



Solving a dynamic facility location problem with partial closing and reopening



Sanjay Dominik Jena^{a,c,*}, Jean-François Cordeau^{b,c}, Bernard Gendron^{a,c}

^a Département d'informatique et de recherche opérationnelle, Université de Montréal, C.P. 6128, succ. Centre-ville, Montréal, Canada H3C 3J7

^b Canada Research Chair in Logistics and Transportation, HEC Montréal, 3000 chemin de la Côte-Sainte-Catherine, Montréal, Canada H3T 2A7

^c Centre interuniversitaire de recherche sur les réseaux d'entreprise, la logistique et le transport (CIRRELT), C.P. 6128, succ. Centre-ville, Montréal, Canada H3C 3J7

ARTICLE INFO

Available online 28 October 2015

Keywords:

Facility location
Dynamic capacity adjustment
Lagrangian relaxation
Mixed-integer programming
Industrial application

ABSTRACT

Motivated by an industrial application, we consider a recently introduced multi-period facility location problem with multiple commodities and multiple capacity levels. The problem allows for the relocation of facilities, as well as for the temporary closing of parts of the facilities, while other parts remain open. In addition, it uses particular capacity constraints that involve integer rounding of the allocated demands. In this paper, we propose a strong formulation for the problem, as well as a hybrid heuristic that first applies Lagrangian relaxation and then constructs a restricted mixed-integer programming model based on the previously obtained Lagrangian solutions. Computational results for large-scale instances emphasize the usefulness of the heuristic in practice. While general-purpose mixed-integer programming solvers do not find feasible solutions for about half of the instances, the heuristic consistently provides high-quality solutions in short computing times, as well as tight bounds on their optimality.

© 2015 Elsevier Ltd. All rights reserved.

1. Introduction

Classical facility location aims at striking a balance between facility construction costs and transportation costs to satisfy customer demands. Operations research practitioners therefore contributed with a considerable variety of extensions to classical models to represent real world applications in a more realistic manner, involving the location of hospitals [31], telecommunication hubs [8], schools [1], manufacturing and distributing systems [27], and many others. The dynamic adjustment of the capacities over a planning horizon has often been a central issue. Problem extensions have been proposed to allow for the expansion and the reduction of capacity along time [23,1], temporary facility closing [8,10] and the relocation of capacities from one location to another [25]. Other important extensions acknowledged uncertainty in the customer demands [28] or the production capacities themselves (for references, see, e.g., [30]). Given the difficulty to solve those problems for real world sized instances, many solution algorithms have been suggested. Exact methods have been proposed for classical variants [35,15], whereas heuristics have proved to be effective for more complex problem variants. Due to the complicated structure of the latter, only a few works have applied methods that provide a bound on the solution quality, such as Benders decomposition and Lagrangian relaxation [10,20]. More

complex problem variants have therefore been solved by methods such as sophisticated local search [22,26], which, by themselves, do not allow for an assessment of the solution quality.

In this paper, we consider a multi-period facility location problem with multiple commodities and multiple capacity levels that has recently been introduced and applied in the forestry sector by Jena et al. [17]. In the application considered by the authors, a logging company must locate camps to host its workers. The problem involves several different ways to adjust capacity, namely, the expansion of capacity, the temporary closing of parts of the facility and the relocation of facilities from one location to another. Many of these features have already been discussed in early literature. The first multi-period models include those by Ballou [2] and Wesolowsky [36]. Multiple commodities have been considered by authors such as Geoffrin [14] and Warszawski [34]. Modular capacity levels have often been treated by offering a choice of facility size [21,29,9,16], whereas capacity expansion has been discussed in detail by Luss [23] and has been found to be a crucial feature in many applications [1,7,25]. Wesolowsky and Truscott [37] have been among the first to consider simple relocation of facilities, followed by several others [27,6,25]. While the temporary closing of entire facilities has been modeled in several studies [33,8,7,10], the problem introduced by Jena et al. [17] was the first to consider the partial closing and reopening of facilities along time. The authors propose a flow based formulation that uses a network structure for each facility location to manage the amount of available capacity and the amount of temporarily closed

* Corresponding author.

capacity of the facility. An integer flow, representing the number of open and closed capacity levels, allows for the closing of open capacity and the reopening of closed capacity. Another feature of the problem is the use of the so-called round-up capacity (RUC) constraints, which imply integer rounding of the total demand for each commodity allocated to the same facility. While this characteristic may correspond to the practice in many industries, to the best of our knowledge, the authors were the first to explicitly model this type of capacity constraints. Modeling the problem's features in detail results in complex models that raise questions of tractability. The problems solved by Jena et al. [17] were therefore of rather small size.

Contributions: In this paper, the above problem, subsequently referred to as the *Dynamic Facility Location Problem with Relocation and Partial Facility Closing (DFLP_RPC)* with RUC constraints, is revisited. A new formulation and a heuristic solution method are proposed to solve instances that are approximately 20 times larger in terms of facility locations and customers. We summarize our contributions as follows. First, a new mixed-integer programming (MIP) formulation for the DFLP_RPC with RUC constraints is introduced, based on the modeling technique proposed by Jena et al. [18]. While the latter considers rather simple variants of dynamic facility location problems, the formulation presented here accounts for additional features, namely the partial closing and reopening of facilities, the relocation of facilities and the round-up capacity (RUC) constraints. The new formulation has several advantages when compared to the formulation proposed by Jena et al. [17]. It yields integrality gaps that are, on average, more than 29 times smaller. Furthermore, it enables a state-of-the-art MIP solver to find feasible solutions for significantly more instances and to achieve a higher solution quality. The new formulation also allows for a more detailed representation of the cost structure. Second, we propose a Lagrangian based heuristic, capable to address large scale instances of the DFLP_RPC with RUC constraints. The heuristic consists of two optimization stages. In the first stage, Lagrangian relaxation is applied to provide lower and upper bounds for the problem. Then, a restricted MIP model, based on the Lagrangian solutions, is solved to improve the final solution quality. The heuristic substantially extends those proposed by Jena et al. [19] and accounts for the additional problem features, i.e., the partial closing and reopening of facilities, the relocation of facilities, and the RUC constraints. The technical challenges induced by these new features impact the algorithm on all levels: the set of relaxed constraints, the dynamic programming algorithm to solve the Lagrangian subproblems, the generation of primal feasible solutions, and the feeding strategy for the restricted MIP. Computational results have shown that the combination of the new formulation and the Lagrangian heuristic is quite powerful. The proposed heuristics are capable of finding high quality solutions in short computing times, even for large-scale instances for which a state-of-the-art MIP solver does not find feasible solutions. Furthermore, due to the strength of the proposed formulation, the heuristics provide significant bounds on the quality of the obtained solutions.

Outline: The remainder of the paper is organized as follows. Section 2 defines the problem and its application in forestry. Then, Section 3 introduces the new formulation for the DFLP_RPC with RUC constraints. The two-stage Lagrangian heuristic is presented in Section 4. Computational experiments for the problem, as well as for simplified problem variants without relocation and without RUC constraints, are presented in Section 5: the linear programming (LP) relaxation and the integrality gaps of the problems are analyzed; furthermore, the quality of the solutions for the industrial problem provided by a general-purpose MIP solver and the proposed heuristics are compared. Finally, conclusions are drawn in Section 6.

2. Problem description

We consider the problem introduced by Jena et al. [17], which extends the Capacitated Facility Location Problem in several aspects: multiple time periods, multiple (modular) capacity levels and multiple commodity types. Given a set of customers with independent demands for each commodity and time period, the objective is to find the optimal locations and opening schedules for facilities that provide sufficient capacity to satisfy the customer demands at minimal costs. New facilities may be constructed and existing facilities may expand their capacity at any time period. Since a facility may not always require its entire capacity, parts of the facility may be temporarily closed, while other parts remain open.

Given that the temporary closing and reopening of capacity is usually much cheaper than the complete shut-down and construction of a facility, this feature may result in a very dynamic opening schedule of the facilities. Throughout this paper, we will denote the capacity that is available for use as the *open* capacity. In contrast, we denote the capacity that is temporarily not available as the *closed* capacity. Closed capacity can be reopened at a later moment. Finally, the *existing* capacity is defined as the sum of the open and the closed capacity. Facilities may be relocated from one location to another, assuming: (1) a facility can only be relocated as a whole, not partially; (2) before it is relocated, the entire capacity of a facility has to be closed; (3) facilities cannot be merged at the same location.

In contrast to classical facility location models, the problem considered here involves particular capacity constraints, the above mentioned round-up capacity (RUC) constraints. These constraints require that, even though facilities may be able to provide the exact level of capacity required, they need to reserve production capacity in multiples of a certain size. This involves rounding the demands for each commodity according to the lot sizes to compute the total capacity necessary at the facility. The following example illustrates these constraints. In a given time period, a set of customers have been allocated to obtain a total of 287 units of commodity A and 113 units of commodity B from a certain facility. Let us assume that this facility needs to reserve blocks of size 100 for the production of commodity A and blocks of size 150 for the production of a commodity B. Even though the facility may produce the exact amount required by the customers, it needs to ensure a total capacity of 300 units, i.e., three blocks, for commodity A and 150 units, i.e., one block, for commodity B.

Application in industry: The DFLP_RPC with RUC constraints was motivated by an industrial application in the forestry sector introduced by Jena et al. [17], where a logging company needs to locate camps to host its workers. Facilities represent logging camps, while customers represent logging regions that specify a total demand for two different commodities: the workforce for wood logging and the workforce for the construction and maintenance of access roads. Demands are specified over a time horizon of five years, each year divided into a summer and a winter season. Logging camps are composed by trailers and therefore have a very flexible structure. The capacity level of a facility thus represents the number of trailers at the camp. The hosting capacity of a logging camp can easily be expanded by adding new trailers. Some trailers may be closed, while others remain open. Trailers are only available for use when they are open. The total number of trailers of a camp, i.e., the sum of open and closed trailers, is also referred to as the number of existing trailers. Demands are specified as the average number of crews working throughout the entire season. It is likely that a crew will only work a part of the season in a given region, which leads to a fractional demand. Given that crews always work together, the logging camp must ensure sufficient hosting capacity for the entire crew. The RUC constraints therefore ensure that capacity is modeled in a

realistic manner. As an example, let us assume that one crew will work the entire season, while another one will work only 40% of the season at a given site. The total demand for the two crews is thus 1.4 and a logging camp would need to ensure a total capacity for at least $\lceil 1.4 \rceil = 2$ crews in order to host the workers of both crews in the same time period.

Even though the two new features introduced in the problem considered here, namely the partial temporary closing and the RUC constraints, have been motivated by an application in forestry, they also apply to different contexts. For example, a company may have the possibility to temporarily close parts of warehouses, production facilities or stores, according to seasonal demand. Here, the cost of acquisition and sale (i.e., shut-down) of such capacity is different from the costs of temporary closing and reopening. Furthermore, production facilities often produce items in lots of a certain size. However, previous models may have simplified such details either to avoid the additional modeling complexity or due to the strategic nature of the corresponding problem.

3. Mathematical formulations

In this section, we propose a new formulation based on a modeling technique that has shown to yield very strong LP relaxation bounds. We first review the input data used to model the problem. We denote by J the set of potential facility locations and by $L = \{0, 1, 2, \dots, q\}$ the set of possible capacity levels for each facility. We also denote by I the set of customer demand points and by $T = \{1, 2, \dots, \bar{t}\}$ the set of time periods in the planning horizon. We assume throughout that the beginning of period $t + 1$ corresponds to the end of period t . The set of different commodities is denoted by P . The demand of customer i for commodity $p \in P$ in period t is denoted by d_p^{it} . The cost to serve one unit of commodity p to customer i from facility j open at capacity level ℓ during period t is denoted by g_{jp}^{it} . We denote by s_p the block size (in commodity units) that has to be reserved at a facility to provide commodity type p . A facility may temporarily close parts of its capacity. The capacity of a facility with ℓ open capacity levels at location j is given by u_ℓ^j (with $u_0^j = 0$). Furthermore, we let $J^0 \subseteq J$ be the set of locations that already possess existing facilities at the beginning of the planning horizon and \hat{n}^j be the capacity level (i.e., the total capacity) of the facility initially existing at location $j \in J^0$. The costs to construct a facility of size $n \in L$ (i.e., a new facility with n capacity levels) or to expand the capacity of an existing facility by n capacity levels at location $j \in J$ is denoted by c_{jn}^C . Newly constructed capacity is assumed to be open, i.e., ready for use, after construction. New capacity therefore affects the levels of the existing and open capacity. The costs to reopen and close ℓ capacity levels of the same facility are given by c_ℓ^{TO} and c_ℓ^{TC} , respectively. The maintenance costs for ℓ open capacity levels at a facility during period t is given by c_ℓ^M . Finally, c_n^R represents the costs for relocating a facility with n closed capacity levels.

We now present a new formulation for the DFLP_RPC with RUC constraints, inspired by the work of Jena et al. [18] on a simpler dynamic facility location problem with capacity adjustment, where it was shown that using binary variables with indices that represent explicit capacity changes from level ℓ_1 to ℓ_2 results in significantly stronger formulations. In the DFLP_RPC, one needs to simultaneously manage capacity on two levels: the existing capacity and the open capacity. We therefore extend this modeling technique and use binary variables $y_{\ell_1 \ell_2 n_1 n_2}^{jt}$ that take value 1 if a facility at location j changes its existing capacity from level n_1 to n_2 and its open capacity from level ℓ_1 to ℓ_2 at the beginning of time period t . Clearly, variables are defined only for $\ell_1 \leq n_1$ and $\ell_2 \leq n_2$.

Furthermore, we assume that $n_1 \leq n_2$, since facilities may only expand their capacity, but not remove existing capacity. Integer

variables $z_{\ell p}^{jt} \in Z_0^+$ represent the total number of blocks of commodity type p reserved at a facility with ℓ open capacity levels, located at $j \in J$ at time period t . Binary variables $\hat{w}_{\ell n}^{jt}$ indicate whether a facility of size n (i.e., a facility with n existing capacity levels), open at capacity level ℓ , is closed and relocated from location j to another location at the beginning of period t . Binary variables \check{w}_n^{jt} indicate whether a facility of size n is relocated to location j at the beginning of period t . The continuous variables $x_{\ell p}^{jt} \in [0, 1]$ denote the fraction of the demand d_p^{it} satisfied by a facility at location j open at capacity level ℓ .

The objective function coefficients associated to the capacity change decisions $y_{\ell_1 \ell_2 n_1 n_2}^{jt}$ are given by $f_{\ell_1 \ell_2 n_1 n_2}^{jt}$ and describe the aggregated costs to change the open capacity of a facility at location j from level ℓ_1 to ℓ_2 and the existing capacity from level n_1 to n_2 at the beginning of period t , as well as the costs to operate the facility at levels ℓ_2 and n_2 throughout time period t . This cost structure is more general than the one of the flow formulation used by Jena et al. [17] and could therefore represent the capacity transitions in a more realistic manner. However, to obtain a formulation equivalent to that of Jena et al. [17], we set the cost matrix $f_{\ell_1 \ell_2 n_1 n_2}^{jt}$ as follows.

The number of capacity levels constructed (nFC), the number of capacity levels reopened (nRE) and the number of capacity levels closed (nCL) that are represented by a decision variable $y_{\ell_1 \ell_2 n_1 n_2}^{jt}$ can be computed by

$$\begin{aligned} nFC &= n_2 - n_1 \\ nRE &= \max\{0, (\ell_2 - \ell_1) - nFC\} \\ nCL &= \max\{0, (\ell_1 - \ell_2) + nFC\}. \end{aligned}$$

The cost coefficients are then defined as

$$f_{\ell_1 \ell_2 n_1 n_2}^{jt} = c_{\ell_2}^M + c_{j(nFC)}^C + c_{(nRE)}^{TO} + c_{(nCL)}^{TC}.$$

We define the GMC based formulation for the DFLP_RPC with RUC constraints (RPCr-GMC) as follows:

$$\begin{aligned} \min \quad & \sum_{j \in J} \sum_{\ell_1 \in L} \sum_{\ell_2 \in L} \sum_{n_1 \in L} \sum_{n_2 \in L} \sum_{t \in T} f_{\ell_1 \ell_2 n_1 n_2}^{jt} y_{\ell_1 \ell_2 n_1 n_2}^{jt} \\ & + \sum_{i \in I} \sum_{j \in J} \sum_{\ell \in L} \sum_{p \in P} \sum_{t \in T} g_{jp}^{ijt} x_{\ell p}^{ijt} + \sum_{j \in J} \sum_{\ell \in L} \sum_{n \in L} \sum_{t \in T} \left(c_\ell^{TC} + \frac{c_n^R}{2} \right) \hat{w}_{\ell n}^{jt} \\ & + \sum_{j \in J} \sum_{n \in L} \sum_{t \in T} \frac{c_n^R}{2} \check{w}_n^{jt} \end{aligned} \quad (1)$$

$$\text{s.t.} \quad \sum_{j \in J} \sum_{\ell \in L} x_{\ell p}^{ijt} = 1 \quad \forall i \in I, \quad \forall p \in P, \quad \forall t \in T \quad (2)$$

$$\sum_{i \in I} d_p^{it} x_{\ell p}^{ijt} \leq s_p z_{\ell p}^{jt} \quad \forall j \in J, \quad \forall \ell \in L, \quad \forall p \in P, \quad \forall t \in T \quad (3)$$

$$\sum_{p \in P} s_p z_{\ell p}^{jt} \leq \sum_{\ell_1 \in L} \sum_{\ell_2 \in L} u_{\ell_2}^j y_{\ell_1 \ell_2 n_1 n_2}^{jt} \quad \forall j \in J, \quad \forall \ell \in L, \quad \forall t \in T \quad (4)$$

$$\begin{aligned} \sum_{\ell_1 \in L} \sum_{\ell_2 \in L} y_{\ell_1 \ell_2 n_1 n_2}^{jt} &= \sum_{\ell_2 \in L} \sum_{n_2 \in L} y_{0 \ell_2 n_2}^{jt} + \hat{w}_{0n}^{jt} \\ \forall j \in J, \quad \forall n \in L \setminus \{0\}, \quad \forall \ell = 1, \dots, n, \quad \forall t \in T \setminus \{1\} \end{aligned} \quad (5)$$

$$\begin{aligned} \sum_{\ell_1 \in L} \sum_{\ell_2 \in L} y_{\ell_1 0 n_1 n_2}^{jt} + \check{w}_n^{jt} &= \sum_{\ell_2 \in L} \sum_{n_2 \in L} y_{0 \ell_2 n_2}^{jt} + \hat{w}_{0n}^{jt} \\ \forall j \in J, \quad \forall n \in L \setminus \{0\}, \quad \forall t \in T \setminus \{1\} \end{aligned} \quad (6)$$

$$\sum_{\ell_2 \in L} \sum_{n_2 \in L} y_{\hat{n}^j \ell_2 n_1 n_2}^{jt} = 1 \quad \forall j \in J^0 \quad (7)$$

$$\sum_{\ell_1 \in L} \sum_{\ell_2 \in L} \sum_{n_1 \in L} \sum_{n_2 \in L} y_{\ell_1 \ell_2 n_1 n_2}^{jt} \leq 1 \quad \forall j \in J, \quad \forall t \in T \quad (8)$$

$$\sum_{j \in J} \sum_{\ell \in L} \hat{w}_{\ell n}^{jt} = \sum_{j \in J} \hat{w}_n^{jt} \quad \forall n \in L, \quad \forall t \in T \quad (9)$$

$$x_{\ell p}^{jit} \in \mathbb{R}_0^+ \quad \forall i \in I, \quad \forall j \in J, \quad \forall \ell \in L, \quad \forall p \in P, \quad \forall t \in T \quad (10)$$

$$y_{\ell_1 \ell_2 n_1 n_2}^{jt} \in \{0, 1\} \quad \forall i \in I, \quad \forall j \in J, \quad \forall n_1 \in L, \quad \forall n_2 = n_1, \dots, q, \\ \forall \ell_1 = 0, \dots, n_1, \quad \forall \ell_2 = 0, \dots, n_2, \quad \forall t \in T \quad (11)$$

$$\hat{w}_{\ell n}^{jt} \in \{0, 1\} \quad \forall j \in J, \quad \forall n \in L, \quad \forall \ell = 0, \dots, n, \quad \forall t \in T \quad (12)$$

$$\check{w}_n^{jt} \in \{0, 1\} \quad \forall j \in J \quad \forall n \in L, \quad \forall t \in T \quad (13)$$

$$z_{\ell p}^{jt} \in \mathbb{Z}_0^+ \quad \forall j \in J, \quad \forall \ell \in L, \quad \forall p \in P, \quad \forall t \in T. \quad (14)$$

The objective function (1) minimizes the total costs for changing the capacity levels and allocating the demand. Note that the relocation costs c_n^R are equally split on variables \hat{w} and \check{w} in order to better use both variables within the Lagrangian relaxation. Constraints (2) are the demand constraints for the customers. Constraints (3) and (4) are the round-up capacity constraints at the facilities that first round up the total demand (specified in number of blocks) and then use the rounded number of blocks to determine the total capacity necessary at the facility. Constraints (5) and (6) are the flow conservation constraints that link the capacity change variables in consecutive time periods. These constraints also allow the use of the relocation variables $\hat{w}_{\ell n}^{jt}$ and \check{w}_n^{jt} to remove flow from one location and add it to another location, respectively. Constraints (7) are the flow initialization constraints for the locations that already possess facilities with \hat{n}^j existing and open capacity levels, specifying that exactly one capacity level is chosen at the beginning of the planning horizon. Constraints (8) guarantee that at most one capacity change variable is selected at each time period and location. Finally, constraints (9) are the relocation linking constraints that match the outgoing and incoming relocations of facilities of the same size. The flow conservation constraints (5)–(7) represent a network flow structure, which is illustrated in Fig. 1 for a small example with four time periods and two capacity levels. Each node represents the number of open and existing capacity (open capacity level/existing capacity level). The binary capacity change variables are represented by arcs, which allow for the construction of new capacity, as well as the closing and reopening of existing capacity. Note that, after the construction of new capacity levels, open capacity levels may be temporarily closed and temporarily closed capacity may be reopened. The binary capacity change variables may therefore represent complex combinations of construction, temporary closing and reopening.

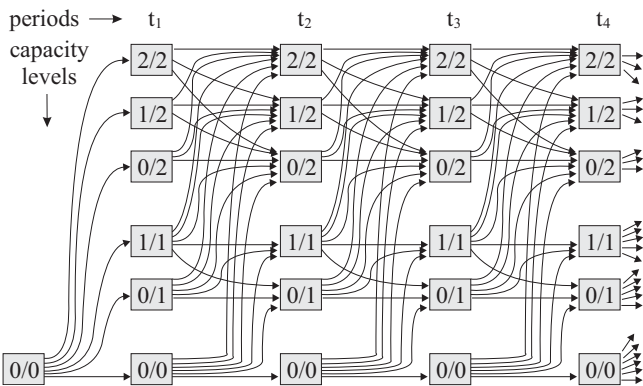


Fig. 1. Network flow structure to manage partial facility closing and reopening. Each node indicates the level of open and existing capacity.

Valid inequalities: We propose the following valid inequalities for our model:

$$x_{\ell p}^{jit} \leq \sum_{\ell_1 \in L_{n_1}} \sum_{\ell_2 \in L_{n_2}} y_{\ell_1 \ell_2 n_1 n_2}^{jt} \quad \forall i \in I, \quad \forall j \in J, \quad \forall \ell \in L, \quad \forall p \in P, \quad \forall t \in T \quad (15)$$

$$\sum_{j \in J} \sum_{\ell_1 \in L_{\ell_2}} \sum_{n_1 \in L_{n_2}} \sum_{\ell_2 \in L} u_{\ell_2}^j y_{\ell_1 \ell_2 n_1 n_2}^{jt} \geq \lceil \sum_{i \in I} \sum_{p \in P} d_p^{it} \rceil \quad \forall t \in T. \quad (16)$$

The *Strong Inequalities (SI)* (15), typically used in facility location and network design problems [32,13], are known to provide a tight upper bound for the demand assignment variables. The valid inequalities (16) are a strengthened variant of the *Aggregated Demand Constraints (ADC)*, where the right-hand side is replaced by $\sum_{i \in I} \lceil \sum_{p \in P} d_p^{it} \rceil$. Although the ADCs are redundant for the LP relaxation, the strengthened variant may tighten the formulation. Furthermore, adding them to the model enables MIP solvers to generate cover cuts that further strengthen the formulation.

3.1. Problem variants

In this section, we present variants of the problem, which are introduced to assess the impact of the particular features of the DFLP_RPC, i.e., the relocation of facilities and the RUC constraints. The latter are a particular characteristic of the industrial application considered here. The majority of facility location models in the literature uses classical capacity constraints such as the following:

$$\sum_{i \in I} \sum_{p \in P} d_p^{it} x_{\ell p}^{jit} \leq \sum_{\ell_1 \in L_{n_1}} \sum_{\ell_2 \in L_{n_2}} u_{\ell_2}^j y_{\ell_1 \ell_2 n_1 n_2}^{jt} \quad \forall j \in J, \quad \forall \ell \in L, \quad \forall t \in T. \quad (17)$$

Note that, when using the classical capacity constraints (17), the strengthened ADCs (16) are no longer valid. Instead, we use the classical ADCs, defined in a similar manner, but without the ceiling operator on the right-hand side of (16).

Based on the classical capacity constraints (17), we define two simplified variants of the problem: the DFLP_RPC allows for relocation, while the DFLP_PC does not. The GMC based formulation for the DFLP_RPC, referred to as the RPC-GMC formulation, is defined by objective function (1) and constraints (2), (5)–(13) and (17). The GMC based formulation for the DFLP_RPC is referred to as the PC-GMC formulation and is defined by objective function (1) and constraints (2), (5)–(8), (10)–(13) and (17), but without the relocation variables $\hat{w}_{\ell n}^{jt}$ and \check{w}_n^{jt} and the RUC integer variables $z_{\ell p}^{jt}$. We refer to the flow formulation proposed by Jena et al. [17] as the RPCr-2i formulation. Based on this formulation, we also define formulations for the DFLP_PC and the DFLP_RPC with classical capacity constraints. We denote the corresponding formulations as the PC-2i and RPC-2i, respectively.

4. Lagrangian heuristics

When applying Lagrangian relaxation to capacitated facility location problems, it is common to relax either the capacity constraints [33,3] or the demand constraints [29,4,38]. Jena et al. [19] applied the latter Lagrangian relaxation to the GMC formulation. By relaxing the demand constraints, the Lagrangian subproblem decomposes into independent subproblems, one for each candidate facility location. These independent subproblems can then be efficiently solved by dynamic programming.

When the relocation of facilities is allowed, as it is the case for our problem, the relocation constraints (9) are an additional link between the candidate facility locations. Therefore, the relaxation of the demand constraints is not sufficient to decompose the

Lagrangian subproblem by location. There are two possibilities to overcome this issue. One can relax both the demand constraints (2) and the relocation constraints (9) in order to obtain a subproblem that can be decomposed by location. Alternatively, one can relax only the demand constraints (2). The remaining Lagrangian subproblem then still includes the relocation linking constraints (9) and, therefore, cannot be decomposed by location. Instead, the subproblem may be transformed into a pure integer program and be solved by a generic MIP solver. In our computational experiments, the latter approach has not been competitive with the former one, given that the MIP solver may still take considerable time to solve the resulting integer program. We therefore report on the first approach, relaxing both the demand constraints (2) and the relocation linking constraints (9).

Let α be the vector of Lagrange multipliers associated to the relaxed demand constraints and β the one associated to the relaxed relocation linking constraints. Let $\tilde{c}_{\ell p}^{ijt} = g_{\ell p}^{ijt} d_p^{it} - \alpha_{ipt}$ denote the modified variable coefficients for the x variables. The Lagrangian subproblem, including the Strong Inequalities, can be stated as follows:

$$L(\alpha, \beta) = \min \sum_{j \in J} \sum_{\ell_1 \in L} \sum_{\ell_2 \in L} \sum_{n_1 \in N} \sum_{n_2 \in N} \sum_{t \in T} f_{\ell_1 \ell_2 n_1 n_2}^{jt} y_{\ell_1 \ell_2 n_1 n_2}^{jt} + \sum_{j \in J} \sum_{\ell \in L} \sum_{n \in N} \sum_{t \in T} \left(c_{\ell}^{TC} + \frac{c_n^R}{2} - \beta_{nt} \right) \hat{w}_{\ell n}^{jt} + \sum_{j \in J} \sum_{n \in N} \sum_{t \in T} \left(\frac{c_n^R}{2} + \beta_{nt} \right) \hat{w}_n^{jt} + \sum_{i \in I} \sum_{j \in J} \sum_{\ell \in L} \sum_{p \in P} \sum_{t \in T} \tilde{c}_{\ell p}^{ijt} x_{\ell p}^{ijt} + \sum_{i \in I} \sum_{p \in P} \sum_{t \in T} \alpha_{ipt}$$

s.t. (3)–(8), (10)–(15).

4.1. Solution of the Lagrangian subproblem

We separate the Lagrangian subproblem into $|J|$ independent subproblems, one for each potential facility location for a fixed set of Lagrange multipliers α and β . The Lagrangian subproblem is then defined as $L(\alpha, \beta) = \sum_{j \in J} L_j(\alpha, \beta) + \sum_{i \in I} \sum_{p \in P} \sum_{t \in T} \alpha_{ipt}$, where $L_j(\alpha, \beta)$ corresponds to the problem of finding the optimal opening schedule for the capacities of facility j with modified demand allocation costs $\tilde{c}_{\ell p}^{ijt}$. To solve this problem, we extend the dynamic programming algorithm presented by [19]. Let $O_j^{\alpha, \beta}(\ell, n, t)$ denote the value of the optimal opening schedule for facility j from time period 1 to t , including the costs to satisfy the customer demand during these time periods and assuming that a facility of size n (i.e., a facility with n existing capacity levels), open at capacity level ℓ is available at the end of time period t . To compute these values, we need to evaluate the following four combinations of incoming and outgoing relocations:

1. No incoming relocation, no outgoing relocation. The cheapest capacity level is chosen, including the costs to satisfy demand until period $t - 1$ and the costs for the capacity transition:

$$\hat{C}_1^j(\ell, n, t) = \min_{0 \leq n_1 \leq n, 0 \leq \ell_1 \leq n_1} \{ f_{\ell_1 \ell n_1 n}^{jt} + O_j^{\alpha, \beta}(\ell_1, n_1, t - 1) \}.$$

2. Incoming relocation, no outgoing relocation. A facility with n_1 existing capacity levels has been relocated to location j and possibly expanded by additional capacity, resulting in a final level of existing capacity n . Furthermore, some unused capacity may have been closed, resulting in a final level of open capacity ℓ :

$$\hat{C}_{RelocIN}^j(\ell, n, t) = \min_{1 \leq n_1 \leq n} \left\{ \frac{c_{n_1}^R}{2} + c_{(nRE)}^{TO} + c_{(nCL)}^{TC} + \beta_{n_1 t} + c_{j(n-n_1)}^C \right\},$$

where $nRE = \max\{0, \ell - n + n_1\}$ and $nCL = \max\{0, n - n_1 - \ell\}$.

$$\hat{C}_2^j(\ell, n, t) = \hat{C}_{RelocIN}^j(\ell, n, t) + c_{\ell}^M + O_j^{\alpha, \beta}(0, 0, t - 1).$$

3. No incoming relocation, outgoing relocation. A facility has been relocated to another location and a facility of size $\langle \ell, n \rangle$ (i.e., ℓ open and n existing capacity levels) has been constructed afterwards:

$$\hat{C}_{RelocOUT}^j = \min_{1 \leq n_1 \leq q, 0 \leq \ell_1 \leq n_1} \left\{ c_{\ell_1}^{TC} + \frac{c_{n_1}^R}{2} - \beta_{n_1 t} + O_j^{\alpha, \beta}(\ell_1, n_1, t - 1) \right\}$$

$$\hat{C}_3^j(\ell, n, t) = \hat{C}_{RelocOUT}^j + J_{0\ell n}^{jt}.$$

4. Incoming relocation, outgoing relocation. A facility has been relocated to another location, while a facility has been relocated to the current location, eventually followed by a capacity expansion, resulting in a final capacity level n :

$$\hat{C}_4^j(\ell, n, t) = \hat{C}_{RelocIN}^j(\ell, n, t) + \hat{C}_{RelocOUT}^j + c_{\ell}^M.$$

Based on these four cases, the optimal value for $O_j^{\alpha, \beta}(\ell, n, t)$ with $t \geq 1$ is computed as

$$O_j^{\alpha, \beta}(\ell, n, t) = \hat{L}_j^{\alpha}(\ell, t) + \min \left\{ \hat{C}_1^j(\ell, n, t), \hat{C}_2^j(\ell, n, t), \hat{C}_3^j(\ell, n, t), \hat{C}_4^j(\ell, n, t) \right\}, \quad (18)$$

where $\hat{L}_j^{\alpha}(\ell, t)$ is the cost of the optimal demand allocation at facility j with ℓ open capacity levels at period t , which is computed as shown below. To initialize the recursive function at $t=0$, we differentiate two cases. If $j \notin J_0$, we set $O_j^{\alpha, \beta}(\ell, n, 0)$ to 0 if $\ell = 0$ and $n = 0$, and to $+\infty$ for all other values of ℓ and n . If $j \in J_0$, we set $O_j^{\alpha, \beta}(\ell, n, 0)$ to 0 if $\ell = 0$ and $n = \hat{n}^j$, and to $+\infty$ for all other values of ℓ and n . Note that, for the DFLP_PC, the recursive functions above are significantly simpler, as they ignore the relocation decisions.

The subproblem for location j is solved by selecting the minimum among all possible facility sizes:

$$L_j(\alpha, \beta) = \min_{0 \leq \ell \leq q, \ell \leq n \leq q} \left\{ O_j^{\alpha, \beta}(\ell, n, \bar{t}) \right\}.$$

Note that, using the RUC constraints, the Lagrangian subproblem does not have the integrality property, since its linear relaxation can yield fractional values for the integer variables z . The lower bound provided by the Lagrangian subproblem may thus be better than the bound provided by the LP relaxation of the original problem.

Computation of the optimal demand allocation: When using common capacity constraints, as used in the Capacitated Facility Location Problem, the optimal demand allocation $\hat{L}_j^{\alpha}(\ell, t)$, taking into account the adjusted demand allocation costs $\tilde{c}_{\ell p}^{ijt}$ for a given α, j, ℓ and t , can be obtained by solving a continuous knapsack problem [29]. However, when using the RUC constraints, finding the optimal demand allocation is identical to solve the following MIP:

$$\begin{aligned} \hat{L}_j^{\alpha}(\ell, t) = \min & \sum_{i \in I} \sum_{p \in P} \tilde{c}_{\ell p}^{ijt} x_{\ell p}^{ijt} \\ \text{s.t.} & \sum_{i \in I} d_p^{it} x_{\ell p}^{ijt} \leq s_p z_{\ell p}^{it} \quad \forall p \in P \\ & \sum_{p \in P} s_p z_{\ell p}^{it} \leq u_{\ell}^j \\ & x_{\ell p}^{ijt} \leq 1 \quad \forall i \in I, \quad \forall p \in P \end{aligned}$$

$$\begin{aligned} x_{\ell p}^{jt} &\in \mathbb{R}_0^+ \quad \forall i \in I, \quad \forall p \in P \\ z_{\ell p}^{jt} &\in \mathbb{Z}_0^+ \quad \forall p \in P. \end{aligned}$$

This problem contains two embedded knapsack problems. The overall knapsack problem consists in selecting, for each p , an integer value for $z_{\ell p}^{jt}$ such that the total capacity is respected and the costs are minimized. The cost for each of these integer values depends on the choice of x and is computed by a continuous knapsack that selects x variables such that the total costs are minimized. The steps to solve this problem are therefore as follows:

1. First, all feasible integer values for $z_{\ell p}^{jt}$ are identified. For each p , the feasible integer values that $z_{\ell p}^{jt}$ may take are given by set $\Omega_p = \{0, 1, 2, \dots, \lfloor u_{\ell}^j / s_p \rfloor\}$.
2. The costs for each of the integer values of $z_{\ell p}^{jt}$ are computed by solving a continuous knapsack problem for each $p \in P$, by rearranging the demand nodes d_p^{jt} in increasing order of their ratio of adjusted transportation costs and demand quantity, i.e., $\tilde{c}_{\ell p}^{jt} / d_p^{jt}$, and by serving demands until either the entire capacity, given by the value of $s_p z_{\ell p}^{jt}$, is filled or a demand node with a positive adjusted transportation cost is met.
3. A Multiple-Choice Knapsack Problem [24] is solved by using dynamic programming, in which each integer value for a $z_{\ell p}^{jt}$ variable represents an object with a cost coefficient given by the solution of the previously solved continuous knapsack. The weight of object $z_{\ell p}^{jt}$ is given by $s_p z_{\ell p}^{jt}$ and the total knapsack capacity is given by u_{ℓ}^j . Exactly one object for each $p \in P$ is selected.

Note that solving the series of continuous knapsacks for a given p in step 2 can be performed efficiently, as the optimal solution for the continuous knapsack of a capacity z' is necessarily part of an optimal solution for any capacity z'' with $z'' > z'$.

Classical capacity constraints: When applying Lagrangian relaxation to the simplified problem variants with classical capacity constraints, the optimal demand allocation in the subproblem becomes significantly easier to compute. Instead of solving a multiple-choice integer knapsack problem when RUC constraints are used, classical capacity constraints involve only the solution of simple continuous knapsack problems. This reduces the computational complexity of solving the Lagrangian subproblem.

4.2. Solution of the Lagrangian dual

The Lagrangian dual problem is maximized to obtain the optimal Lagrange multipliers:

$$\max_{\alpha, \beta} L(\alpha, \beta).$$

The Lagrangian function $L(\alpha, \beta)$ is non-differentiable. However, a subgradient direction can easily be computed. The subgradient direction is composed of the two vectors γ_{ipt} and μ_{nt} , which represent the subgradients for the relaxed demand and relocation linking constraints, respectively. At the k -th iteration, they are computed as the derivative of the relaxed constraints in α and β , respectively, with variables x , \hat{w} and \tilde{w} fixed to the values found in the Lagrangian subproblem:

$$\begin{aligned} \gamma_{ipt}^k &= 1 - \sum_{j \in J} \sum_{\ell \in L} x_{\ell p}^{ijt} \quad \forall i \in I, \quad \forall p \in P, \quad \forall t \in T \\ \mu_{nt}^k &= \sum_{j \in J} \tilde{w}_n^{jt} - \sum_{j \in J} \sum_{\ell \in L} \tilde{w}_{\ell n}^{jt} \quad \forall n \in L \setminus \{0\}, \quad \forall t \in T. \end{aligned}$$

We chose to use a bundle method to solve the Lagrangian dual. Bundle methods are known to possess stronger convergence properties than the classical subgradient method (e.g., [5]). We use an implementation of the aggregated bundle method based on Frangioni [11] that uses a subset of the tuples $\langle L(\alpha^s, \beta^s), \gamma^s \rangle$ where $s \in B$ is an identifier for the elements within B , which is referred to as the bundle of subgradients γ^s , and α^s and β^s are the corresponding multipliers. From the primal view point, the following quadratic problem has to be solved at each iteration [12]:

$$\min_{\theta^s} \left\{ \frac{1}{2} \left\| \sum_{s \in B} \gamma^s \theta^s \right\|^2 + \frac{1}{R} E_B \theta; \quad \text{s.t. } \sum_{s \in B} \theta^s = 1, \quad \theta^s \geq 0 \right\},$$

where R is the so-called trust region parameter for the tentative ascent direction, and $E_s = L(\alpha, \beta) + \gamma((\hat{\alpha}, \hat{\beta}) - (\alpha, \beta)) - L(\hat{\alpha}, \hat{\beta})$ is the linearization error from the current point $(\hat{\alpha}, \hat{\beta})$, with (α, β) denoting the concatenation of vectors α and β . The solution values for θ^s , given for each bundle member, hold valuable information and can be used to construct feasible integer solutions (see Section 4.4). The tentative ascent direction is then computed by the convex combination of the subgradients, using the multipliers θ^s . Alternatively, the dual problem can be solved to compute the ascent direction, or directly the new point. Frangioni and Gallo [12] elaborate on this relationship in detail.

4.3. Generation of upper bounds

Feasible solutions are generated based on the current Lagrangian solution. The costs of these solutions serve as upper bounds for the optimal value and impact the convergence of the bundle methods. We extend the approach followed by Jena et al. [19] by also considering partial closing and reopening of facilities, relocation of facilities and the RUC constraints. The Lagrangian solution provides a facility opening schedule for the entire planning horizon, defined by a capacity level pair (ℓ_{jt}^i / n_{jt}^i) for each j and t , as well as the corresponding demand allocation. As the demand constraints (2) have been relaxed, the set of demands d_p^{jt} can be separated into three subsets Σ_1 , Σ_2 and Σ_3 , which denote the demands defined by triplets $\langle i, p, t \rangle$ that are exactly met, over-served and under-served, respectively.

To obtain an integer feasible solution, we perform the following steps:

1. *Identify feasible relocation pairs:* If the problem allows for the relocation of facilities, feasible pairs of relocation decisions are identified by matching the outgoing relocations $\hat{w}_{\ell n}^{jt}$ and incoming relocations \tilde{w}_n^{jt} from the Lagrangian solution. For each pair of facility size n (i.e., n existing capacity levels) and time period t , we choose the maximum number of facility matches j' and j'' (with $j' \neq j''$) among the $\hat{w}_{\ell n}^{j't}$ and $\tilde{w}_n^{j''t}$ decisions made in the Lagrangian solution. The procedure to find these pairs considers facilities without a specific order and excludes infeasible configurations, i.e., no outgoing relocation of a facility is smaller than a previously incoming relocation and subsequent incoming relocations at the same location are separated by outgoing relocation. For locations for which no relocation pairs have been selected, we generate the optimal opening schedule without relocations.
2. *Reduce demand allocation:* For each $\langle i, p, t \rangle \in \Sigma_2$, all facility/size pairs $(j, (\ell_{jt}^i))$ are sorted in decreasing order of their allocation costs $g_{\ell p}^{ijt}$. By following the sorted list, the allocated flow is removed until the total allocated demand for $\langle i, p, t \rangle$ equals 1, i.e., Σ_2 is empty and its previous elements are now in Σ_1 .
3. *Increase capacities:* If $\Sigma_3 \neq \emptyset$ and the total remaining capacity is smaller than the total remaining demand, capacity is increased sequentially at each time period according to the following

steps until the total demand can be met. Facilities are considered without a specific order. We consider two simple possibilities to increase capacity: if $\ell'_{jt} < n'_{jt}$, we incrementally increase ℓ'_{jt} until the missing capacity is covered or $\ell'_{jt} = n'_{jt}$; if no facility exists, we incrementally increase both ℓ'_{jt} and n'_{jt} until the missing capacity is covered or the maximum capacity level for this facility is reached. For any time period $t' > t$, the existing capacity level n'_{jt} is increased to the new level $n'_{jt'}$ if $n'_{jt'} < n'_{jt}$.

4. *Increase the demand allocation:* For each $(i, p, t) \in \Sigma_3$, all facility/size pairs $(j, (\ell'_{jt}))$ with remaining capacity are sorted in increasing order of their allocation costs g_{jp}^{jt} . Demand is allocated to these pairs until the total allocated demand for (i, p, t) equals 1. Note that, due to the rounding in the capacity constraints, certain demand may be allocated to a facility without consuming additional capacity, if the facility has a commodity p block that is not yet completely filled. Furthermore, allocating a new commodity p block is subject to capacity availability at the facility for the entire block.
5. *Reduce unused capacities of open facilities:* For each facility, a dynamic programming algorithm, similar to the one used to solve the Lagrangian subproblem, computes the optimal opening schedule that guarantees sufficient capacity to satisfy the demand allocated to that facility. The algorithm takes into account the total lot size for each commodity reserved at the facility, not only the allocated customer demands.

Even though the resulting solution is integer feasible, its demand allocation may still be improved. Therefore, a final step consists in computing the optimal demand allocation for the current opening schedule using the CPLEX network algorithm.

4.4. Restricted MIP model

To improve the final solution quality, we may use a restricted MIP based on the convexified solutions provided by the bundle method (see Section 4.2). The restricted MIP model for the DFLP_RPC needs to decide for the level of open and existing capacity levels for each location and time period. We therefore define the restricted MIP in terms of capacity level pairs (ℓ, n) .

As explained in Section 4.2, the bundle method provides a multiplier θ^s for each Lagrangian solution s such that $\sum_s \theta^s = 1$. The value θ^s can be seen as the likelihood that solution s provides a good opening schedule. We may therefore derive such likelihoods for each of the decisions involved in s . Let $y_{j\ell n t}^p$ be the likelihood that capacity level (ℓ, n) at j and t is a good opening decision, defined as $y_{j\ell n t}^p = \sum_s \theta^s y_{j\ell n t}^{s,p}$, where $y_{j\ell n t}^{s,p}$ is 1 if solution s selects capacity level pair (ℓ, n) for location j at period t . Furthermore, let L_{jt}^R be the set of (ℓ, n) pairs for location j and period t available in the restricted MIP. The restricted MIP is then defined as follows:

- *Selection of capacity levels:* If the decision for location j and period t is not fixed, L_{jt}^R is composed by the n^S capacity level pairs (ℓ, n) that have the highest $y_{j\ell n t}^p$ values, with $y_{j\ell n t}^p > 0.001$.
- *Defining the set of capacity transitions:* Decision variables $y_{\ell_1 \ell_2 n_1 n_2}^{jt}$ are defined for all combinations between (ℓ_1, n_1) and (ℓ_2, n_2) , with $(\ell_1, n_1) \in L_{jt}^R$ and $(\ell_2, n_2) \in L_{j(t+1)}^R$, if available in the original RPCr-GMC formulation.

Relocation decisions are added to the restricted MIP by computing $\hat{w}_{j\ell n t}^p$ and $\check{w}_{j\ell n t}^p$, representing the likelihoods that an outgoing relocation $\hat{w}_{\ell n}^{jt}$ and an incoming relocation $\check{w}_{\ell n}^{jt}$, respectively, are good decisions. We set $\hat{w}_{j\ell n t}^p = \sum_s \theta^s \hat{w}_{j\ell n t}^{s,p}$, where $\hat{w}_{j\ell n t}^{s,p}$ is 1 if solution s relocates a facility with n existing and ℓ open capacity levels from location j to another location at period t . In the same way, we set $\check{w}_{j\ell n t}^p = \sum_s \theta^s \check{w}_{j\ell n t}^{s,p}$, where $\check{w}_{j\ell n t}^{s,p}$ is 1 if solution s relocates

a facility with n existing and ℓ open capacity levels from location j to another location at period t . All relocation variables with their corresponding $\hat{w}_{j\ell n t}^p$ and $\check{w}_{j\ell n t}^p$ greater than or equal to 0.001 are added to the restricted model. To ensure their feasibility with respect to the flow conservation constraints, certain capacity levels are added to the sets of available capacity levels L^R . To be precise, when adding a relocation decision $\check{w}_{\ell n}^{jt}$ to the restricted MIP, capacity level pair $(0, n)$ is added to $L_{j(t-1)}^R$ and capacity level pairs $(0, n)$ and (n, n) are added to L_{jt}^R to ensure that the flow conservation constraints contain the capacity transition variables $y_{\ell_1 \ell_2 n_1 n_2}^{jt}$ that either maintain the facility closed or reopen it at its maximum capacity level n .

5. Computational experiments

In this section, the performance of the Lagrangian heuristic and that of the MIP solver CPLEX for the different problem variants and formulations are evaluated and compared by means of computational experiments. Results are therefore reported for the DFLP_RPC with RUC constraints and for the two simplified problem variants DFLP_PC and DFLP_RPC with classical capacity constraints (17). All CPLEX models include the strong inequalities (15), as well as the ADCs or strengthened ADCs (16) for the problem variants without or with RUC constraints, respectively.

First, we explain how test instances are generated. Then, the solution of the LP relaxation, as well as the integrality gaps of the different problem variants and formulations are analyzed to assess the impact of the additional features, i.e., facility relocation and RUC constraints. CPLEX optimization results are then summarized for each of the problem variants and their formulations, and the 2i and GMC formulations are compared in detail for the DFLP_RPC with RUC constraints. Finally, computational results are presented to compare the performance of the Lagrangian heuristics and that of CPLEX for the DFLP_RPC with RUC constraints.

All mathematical models and the Lagrangian based heuristics have been implemented in C/C++ using the IBM CPLEX 12.6.0 Callable Library. The code has been compiled and executed on openSUSE 11.3. Each problem instance has been run on a single Intel Xeon X5650 processor (2.67 GHz), limited to 24 GB of RAM.

5.1. Test instances

Due to the lack of openly available instance sets that include the input data required by the DFLP_PC and DFLP_RPC with classical and with RUC constraints, we generated instances by extending the scheme described in Jena et al. [19]. In the following, we summarize the general properties of the instances and focus on the information that extend the previous instances. New information concerns the partial facility closing, facility relocation and the RUC constraints. For further details, we refer to the freely accessible online appendix of Jena et al. [19].

Instances have been generated with different numbers of candidate facility locations $|J|$ and customers $|I|$, combining all pairs of $|J| \in \{50, 100, 150, 200, 250\}$ and $|I| \in \{|J|, 4 \cdot |J|\}$. The highest capacity level at any facility, denoted by q , has been selected such that $q \in \{3, 5, 10\}$. Three different networks have been randomly generated on squares of the following sizes: 300 km, 380 km and 450 km. We consider two different demand scenarios. In both scenarios, the demand for each of the customers is randomly generated and randomly distributed over time. The two scenarios differ in their total demand summed over all customers in each time period. In the first scenario, the total demand is similar in each time period. The second scenario assumes that the total

demand follows strong variations along time and therefore varies at each time period.

The number of commodities $|P|$ has been selected such that $|P| \in \{1, 3, 5\}$. The demands for the second to fifth commodities are based on the demand for the first commodity. To be precise, the demand d_p^{it} for $p \geq 2$ is computed as $d_p^{it} = d_1^{it} \cdot r_p \cdot \text{avgDem}_p / \text{avgDem}_1$, where $\text{avgDem}_1 = 10$, $\text{avgDem}_2 = 6$, $\text{avgDem}_3 = 9$, $\text{avgDem}_4 = 5$, $\text{avgDem}_5 = 8$, and $r_p \sim N(1.0, 0.2^2)$ is a normally distributed random variable with mean of 1.0 and standard deviation of 0.2, truncated at 0 such that $r_p \geq 0$. When RUC constraints are used, the demands need to be scaled, since the RUC capacity constraints operate in units of lot sizes s_p . Demands d_p^{it} are therefore divided by the corresponding lot size s_p , which have been set to $s_1 = 6$, $s_2 = 3$, $s_3 = 5$, $s_4 = 2$ and $s_5 = 7$. Fixed costs are given by the construction of facilities or additional capacity and the change of their capacity levels (i.e., closing and reopening). Variable costs are composed of the costs to produce and transport the commodities. All costs include economies of scale with respect to the capacity involved in the corresponding operations. Transportation costs have been computed based on the Euclidean distance between the points, including a small modification that results in a slight clustering effect of the customers close to a facility. The costs to relocate a facility of 1, ..., 10 capacity levels from one location to another are set to 12,823.70, 19,431.90, 24,077.00, 30,247.00, 42,639.10, 48,854.50, 50,579.90, 56,314.10, 67,804.10 and 73,558.50, respectively. The combination of the different properties listed above results in a total of $5 \times 2 \times 3 \times 3 \times 2 \times 3 = 540$ instances. All instances contain ten time periods, which is found to be sufficient to demonstrate capacity changes along time. Note that we assume that the problem instances do not contain initially existing facilities.

5.2. LP relaxation and integrality gaps

We now address the solution of the LP relaxations for the three problem variants and their formulations, and assess their strength by means of their integrality gaps. The comparison is performed on two scopes. Table 1 summarizes, separately for each problem variant and formulation, the results of the LP relaxation solution. The results take into account only those instances for which the LP relaxation has been solved within 12 h of computing time (“# inst”) and include the average integrality gaps (as estimated by the best known upper bounds) and average computing times. Then, Table 2 provides a direct comparison of the integrality gaps by

Table 1
Solution of LP relaxation and average integrality gaps (in %) separately for all problem variants and formulations.

q	2i formulation			GMC based formulation		
	# inst.	Avg % integr. gap	Time (s)	# inst.	Avg % integr. gap	Time (s)
DFLP_PC						
3	180	1.74	335.0	177	0.17	261.1
5	174	3.09	661.8	162	0.44	734.5
10	153	6.86	3175.9	75	1.55	5407.3
All	507	3.75	1304.5	414	0.52	1378.6
DFLP_RPC						
3	180	2.65	891.5	176	0.15	2120.2
5	174	3.96	1997.0	156	0.49	3265.9
10	162	7.47	2785.3	70	1.94	9261.3
All	516	4.60	1858.9	402	0.60	3808.2
DFLP_RPC with RUC constraints						
3	178	2.71	2046.9	161	0.19	3775.4
5	171	4.08	2818.3	150	0.60	6060.4
10	162	8.27	3308.6	69	2.35	11,546.6
All	511	4.93	2705.0	380	0.74	6088.5

Table 2
Comparison of average integrality gaps (in %) for the three problem variants for instances where the optimal integer solution is known and the LP relaxation of all formulations has been solved.

q	Instance size	# inst.	DFLP_PC		DFLP_RPC		DFLP_RPC w/RUC	
			2i	GMC	2i	GMC	2i	GMC
3	50/50	13	3.34	0.57	4.96	0.59	4.97	0.60
	50/200	18	1.06	0.04	1.52	0.05	1.52	0.05
	100/100	17	2.50	0.10	3.93	0.09	3.93	0.09
	100/400	18	0.98	0.01	1.45	0.01	1.45	0.01
	150/150	17	2.08	0.07	3.46	0.05	3.46	0.05
	150/600	14	0.95	0.01	1.46	0.01	1.46	0.01
	200/200	17	1.89	0.05	3.16	0.03	3.16	0.03
	200/800	8	0.84	0.02	1.39	0.02	1.40	0.02
	250/250	17	1.69	0.02	2.97	0.02	2.97	0.02
	250/1000	4	0.81	0.02	1.35	0.03	1.35	0.03
Avg All	143	1.69	0.09	2.69	0.09	2.69	0.09	
5	50/50	4	4.98	0.95	6.22	0.81	6.22	0.81
	50/200	11	1.80	0.11	2.29	0.10	2.30	0.11
	100/100	9	3.47	0.18	4.86	0.22	4.86	0.22
	100/400	11	1.58	0.01	2.07	0.01	2.07	0.01
	150/150	6	3.07	0.10	4.55	0.07	4.55	0.07
	150/600	4	1.49	0.01	2.11	0.01	2.11	0.01
	200/200	7	2.67	0.07	3.95	0.02	3.95	0.02
	200/800	1	1.64	0.01	2.34	0.01	2.34	0.01
	250/250	5	2.24	0.02	3.60	0.03	3.60	0.02
	250/1000	2	1.47	0.02	2.16	0.03	2.16	0.03
Avg All	60	2.45	0.13	3.41	0.12	3.41	0.12	
10	100/400	1	3.01	0.03	3.10	0.00	3.14	0.04
	Avg All	1	3.01	0.03	3.10	0.00	3.14	0.04
All	Avg All	204	1.92	0.10	2.90	0.10	2.91	0.10

focusing on problem instances that have been solved by all formulations. The reported values are computed over the instances of the same (#facilities/#customers) pairs for which all six formulations have solved the LP relaxation within 12 h of computing time and for which the optimal integer solution is known within 0.1%.

Comparison between problem variants: The separate results in Table 1 and the results for the same set of instances in Table 2 indicate that both the computing times for solving the LP relaxation and the integrality gaps tend to increase as they become more complex (for the 2i formulation as relocation is added and for both formulations as RUC constraints are added). As a consequence, one may expect that solving the MIP models also becomes more difficult as the problems' complexity increases.

Comparison between 2i and GMC formulations: It can be observed that the LP relaxation of the 2i formulation is generally easier to solve than for the GMC formulation, but the latter seems to provide substantially better bounds. When comparing the integrality gaps between the 2i and GMC formulation for the same problem variant in Table 2, the results demonstrate the same trend for all three problem variants: the GMC based formulations provide significantly lower integrality gaps, on average 19–30 times lower than those provided by the 2i formulations.

5.3. CPLEX optimization

As illustrated above, the LP relaxation of the 2i formulations is solved much faster than that of the GMC formulations. However, the latter provides a substantially better bound. Furthermore, the difficulty of solving the LP relaxation increases, as the problem variant becomes more complex. We now compare the different problem variants and formulations in terms of their ability to find the optimal integer solutions. As before, the comparison is performed on two scopes. First, Table 3 presents the results separately for each of the problem variants and formulations on the total set of 540 instances. The results indicate the number of instances for

Table 3
CPLEX results separately for all problem variants and formulations.

q	2i formulation				GMC based formulation			
	# inst.	Avg gap%	Max gap%	Time (s)	# inst.	Avg gap%	Max gap%	Time (s)
DFLP_PC								
3	159	0.41	36.02	1788.1	159	0.02	1.09	694.3
5	126	0.41	6.61	3626.5	107	0.19	2.67	2426.6
10	38	9.41	91.09	6321.6	25	1.52	3.78	6195.4
All	323	1.47	91.09	3038.6	291	0.21	3.78	1803.9
DFLP_RPC								
3	133	0.50	4.98	1633.0	155	0.33	5.26	802.1
5	45	0.86	3.91	3172.5	109	0.77	13.59	1891.5
10	1	1.00	1.00	865.0	21	1.91	4.12	5654.8
All	179	0.59	4.98	2013.3	285	0.61	13.59	1576.3
DFLP_RPC with RUC constraints								
3	101	0.23	4.86	2850.1	154	0.10	3.71	2036.3
5	29	0.27	3.64	3172.5	86	0.86	13.59	3673.5
10	1	0.28	0.28	7201.0	11	1.59	3.33	7200.5
All	131	0.24	4.86	2954.7	251	0.43	13.59	2823.6

Table 4
CPLEX results comparing the two formulations for the DFLP_RPC with RUC constraints, considering instances where both formulations found feasible solutions.

q	Instance size	# inst.	RPCr-2i			RPCr-GMC		
			Avg gap %	Max gap %	Time (s)	Avg gap %	Max gap %	Time (s)
3	50/50	16	0.20	1.48	2561.5	0.17	1.50	2393.1
	50/200	18	0.00	0.01	1398.4	0.00	0.01	733.3
	100/100	16	0.08	1.22	1795.3	0.02	0.21	854.9
	100/400	12	0.16	1.92	2968.2	0.00	0.01	529.3
	150/150	13	0.01	0.02	2324.5	0.00	0.01	459.8
	150/600	6	1.65	4.75	7200.5	0.01	0.02	2682.5
	200/200	13	0.04	0.20	3547.5	0.22	2.79	1721.2
	250/250	7	0.33	1.29	5403.0	0.00	0.01	1443.4
	Avg All	101	0.19	4.75	2850.1	0.06	2.79	1248.2
	5	50/50	6	0.52	2.32	3819.7	0.19	0.61
50/200		8	0.00	0.01	1762.8	0.00	0.01	347.1
100/100		5	0.01	0.01	1972.6	0.01	0.01	369.6
100/400		3	0.12	0.35	3744.3	0.00	0.00	745.0
150/150		4	0.06	0.22	4819.3	0.04	0.15	2395.5
200/200		2	0.00	0.00	3705.0	0.00	0.01	1245.5
Avg All		28	0.13	2.32	3028.7	0.05	0.61	1234.9
10		50/200	1	0.28	0.28	7201.0	0.34	0.34
	Avg All	1	0.28	0.28	7201.0	0.34	0.34	7200.0
All	Avg All	130	0.18	4.75	2922.0	0.06	2.79	1291.1

which CPLEX found feasible solutions (“# inst”) within the time limit of two hours, the average and maximum optimality gaps proven by CPLEX, and the average computing times. Then, Table 4 focuses on the results for the DFLP_RPC with RUC constraints, comparing the two formulations for the instances for which both formulations found feasible solutions within the given time limit of two hours.

Comparison between problem variants: As can be observed in Table 3, the number of instances for which CPLEX found feasible solutions decreases for both formulations as the problem variant gets more complex: for the 2i formulation, 323, 179 and 131 feasible solutions are found for the three problem variants, respectively; for the GMC formulation, 291, 285 and 251 feasible solutions are found, respectively. These results confirm the increasing difficulty of solving the different problem variants as facility relocation and RUC constraints are added to the problem.

Comparison between 2i and GMC formulations: When comparing the GMC formulations to the 2i formulations, the results indicate that the former generally facilitate the solution of the problems. For the DFLP_RPC without and with RUC constraints, the GMC based formulations lead to significantly higher numbers of instances with feasible solutions than the 2i formulations (285 vs. 179 instances for the DFLP_RPC and 251 vs. 131 for the DFLP_RPC with RUC constraints). Comparing on the same set of instances for the DFLP_RPC with RUC constraints in Table 4, the GMC formulation clearly outperforms the 2i formulation regarding the average and maximum deviations from the best known lower bound, as well as the computing times. A further analysis for the simplified problem variants has shown similar results. For the DFLP_PC, CPLEX found feasible solutions for both the PC-2i and the PC-GMC formulations for 280 instances. The PC-2i formulation reports an average optimality gap of 1.09%, a maximum gap of 90.83% and an average solution time of 2768 s, whereas the PC-GMC formulation results in a lower average gap of 0.16%, a maximum gap of 3.78% and an average solution time of 1613 s. For the DFLP_RPC, CPLEX found feasible solutions for both formulations for 179 instances. The RPC-2i formulation resulted in an average gap of 0.44%, a maximum gap of 4.83% and an average solution time of 1956 s. Again, the RPC-GMC formulation reported superior performance with an average gap of 0.21% and a maximum gap of 1.39% in an average solution time of only 592 s. Based on these results, it can be concluded that the GMC based formulations provide a clear advantage in terms of solution quality and computing time when compared to the traditional 2i formulations.

5.4. Performance of the Lagrangian heuristic

We now present results for the Lagrangian heuristics and compare their performance to CPLEX. When using RUC constraints, the solution of the Lagrangian subproblem is more difficult and consumes significantly more computing time. However, it is likely that the problem variant with RUC constraints selects similar facility locations in their optimal solutions as the problem variant without RUC constraints. The Lagrangian heuristics presented in this section therefore initialize the Lagrange multipliers by first solving the problem variant without RUC constraints and then by solving the problem variant with RUC constraints. The initialization phase without RUC constraints is terminated after a maximum of 300 iterations of the bundle method or when the best upper bound lies within 1% of the best known lower bound. Note that, even though we solve the subproblem for the problem variant without RUC constraints, we generate upper bounds for the problem variant with RUC constraints. Furthermore, note that the lower bounds from the initialization phase are also valid for the problem variant with RUC constraints. After the initialization phase, the original problem is solved by the bundle method, limited to a maximum of 500 iterations (including the iterations performed in the initialization phase). In a final optimization phase, we solve the restricted MIP to improve the solution quality.

The following experiments allow for a total of three hours of computing time. For all experiments, a 0.01% optimality stopping criterion has been used. Table 5 presents the results for the Lagrangian heuristic applied to the DFLP_RPC with RUC constraints. The results are given for two configurations of the heuristic. The first configuration uses only the bundle method, whereas the second configuration adds the restricted MIP model afterwards. The restricted MIP has been used with parameter $n^s = 10$. For the second configuration, the restricted MIP is started after two hours at the latest. Adding the restricted MIP significantly improves the average solution quality, while the maximum optimality gap and average computing time remain similar. The lines at the bottom of the table provide results separating the

Table 5
Results for the Lagrangian heuristic for all 540 instances for the DFLP_RPC with RUC constraints.

q	Instance size	Bundle only				Bundle + R-MIP			
		Avg gap %	Max gap%	Time (s)	# 1% opt	Avg gap%	Max gap%	Time (s)	# 1% opt
3	50/50	3.77	11.64	325.4	1	0.31	1.22	4142.9	9
	50/200	0.78	1.54	499.1	15	0.53	0.96	268.3	18
	100/100	1.75	6.14	858.7	8	0.40	0.87	273.2	18
	100/400	0.85	2.82	914.9	16	0.59	0.92	650.8	18
	150/150	1.86	5.13	1559.9	8	0.29	0.99	610.8	18
	150/600	1.02	2.86	1291.0	14	0.70	0.98	760.5	18
	200/200	1.89	6.40	2576.7	10	0.42	0.94	1098.2	18
	200/800	1.47	4.70	2479.1	11	0.51	1.00	1707.8	18
	250/250	1.54	6.44	2686.9	10	0.40	0.93	1381.2	18
	250/1000	1.88	4.67	5942.3	6	0.29	1.00	4081.3	18
5	Avg All	1.68	11.64	1913.4	99	0.44	1.22	1497.5	171
5	50/50	9.18	20.01	639.5	1	1.70	4.57	6763.3	2
	50/200	1.80	4.31	3259.6	5	0.49	1.32	1907.4	16
	100/100	6.21	20.14	2382.9	1	0.73	2.26	3825.7	10
	100/400	1.16	3.30	2798.1	13	0.50	0.99	1605.0	18
	150/150	5.21	9.84	4834.0	1	0.60	1.50	1943.1	15
	150/600	1.09	2.98	3419.6	10	0.43	0.95	2339.4	18
	200/200	3.99	9.31	6061.1	2	0.45	1.20	2948.4	17
	200/800	1.71	4.96	4154.0	9	0.53	0.98	2900.0	18
	250/250	3.15	8.80	5849.7	5	0.35	0.98	3313.8	18
	250/1000	2.05	4.35	7494.4	5	0.30	0.97	5024.0	18
	Avg All	3.56	20.14	4089.3	52	0.60	4.57	3224.8	150
10	50/50	15.93	27.86	1770.1	0	6.38	20.19	7404.5	0
	50/200	9.80	18.18	5830.8	0	3.45	18.18	8189.3	2
	100/100	17.78	29.99	5134.7	0	4.99	20.86	8427.6	0
	100/400	5.85	13.43	7882.2	2	2.24	13.43	6213.4	12
	150/150	18.42	29.05	7207.3	0	6.04	29.05	9140.5	1
	150/600	5.03	10.94	8013.6	3	2.88	10.94	6707.2	11
	200/200	17.15	26.92	8040.9	0	5.16	26.26	9010.6	1
	200/800	4.65	11.91	9568.7	0	2.96	11.91	7356.3	12
	250/250	15.21	27.29	8357.2	0	6.26	27.29	8666.2	4
	250/1000	3.05	5.68	10,893.3	1	0.86	5.68	7598.8	15
	Avg All	11.29	29.99	7269.9	6	4.11	29.05	7864.3	58
All	P = 1	4.08	21.63	1869.0	42	0.56	3.46	1372.5	147
	P = 3	5.54	27.86	5019.5	60	1.32	17.95	4994.2	126
	P = 5	6.91	29.99	6384.1	55	3.28	29.05	6247.4	106
	Avg All	5.51	29.99	4424.2	157	1.72	29.05	4194.1	379

instances by number of commodities. As it may be expected, the problem difficulty increases as the problem contains more commodities. The last column for each configuration indicates the number of instances for which the heuristic proved optimality within 1%. Given the strong lower bounds provided by the heuristic, the heuristic proves optimality within 1% for 379 of the 540 instances.

Table 6 provides a direct comparison between the results of CPLEX and the Lagrangian heuristic. Column “#ns” indicates the number of problem instances for which CPLEX did not find any feasible solution. The MIP solver does not find feasible solutions for about half of the instances, in particular for those with a high number of capacity levels, i.e., $q=10$. For the instances where CPLEX finds feasible solutions, the Lagrangian heuristic provides more reliable results, having a smaller maximum deviation of the provided solution value from the best known lower bound, while the computing times are significantly lower. Interestingly, for those instances, the number of commodities impacts less on the solution quality. Given that the model size increases with the number of commodities, the increase in computing time is as expected. Finally, given the strong lower bounds provided by the heuristic, the latter is capable to prove optimality within 1% for almost the same number of instances as CPLEX (218 vs. 234).

6. Conclusions

We have considered a recently introduced multi-period facility location problem with multiple capacity levels and multiple commodities. This problem, motivated by an industrial application in forestry, allows for several ways to adjust capacity along time, such as the expansion of capacity and the relocation of facilities. As for many problems motivated by industrial applications, the features of the problem go beyond classical variants and significantly complicate the solution. In particular, the problem extends classical facility location by considering the partial closing and reopening of facilities, as well as capacity constraints that consider rounding on the left-hand side of the constraints.

In this paper, we proposed a new formulation for this problem, as well as for two simplified variants without relocation and with classical capacity constraints. For our test instances, the proposed formulations provide significantly lower integrality gaps than previous formulations, on average 19–30 times lower. As a consequence, MIP solvers perform much better when using the new formulation. Next to computational advantages, the proposed modeling technique also allows for a better representation of the cost structure of the problem.

Lagrangian relaxation based heuristics have been developed to find high quality solutions for the problems. While the Lagrangian

Table 6

Comparison of solution quality for CPLEX and the Lagrangian heuristic considering instances for the DFLP_RPC with RUC constraints where CPLEX found feasible solutions.

q	Instance size	CPLEX					Bundle + R-MIP				
		# ns	Avg gap%	Max gap%	Time (s)	# 1% opt	Avg gap%	Max gap %	Time (s)	# 1% opt	
3	50/50	0	0.23	1.21	4327.7	16	0.31	1.22	4142.9	9	
	50/200	0	0.00	0.01	751.2	18	0.53	0.96	268.3	18	
	100/100	0	0.01	0.18	1325.3	18	0.40	0.87	273.2	18	
	100/400	0	0.00	0.01	719.8	18	0.59	0.92	650.8	18	
	150/150	0	0.02	0.26	1665.1	18	0.29	0.99	610.8	18	
	150/600	1	0.01	0.02	3017.2	17	0.69	0.98	756.5	17	
	200/200	0	0.12	2.04	2949.2	17	0.42	0.94	1098.2	18	
	200/800	9	0.40	3.48	6120.1	8	0.22	0.81	1411.7	9	
	250/250	0	0.10	1.79	2978.1	17	0.40	0.93	1381.2	18	
	250/1000	15	0.84	2.51	8152.7	2	0.05	0.11	1400.7	3	
	Avg All	25	0.10	3.48	2553.1	149	0.43	1.22	1170.5	146	
5	50/50	3	2.15	8.39	8005.7	9	1.19	3.28	5831.7	2	
	50/200	0	0.45	3.43	4956.9	16	0.49	1.32	1907.4	16	
	100/100	4	0.36	1.48	4955.4	12	0.42	1.29	2104.2	10	
	100/400	7	0.01	0.03	2504.8	11	0.48	0.97	653.5	11	
	150/150	7	0.92	7.56	6093.3	9	0.34	0.89	812.4	10	
	150/600	12	0.04	0.19	5291.7	6	0.20	0.77	651.0	6	
	200/200	10	0.09	0.39	4031.4	8	0.16	0.50	800.8	8	
	200/800	17	0.01	0.01	4005.0	1	0.84	0.84	1145.0	1	
	250/250	12	0.04	0.14	5347.5	6	0.18	0.73	557.7	6	
	Avg All	90	0.63	8.39	5259.5	78	0.50	3.28	1950.8	70	
	10	50/50	12	1.78	4.07	10,800.2	3	1.85	2.81	611.5	0
50/200		15	0.71	1.51	10,110.0	3	0.60	1.21	264.3	2	
100/100		13	2.88	8.82	10,800.4	1	1.92	3.14	2254.6	0	
Avg All		166	1.94	8.82	10,652.4	7	1.61	3.14	1123.9	2	
All	All $ P = 1$	62	0.30	8.82	3217.0	109	0.47	3.14	390.6	101	
	All $ P = 3$	100	0.23	6.57	3602.7	76	0.50	2.39	1693.7	70	
	All $ P = 5$	119	0.74	8.39	5744.1	49	0.63	3.28	3169.4	47	
	Avg All	281	0.38	8.82	3931.3	234	0.52	3.28	1433.1	218	

relaxation could have been based on the existing 2i formulation, the use of the new GMC based formulation has the important advantage of providing sharp bounds on the optimal integer solution value and therefore on the quality of the provided solutions.

The Lagrangian heuristics consist of two optimization phases. In the first phase, the Lagrangian dual is solved by a bundle method, providing lower and upper bounds. Then, a restricted MIP model is solved to improve the final solution quality. Even though the relocation of facilities, as well as the particular capacity constraints represent an additional obstacle when decomposing the problem, we demonstrate how to efficiently construct feasible facility relocations after relaxing the relocation linking constraints and how to solve the round-up capacity constraints in a combinatorial manner.

Computational results show that the proposed heuristics outperform state-of-the-art MIP solvers, providing better average and maximum deviations from the best known lower bounds in significantly shorter computing times. While the MIP solver does not find feasible solutions for about half of the instances, the heuristics are able to provide high quality solutions for all instances. The average deviation from the best known lower bounds for all 540 test instances is 1.72% for the original problem considering relocation and round-up capacity constraints.

Acknowledgments

The authors would like to thank Antonio Frangioni and Enrico Gorgone for providing the implementation of the bundle method, as well as their valuable advice on its use. The authors are also grateful to Mitacs, the Natural Sciences and Engineering Research Council of Canada (NSERC) (Grant no. 2014-04959) and the Fonds

de recherche du Québec Nature et Technologies (FRQNT) for their financial support. Finally, they also thank Calcul Québec and Compute Canada for providing the computing infrastructure used for the experiments, and two anonymous referees for their insightful comments on an earlier version of this paper.

References

- [1] Antunes AP, Peeters D. On solving complex multi-period location models using simulated annealing. *Eur J Oper Res* 2001;130(1):190–201.
- [2] Ballou RH. Dynamic warehouse location analysis. *J Marketing Res* 1968;5(3):271–6.
- [3] Barcelo J, Hallejord Å, Fernandez E, Jörnsten K. Lagrangean relaxation and constraint generation procedures for capacitated plant location problems with single sourcing. *OR Spektrum* 1990;12(2):79–88.
- [4] Beasley JE. Lagrangean heuristics for location problems. *Eur J Oper Res* 1993;65(3):383–99.
- [5] Borghetti A, Frangioni Antonio, Lacalandra F, Nucci Ca. Lagrangian heuristics based on disaggregated bundle methods for hydrothermal unit commitment. *IEEE Trans Power Syst* 2003;18(1):313–23.
- [6] Brotcorne L, Laporte G, Semet F. Ambulance location and relocation models. *Eur J Oper Res* 2003;147(3):451–63.
- [7] Canel C, Khumawala BM, Law J, Loh A. An algorithm for the capacitated, multi-commodity multi-period facility location problem. *Comput Oper Res* 2001;28(5):411–27.
- [8] Chardaire P, Sutter A, Costa M-C. Solving the dynamic facility location problem. *Networks* 1996;28(2):117–24.
- [9] Correia I, Captivo ME. A Lagrangean heuristic for a modular capacitated location problem. *Ann Oper Res* 2003;122:141–61.
- [10] Dias J, Captivo ME, Clímaco J. Capacitated dynamic location problems with opening, closure and reopening of facilities. *IMA J Manag Math* 2006;17(4):317–48.
- [11] Frangioni A. About Lagrangian methods in integer optimization. *Ann Oper Res* 2005;139(1):163–93.
- [12] Frangioni A, Gallo G. A bundle type dual-ascent approach to linear multi-commodity min-cost flow problems. *INFORMS J Comput* 1999;11(4):370–93.
- [13] Gendron B, Crainic TG. Relaxations for multicommodity capacitated network design problems. Technical report, Publication CRT-945, Centre de recherche sur les transports, Université de Montréal; 1994.

- [14] Geoffrion AM. Lagrangean relaxation for integer programming. *Math Progr Study* 1974;2(2):82–114.
- [15] Görtz S, Klose A. A simple but usually fast branch-and-bound algorithm for the capacitated facility location problem. *INFORMS J Comput* 2012;24(4):597–610.
- [16] Gouveia L, Saldanha da Gama F. On the capacitated concentrator location problem: a reformulation by discretization. *Comput Oper Res* 2006;33(5):1242–58.
- [17] Jena SD, Cordeau J-F, Gendron B. Modeling and Solving a Logging Camp Location Problem. *Annals of Operations Research* 2015;232(1):151–77.
- [18] Jena SD, Cordeau J-F, Gendron B. Dynamic Facility Location with Generalized Modular Capacities. *Transportation Science*. 2015;49(3):484–99.
- [19] Jena SD, Cordeau J-F, Gendron B. Lagrangian heuristics for large-scale dynamic facility location with generalized modular capacities. Technical report CIR-RELT-2014-21, Centre interuniversitaire de recherche sur les reseaux d'entreprise, la logistique et le transport; 2014b.
- [20] Kim D-G, Kim Y-D. A Lagrangian heuristic algorithm for a public healthcare facility location problem. *Ann Oper Res* 2013;206(1):221–40.
- [21] Lee CY. An optimal algorithm for the multiproduct capacitated facility location problem with a choice of facility type. *Comput Oper Res* 1991;18(2):167–82.
- [22] Lee D-H, Dong M. A heuristic approach to logistics network design for end-of-lease computer products recovery. *Transp Res Part E: Logist Transp Rev* 2008;44(3):455–74.
- [23] Luss H. Operations research and capacity expansion problems: a survey. *Oper Res* 1982;30(5):907–47.
- [24] Martello S, Toth P. Knapsack problems: algorithms and computer implementations. In: *Interscience Series in Discrete Mathematics and Optimization*, New York, NY, USA: John Wiley & Sons, Inc.; 1990.
- [25] Melo MT, Nickel S, Saldanha da Gama F. Dynamic multi-commodity capacitated facility location: a mathematical modeling framework for strategic supply chain planning. *Comput Oper Res* 2006;33(1):181–208.
- [26] Melo MT, Nickel S, Saldanha da Gama F. A tabu search heuristic for redesigning a multi-echelon supply chain network over a planning horizon. *Int J Prod Econ* 2011;136(1):218–30.
- [27] Min H, Melachrinoudis E. The relocation of a hybrid manufacturing/distribution facility from supply chain perspectives: a case study. *Omega* 1999;27(1):75–85.
- [28] Schütz P, Stougie L, Tomasgard A. Stochastic facility location with general long-run costs and convex short-run costs. *Comput Oper Res* 2008;35(9):2988–3000.
- [29] Shulman A. An algorithm for solving dynamic capacitated plant location problems with discrete expansion sizes. *Oper Res* 1991;39(3):423–36.
- [30] Snyder LV. Facility location under uncertainty: a review. *IIE Trans* 2006;38(7):547–64.
- [31] Vahidnia MH, Alesheikh AA, Alimohammadi A. Hospital site selection using fuzzy AHP and its derivatives. *J Env Manag* 2009;90(10):3048–56.
- [32] Van Roy TJ. A cross decomposition algorithm for capacitated facility location. *Oper Res* 1986;34(1):145–63.
- [33] Van Roy TJ, Erlenkotter D. A dual-based procedure for dynamic facility location. *Manag Sci* 1982;28(10):1091–105.
- [34] Warszawski A. Multi-dimensional location problems (1970–1977). *Oper Res Q* 1973;24(2):165–79.
- [35] Wentges P. Accelerating Benders' decomposition for the capacitated facility location problem. *Math Methods Oper Res* 1996;44(2):267–90.
- [36] Wesolowsky GO. Dynamic facility location. *Manag Sci* 1973;19(11):1241–8.
- [37] Wesolowsky GO, Truscott WG. The multiperiod location-allocation problem with relocation of facilities. *Manag Sci* 1975;22(1):57–65.
- [38] Wu L, Zhang X, Zhang J. Capacitated facility location problem with general setup cost. *Comput Oper Res* 2006;33(5):1226–41.