + \sum_{k\in K}\Bigg(\sum_{i\in O}\delta_i^k o_i^k - \sum_{i\in D}\delta_i^k d_i^k \Bigg) \quad (17)$$

The Lagrangian subproblem decomposes by depot. For each depot $j \in T$, it is solved by considering the two possible alternatives, either $y_j = 0$, for which the optimal value is 0, or $y_j = 1$, in which case the Lagrangian subproblem reduces to a continuous knapsack problem, with optimal value $f_j + g_j(\delta,\gamma)$, where

$$g_j(\delta,\gamma) \;=\; \min \sum_{k\in K}\Bigg(\sum_{i\in O_j} C_{ij}^k(\delta)x_{ij}^k \;+\; \sum_{l\in T_j^-} C_{lj}^k(\delta,\gamma)^k x_{lj}^k \;+\; \sum_{i\in D_j} C_{ji}^k(\delta)x_{ji}^k \;+\; \sum_{l\in T_j^+} C_{jl}^k(\gamma)w_{jl}^k \Bigg) \quad (18)$$

$$\sum_{k\in K} v^k \Bigg(\sum_{i\in O_j} x_{ij}^k + \sum_{l\in T_j^-} x_{lj}^k \Bigg) \le q_j, \quad (19)$$

$$\sum_{k\in K} v^k \Bigg(\sum_{i\in D_j} x_{ji}^k + \sum_{l\in T_j^+} w_{jl}^k \Bigg) \le q_j, \quad (20)$$

$$\sum_{i\in D_j} x_{ji}^k + \sum_{l\in T_j^+} w_{jl}^k \le M^k, \qquad k \in K, \quad (21)$$

$$\sum_{i\in O_j} x_{ij}^k + \sum_{l\in T_j^-} x_{lj}^k \le M^k, \qquad k \in K, \quad (22)$$

$$0 \le x_{ij}^k \le o_i^k, \qquad i \in O_j, k \in K, \quad (23)$$

$$0 \le x_{ji}^k \le d_i^k, \qquad i \in D_j, k \in K, \quad (24)$$

$$w_{jl}^k \geq 0, \qquad l \in T_j^+, k \in K, \tag{25}$$

$$x_{lj}^k \geq 0, \qquad l \in T_j^-, k \in K, \tag{26}$$

where

$$\begin{aligned} C_{ij}^k(\delta) &= c_{ij}^k - \delta_i^k + \delta_j^k, & i \in O_j,\, k \in K,\\ C_{lj}^k(\delta,\gamma) &= c_{lj}^k - \delta_l^k + \delta_j^k + \gamma_{lj}^k, & l \in T_j^-,\, k \in K,\\ C_{ji}^k(\delta) &= c_{ji}^k - \delta_j^k + \delta_i^k, & i \in D_j,\, k \in K,\\ C_{jl}^k(\gamma) &= \gamma_{jl}^k, & l \in T_j^+,\, k \in K. \end{aligned}$$

Among these two alternatives ($y_j = 0$ and $y_j = 1$), we choose the least cost one.

The values for the Lagrange multipliers (δ, γ) are obtained by solving the Lagrangian dual:

$$Z(LD) = \max_{(\delta,\gamma)} Z(\delta, \gamma). \tag{27}$$

We use the bundle method (Frangioni 2005) to solve the Lagrangian dual. Note that the Lagrangian subproblem has the integrality property (Geoffrion 1974). Thus, the lower bound provided by the Lagrangian dual is equal to the LP relaxation bound $Z(LP)$:

$$Z(LD) = Z(LP). \tag{28}$$

Even though the Lagrangian dual and the LP relaxation have the same strength, solving the LP relaxation with a state-of-the-art solver is computationally more expensive for large-scale instances, as we demonstrate with our computational experiments (see Section 5). A key advantage of using Lagrangian relaxation is that the subproblem is decomposable by depot and is easy to calculate. We use the solutions to the subproblem in the Lagrangian heuristic described in the next section.

4. Lagrangian heuristic

Upper bounds on the optimal value of the $MCLB$ are obtained by a Lagrangian heuristic that uses a slope scaling procedure (Kim and Pardalos 2000, Gendron et al. 2003b, Gendron and Gouveia 2017), which is repetitively called every I iterations of the bundle method, where I is a parameter (in our experiments, we use $I = 10$).

At each slope scaling iteration $t \geq 0$, we solve an associated multicommodity minimum cost network flow problem ($MMCF$) with modified linear costs $\bar{c}(t)$, defined as follows:

$$\min \sum_{k \in K} \left(\sum_{(i,j) \in A_{OT}} \bar{c}_{ij}^k(t) x_{ij}^k + \sum_{(l,j) \in A_{TT}} \bar{c}_{lj}^k(t) x_{lj}^k + \sum_{(j,i) \in A_{TD}} \bar{c}_{ji}^k(t) x_{ji}^k \right) \tag{29}$$

subject to constraints (2)-(4), (9) and

$$\sum_{k \in K} v^k \left(\sum_{i \in O_j} x_{ij}^k + \sum_{l \in T_j^-} x_{lj}^k \right) \leq q_j, \qquad j \in T. \tag{30}$$

Given a feasible solution $\tilde{x}$ to this *MMCF*, a feasible solution $(\tilde{x}, \tilde{y})$ to the *MCLB* is obtained as follows:

$$\tilde{y}_j = \begin{cases} 1, & \text{if } \left(\sum_{i \in O_j} \tilde{x}^k_{ij} + \sum_{l \in T^-_j} \tilde{x}^k_{lj} \right) > 0, \\ 0, & \text{otherwise.} \end{cases} \quad j \in T. \tag{31}$$

An upper bound on the optimal value of the *MCLB* is then computed as:

$$Z(\tilde{x}, \tilde{y}) = \sum_{j \in T} f_j \tilde{y}_j + \sum_{k \in K} \left(\sum_{(i,j) \in A_{OT}} c^k_{ij} \tilde{x}^k_{ij} + \sum_{(l,j) \in A_{TT}} c^k_{lj} \tilde{x}^k_{lj} + \sum_{(j,i) \in A_{TD}} c^k_{ji} \tilde{x}^k_{ji} \right). \tag{32}$$

To define the initial modified linear costs $\overline{c}(0)$, we use the Lagrangian subproblem solution $\hat{y}$ obtained at the current iteration of the bundle method. We define the modified linear costs at iteration $t = 0$ as follows:

$$\overline{c}^k_{ij}(0) = (c^k_{ij} + v^k \alpha_j \tfrac{f_j}{q_j})(1 + L(1 - \hat{y}_j)), \qquad (i,j) \in A_{OT}, k \in K,$$
$$\overline{c}^k_{ji}(0) = (c^k_{ji} + v^k (1 - \alpha_j) \tfrac{f_j}{q_j})(1 + L(1 - \hat{y}_j)), \qquad (j,i) \in A_{TD}, k \in K,$$
$$\overline{c}^k_{lj}(0) = (c^k_{lj} + v^k (\alpha_j \tfrac{f_j}{q_j} + (1 - \alpha_l) \tfrac{f_l}{q_l}))(1 + L \max\{1 - \hat{y}_j, 1 - \hat{y}_l\}), \qquad (l,j) \in A_{TT}, k \in K,$$

where $\alpha_j = \xi_j / (\xi_j + \Delta_j)$ with $\xi_j = \sum_{k \in K} v^k \sum_{i \in O_j} o^k_i$ and $\Delta_j = \sum_{k \in K} v^k \sum_{i \in D_j} d^k_i$, for each $j \in T$ and L is a number large enough to penalize the costs associated to closed depots (in our experiments, we set $L = 100$). In these formulas, ξ_j (Δ_j) is the maximum volume that might transit through depot j from all supply (demand) customers adjacent to j. The values α_j and $(1 - \alpha_j)$ thus approximate the fractions of the maximum volume in-transit at depot j that can be imputed to supply and demand customers, respectively. For any arc between a customer and a depot j, if j is open in the Lagrangian subproblem solution $\hat{y}$ (i.e., $\hat{y}_j = 1$), the costs on that arc then correspond to the initial linear costs used in the slope scaling method introduced in Gendron et al. (2003b), while if j is closed in solution $\hat{y}$ (i.e., $\hat{y}_j = 0$), the initial linear costs on that arc are multiplied by $1 + L$, thus discouraging the use of that arc when solving the *MMCF*. For any arc between two depots j and l, if both depots are open in solution $\hat{y}$ (i.e., $\hat{y}_j = \hat{y}_l = 1$), the initial linear costs on that arc are the same as those used in Gendron et al. (2003b), while if at least one depot is closed in solution $\hat{y}$ (i.e., $\hat{y}_j = 0$ or $\hat{y}_l = 0$), the initial linear costs on that arc are multiplied by $1 + L$. Note that such an arc will be used if needed to satisfy the capacity constraints, and its linear costs will then be adjusted accordingly, as we see next.

The modified linear costs $\overline{c}(t)$ at each iteration $t \geq 1$ are computed by using the total volume $\tilde{X}_j(t-1)$ at each depot $j \in T$ in the solution $\widetilde{x}(t-1)$ identified at the previous iteration:

$$\tilde{X}_j(t-1) = \sum_{k \in K} v^k \left(\sum_{i \in O_j} \tilde{x}^k_{ij}(t-1) + \sum_{l \in T^-_j} \tilde{x}^k_{lj}(t-1) \right), \quad j \in T. \tag{33}$$

The modified linear costs are then computed as follows:

$$\bar{c}^k_{ij}(t) = \begin{cases} c^k_{ij} + v^k \alpha_j \frac{f_j}{\tilde{X}_j(t-1)}, & \text{if } \tilde{X}_j(t-1) > 0, \\ \bar{c}^k_{ij}(t-1), & \text{otherwise,} \end{cases} \qquad (i,j) \in A_{OT}, k \in K, \tag{34}$$

$$\bar{c}^k_{ji}(t) = \begin{cases} c^k_{ji} + v^k (1-\alpha_j) \frac{f_j}{\tilde{X}_j(t-1)}, & \text{if } \tilde{X}_j(t-1) > 0, \\ \bar{c}^k_{ji}(t-1), & \text{otherwise,} \end{cases} \qquad (j,i) \in A_{TD}, k \in K, \tag{35}$$

$$\bar{c}^k_{lj}(t) = \begin{cases} c^k_{lj} + v^k (\alpha_j \frac{f_j}{\tilde{X}_j(t-1)} + (1-\alpha_l) \frac{f_l}{\tilde{X}_l(t-1)}), & \text{if } \tilde{X}_j(t-1),\ \tilde{X}_l(t-1) > 0, \\ \bar{c}^k_{lj}(t-1), & \text{otherwise,} \end{cases} \qquad (l,j) \in A_{TT}, k \in K, \tag{36}$$

The stopping criterion of the slope scaling procedure is when the objective function remains unchanged from one iteration to the next. At the end of the slope scaling procedure, we apply an intensification step that proceeds as follows. First, starting from the best feasible solution ($\tilde{x}$, $\tilde{y}$) found during this call to the slope scaling procedure, it removes each closed depot and its incident arcs. Then, it solves the resulting *MMCF*, but with the linear costs equal to the original transportation costs. This simple intensification step often yields an improved solution, since $\tilde{x}$ is optimal with respect to the modified linear costs, but generally not with respect to the original transportation costs.

5. Computational experiments

In this section, we report computational results on a large set of randomly generated instances. Our objective is to evaluate the performance of the Lagrangian relaxation approach, in particular the Lagrangian heuristic method. We compare the lower bounds computed by the bundle method with those obtained from solving the LP relaxation and the root node of the MIP model using a state-of-the-art MIP solver. We compare as well the upper bounds from the Lagrangian heuristic method with those obtained when solving the root node of the MIP model with the same state-of-the-art MIP solver. Note that we do not attempt to compute the optimal values since solving the MIP model exactly would require excessive computing times.

The Lagrangian heuristic method is implemented in C++. For the bundle method, we use the code in Frangioni (2005). For the *MMCF* problem, we use CPLEX (version 12.8.0). The code is compiled and run on a 3.07 GHz Intel Xeon X5675 computer.

We report computational results on problem instances obtained with the generator described in Gendron et al. (2003b). We generate three types of instances, with 10 instances of each type: 2×2 grid, 3×2 grid and 4×2 grid, as in Gendron et al. (2003b). All of these instances have the same number of customers ($|O| = |D| = 500$), the same number of depots ($|T| = 200$) and the

Instance	$\|A_{TT}\|$	$\|A_{TD}\|$	$\|A_{OT}\|$
$2\times2\_0$	796000	2000000	2000000
$2\times2\_1$	796000	2000000	2000000
$2\times2\_2$	796000	2000000	2000000
$2\times2\_3$	796000	2000000	2000000
$2\times2\_4$	796000	2000000	2000000
$2\times2\_5$	796000	2000000	2000000
$2\times2\_6$	796000	2000000	2000000
$2\times2\_7$	796000	2000000	2000000
$2\times2\_8$	796000	2000000	2000000
$2\times2\_9$	796000	2000000	2000000
$3\times2\_0$	796000	1174160	1174160
$3\times2\_1$	796000	1186920	1186920
$3\times2\_2$	796000	1503040	1503040
$3\times2\_3$	796000	1323000	1323000
$3\times2\_4$	796000	1425300	1425300
$3\times2\_5$	796000	1392640	1392640
$3\times2\_6$	796000	1208360	1208360
$3\times2\_7$	796000	1459980	1459980
$3\times2\_8$	796000	1375320	1375320
$3\times2\_9$	796000	1274000	1274000
$4\times2\_0$	796000	1165160	1165160
$4\times2\_1$	796000	1164880	1164880
$4\times2\_2$	796000	1150120	1150120
$4\times2\_3$	796000	1109680	1109680
$4\times2\_4$	796000	1127040	1127040
$4\times2\_5$	796000	1094700	1094700
$4\times2\_6$	796000	1034080	1034080
$4\times2\_7$	796000	1157280	1157280
$4\times2\_8$	796000	1044440	1044440
$4\times2\_9$	796000	1115640	1115640

Table 1 Size of the instances

same number of commodities ($|K| = 20$). The difference between them is the number of arcs, as illustrated in Table 1.

The following methods are used to compute lower and upper bounds on the optimal value of the *MCLB*:

• Root LB: The lower bound obtained by solving the root node of the branch-and-bound method of CPLEX, with the barrier method (otherwise using default parameters).

• LP: The LP relaxation bound computed by the LP solver of CPLEX, with the barrier method (otherwise using default parameters).

• Bundle: The lower bound obtained by the Lagrangian relaxation method to solve the Lagrangian dual presented in Section 3.

• Root UB: The upper bound obtained by solving the root node of the branch-and-bound method of CPLEX, with the barrier method (otherwise using default parameters).

- Slope Scaling: The upper bound computed by the Lagrangian heuristic method with slope scaling described in Section 4.

Two performance measures are provided: the GAP and the CPU times in seconds. For the GAP, we use the following measure:

$$GAP = \frac{|\text{best solution value} - \text{lower or upper bound}|}{\text{best solution value}} \times 100.$$

The best solution value is obtained by solving the root node of the branch-and-bound method of CPLEX (Root UB).

Tables 2 and 3 summarize the computational results obtained for our set of 30 instances. In Table 2, we show the solution values and GAPs for the lower bound yielded by CPLEX at the root (Root LB). We do not include the times for Root LB because the total times associated with the root node computations are reported in Table 3 under column Root UB. Table 2 also shows the solution values, the GAPs and the times associated with the LP relaxation and with the bundle method. The averages of each type of instances and for all instances are reported.

When comparing the LP relaxation and bundle methods, we conclude that the computing time of the bundle is small compared to that of the LP relaxation. In particular, the bundle method is about 4 times faster on 2×2 instances, almost 3 times faster on 3×2 instances and about 1.5 times faster on 4×2 instances. The GAP for the bundle method is between 0.1% and 1.89% with average values of 0.25%, 0.62% and 0.85%, respectively, on 2×2, 3×2 and 4×2 instances. These GAPs are close to those obtained by the LP relaxation, which are 0.09%, 0.10% and 0.12% on 2×2, 3×2 and 4×2 instances respectively. When we compare the Root LB and the LP methods, we observe that the GAP of LP is very close to the one of Root LB, which provides lower bounds that are about 0.01% away from the LP relaxation bounds. This indicates that CPLEX, even with its sophisticated MIP features (including preprocessing and cut generation), is not able to improve the LP relaxation bounds. Note that the computing time for the Root LB (see Table 3) is 3 to 4 times higher than that of LP.

Table 3 reports the values of the upper bounds yielded by CPLEX (Root UB) and the times in seconds, as well as the solution values, GAPs and times in seconds for the slope scaling method. The averages of each type of instances and for all instances are reported. When analyzing the results of the slope scaling method, all the instances have low GAPs: the worst values of GAP for the 2×2, 3×2 and 4×2 instances are, respectively, 1.80%, 2.77% and 1.47%, while the average values are 0.55%, 0.73% and 0.52%. We conclude that the Lagrangian heuristic is robust. Concerning the computing time, the slope scaling heuristic is several orders of magnitude faster: we observe an average of 1743.9 seconds compared to 145865.5 seconds for Root UB. The whole Lagrangian heuristic, including the bundle method, is still one order of magnitude faster with a global computing time of about 14000 seconds on average.

Instances	Root LB		LP			Bundle		
	Solution	GAP (%)	Solution	GAP (%)	Time (sec)	Solution	GAP (%)	Time (sec)
2×2_0	350925444	0.03	350905854	0.04	54766	350551187	0.14	13223
2×2_1	336006068	0.05	335957076	0.06	59118	335314927	0.25	13535
2×2_2	268039678	0.05	268008997	0.06	40914	267683442	0.18	13580
2×2_3	294333110	0.06	294331910	0.06	51956	293710032	0.27	13757
2×2_4	306430176	0.13	306406378	0.14	43133	305990252	0.27	13202
2×2_5	308499446	0.08	308457806	0.09	46756	307917135	0.27	10652
2×2_6	312042692	0.06	312041052	0.06	51106	311391878	0.27	12418
2×2_7	312541642	0.09	312540341	0.09	47331	312145356	0.22	14183
2×2_8	329794483	0.12	329793457	0.12	35300	329385379	0.24	12511
2×2_9	317096075	0.14	317084503	0.14	56345	316465133	0.34	11428
Average	313570881.4	0.08	313552737.4	0.09	48672.5	313055472.1	0.25	12848.9
3×2_0	462902868	0.04	462899188	0.04	29939	459461282	0.78	14390
3×2_1	462196632	0.13	462175261	0.13	35373	461106182	0.36	10420
3×2_2	270282933	0.08	270280507	0.08	24408	269314224	0.44	13220
3×2_3	239360611	0.13	239358231	0.13	25787	237616889	0.86	11333
3×2_4	324810526	0.09	324788093	0.10	40053	323497285	0.49	12398
3×2_5	300810035	0.12	300798456	0.12	40509	299504138	0.55	13708
3×2_6	643236523	0.04	643234940	0.04	22487	641952188	0.24	10815
3×2_7	329932217	0.19	329928409	0.19	41534	327971299	0.79	11518
3×2_8	556601312	0.07	556598680	0.07	56645	556311718	0.12	10870
3×2_9	355777075	0.11	355773655	0.11	31884	350403242	1.62	11833
Average	394591073.2	0.10	394583542	0.10	34861.9	392713844.7	0.62	12050.5
4×2_0	509137774	0.13	509121673	0.13	23686	504750042	0.99	12466
4×2_1	749569623	0.04	749550238	0.04	25178	744895577	0.66	12972
4×2_2	512615632	0.16	512608332	0.16	12425	508396243	0.98	11332
4×2_3	460256664	0.18	460248382	0.19	14605	457815343	0.71	12830
4×2_4	510258242	0.10	510248078	0.10	17569	507829897	0.58	11237
4×2_5	695587597	0.10	695585496	0.10	15980	693886195	0.34	13926
4×2_6	436717386	0.15	436713379	0.15	23859	429142676	1.89	11515
4×2_7	584436722	0.12	584433431	0.12	20609	581645954	0.60	10028
4×2_8	548055019	0.09	548051126	0.09	20692	543123748	0.99	11902
4×2_9	611104201	0.10	611103094	0.10	14592	606862561	0.79	13063
Average	561773886	0.12	561766322.9	0.12	18919.5	557834823.6	0.85	12127.1
Average All	433810768.6	0.10	433801889.9	0.11	32406.4	431536532.9	0.61	12231.7

Table 2 Lower bounds

6. Conclusions

We have studied the multicommodity capacitated location problem with balancing requirements ($MCLB$). For solving this problem, we have presented a Lagrangian relaxation approach based on the bundle method and a Lagrangian heuristic based on a slope scaling procedure. We have shown that the bundle method is faster than solving the LP relaxation with a state-of-the art solver for all the instances used in our experiments (we observe an average of 12231.7 seconds compared to 32406.4 seconds for the LP relaxation). The average GAP obtained by the bundle method for all

Instances	Root UB		Slope Scaling		
	Solution	Time (sec)	Solution	GAP (%)	Time (sec)
$2 \times 2\_0$	351051596	115122	351561773	0.14	1149
$2 \times 2\_1$	336184267	154787	337443555	0.37	1378
$2 \times 2\_2$	268190651	442341	268781050	0.22	1487
$2 \times 2\_3$	294529997	196208	297055088	0.85	1223
$2 \times 2\_4$	306842224	323060	307600967	0.24	975
$2 \times 2\_5$	308762065	350471	309836977	0.34	1217
$2 \times 2\_6$	312254851	197320	314650767	0.76	1347
$2 \times 2\_7$	312836669	206937	313196632	0.11	1346
$2 \times 2\_8$	330207910	254072	336183200	1.80	1463
$2 \times 2\_9$	317545544	226828	319623461	0.65	1179
Average	313840577.4	246714.6	315593347	0.55	1276.4
$3 \times 2\_0$	463114946	93590	464038311	0.19	5206
$3 \times 2\_1$	462815632	151792	463307925	0.10	1771
$3 \times 2\_2$	270504978	163534	270993018	0.18	2935
$3 \times 2\_3$	239678826	108971	240571040	0.37	1143
$3 \times 2\_4$	325118194	141291	327642181	0.77	605
$3 \times 2\_5$	301189597	99170	302481156	0.42	2235
$3 \times 2\_6$	643526866	54046	646194415	0.41	3165
$3 \times 2\_7$	330587311	238669	339758173	2.77	1211
$3 \times 2\_8$	557007675	117148	559757377	0.49	1912
$3 \times 2\_9$	356195243	85690	361763737	1.56	2773
Average	394973926.8	125390.1	397650733.3	0.73	2295.6
$4 \times 2\_0$	509816963	124884	511561788	0.34	2422
$4 \times 2\_1$	749873946	87776	751088152	0.16	3346
$4 \times 2\_2$	513467648	88015	516142853	0.52	1071
$4 \times 2\_3$	461125045	95658	467921405	1.47	1692
$4 \times 2\_4$	510794505	85480	511909523	0.21	1192
$4 \times 2\_5$	696285534	45068	700470147	0.60	1730
$4 \times 2\_6$	437412659	83810	443510217	1.39	734
$4 \times 2\_7$	585193185	131324	586193608	0.17	732
$4 \times 2\_8$	548575868	88680	549871308	0.23	593
$4 \times 2\_9$	611737024	73909	612695052	0.15	1979
Average	562428237.7	90460.4	565136405.3	0.52	1549.1
Average All	434274788.2	145865.5	436813298.8	0.63	1743.9

Table 3 Upper bounds

the instances is 0.61% compared to 0.11% for the LP relaxation. Considering the upper bounds, we have shown that the slope scaling procedure is robust and orders of magnitude faster than solving the root node by CPLEX, for an average GAP of 0.63% on all the instances used in our experiments. For further research, we plan to explore alternative Lagrangian relaxations and other approaches such as Benders decomposition. We also propose to develop exact methods for solving the *MCLB*.

References

Bonnans, J. Frédéric, Jean Charles Gilbert, Claude Lemaréchal, Claudia A. Sagastizábal. 2006. *Numerical Optimization: Theoretical and Practical Aspects (Universitext)*. Springer-Verlag New York, Inc., Secaucus, NJ, USA.

Campbell, James F. 1994. Integer programming formulations of discrete hub location problems. *European Journal of Operational Research* **72**(2) 387–405.

Contreras, Ivan, Jean-François Cordeau, Gilbert Laporte. 2012. Exact solution of large-scale hub location problems with multiple capacity levels. *Transportation Science* **46**(4) 439–459.

Crainic, Teodor G., Michel Gendreau, Patrick Soriano, Michel Toulouse. 1993a. A tabu search procedure for multicommodity location/allocation with balancing requirements. *Annals of Operations Research* **41**(4) 359–383.

Crainic, Teodor Gabriel, Pierre Dejax, Louis Delorme. 1989. Models for multimode multicommodity location problems with interdepot balancing requirements. *Annals of Operations Research* **18**(1) 277–302.

Crainic, Teodor Gabriel, Louis Delorme. 1993. Dual-ascent procedures for multicommodity location-allocation problems with balancing requirements. *Transportation Science* **27**(2) 90–101.

Crainic, Teodor Gabriel, Louis Delorme, Pierre Dejax. 1993b. A branch-and-bound method for multicommodity location with balancing requirements. *European Journal of Operational Research* **65**(3) 368–382.

Frangioni, Antonio. 2005. About lagrangian methods in integer optimization. *Annals of Operations Research* **139**(1) 163–193.

Gendron, Bernard, Teodor Gabriel Crainic. 1995. A branch-and-bound algorithm for depot location and container fleet management. *Location Science* **3**(1) 39–53.

Gendron, Bernard, Teodor Gabriel Crainic. 1997. A parallel branch-and-bound algorithm for multicommodity location with balancing requirements. *Computers and Operations Research* **24**(9) 829–847.

Gendron, Bernard, Luis Gouveia. 2017. Reformulations by discretization for piecewise linear integer multicommodity network flow problems. *Transportation Science* **51**(2) 629–649.

Gendron, Bernard, Jean-Yves Potvin, Patrick Soriano. 1999. Tabu search with exact neighbor evaluation for multicommodity location with balancing requirements. *INFOR: Information Systems and Operational Research* **37**(3) 255–270.

Gendron, Bernard, Jean-Yves Potvin, Patrick Soriano. 2003a. A parallel hybrid heuristic for the multicommodity capacitated location problem with balancing requirements. *Parallel Computing* **29**(5) 591–606.

Gendron, Bernard, Jean-Yves Potvin, Patrick Soriano. 2003b. A tabu search with slope scaling for the multicommodity capacitated location problem with balancing requirements. *Annals of Operations Research* **122**(1) 193–217.

Geoffrion, A. M. 1974. *Lagrangean relaxation for integer programming*. Springer Berlin Heidelberg, Berlin, Heidelberg, 82–114.

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

Kim, Dukwon, Panos M. Pardalos. 2000. Dynamic slope scaling and trust interval techniques for solving concave piecewise linear network flow problems. *Networks* **35**(3) 216–222.

Krarup, Jakob, Peter Mark Pruzan. 1983. The simple plant location problem: Survey and synthesis. *European Journal of Operational Research* **12**(1) 36–81.

Marin, Alfredo. 2005. Formulating and solving splittable capacitated multiple allocation hub location problems. *Computers Operations Research* **32**(12) 3093–3109.