Submitted to *INFORMS Journal on Computing*
manuscript (Please, provide the mansucript number!)

Authors are encouraged to submit new papers to INFORMS journals by means of a style file template, which includes the journal title. However, use of a template does not certify that the paper has been accepted for publication in the named journal. INFORMS journal templates are for the exclusive purpose of submitting to an INFORMS journal and should not be used to distribute the papers in print or online or to submit the papers to another publication.

Lagrangian Heuristics for Large-Scale Dynamic Facility Location with Generalized Modular Capacities

Sanjay Dominik Jena
Centre interuniversitaire de recherche sur les réseaux d'entreprise, la logistique et le transport (CIRRELT) and Département d'informatique et de recherche opérationnelle, Université de Montréal, sanjay.jena@cirrelt.ca,

Jean-François Cordeau
Centre interuniversitaire de recherche sur les réseaux d'entreprise, la logistique et le transport (CIRRELT) and Canada Research Chair in Logistics and Transportation, HEC Montréal, jean-francois.cordeau@hec.ca,

Bernard Gendron
Centre interuniversitaire de recherche sur les réseaux d'entreprise, la logistique et le transport (CIRRELT) and Département d'informatique et de recherche opérationnelle, Université de Montréal, bernard.gendron@cirrelt.ca

We consider the Dynamic Facility Location Problem with Generalized Modular Capacities, a multi-period facility location problem in which the costs for capacity changes are defined for all pairs of capacity levels. The problem embeds a complex cost structure and generalizes several existing facility location problems, such as those that allow temporary facility closing or capacity expansion and reduction. As the model may grow very large, general-purpose mixed-integer programming (MIP) solvers are limited to solving instances of small to medium size. In this paper, we extend the generalized model to the case of multiple commodities. We propose Lagrangian heuristics, based on subgradient and bundle methods, to find good quality solutions for large-scale instances with up to 250 facility locations and 1000 customers. To improve the final solution quality, a restricted MIP model is solved based on the information collected through the solution of the Lagrangian dual. Computational results show that the Lagrangian based heuristics provide highly reliable results for all problem variants considered. They produce good quality solutions in short computing times even for instances where state-of-the-art MIP solvers do not find feasible solutions. The strength of the formulation also allows the method to provide tight bounds on the optimal value.

1. Introduction

Dynamic facility location problems aim at providing capacity planning over a multiple-period planning horizon. Given that customer demands may vary significantly over time, facilities often adjust their capacities. These problems find applications in both the public and private sectors, for the location of production facilities (Fleischmann et al. 2006), schools (Peeters and Antunes 2001), health care facilities (Correia and Captivo 2003, Kim and Kim 2013) and many more, as documented in several recent literature surveys (Thomas and Griffin 1996, Brotcorne et al. 2003, Revelle et al. 2008, Melo et al. 2009, Smith et al. 2009). To represent the adjustment of capacities in such problems, common actions include capacity expansion and reduction (Luss 1982, Jacobsen 1990, Peeters and Antunes 2001, Troncoso and Garrido 2005, Dias et al. 2007), temporary closing of facilities (Chardaire and Sutter 1996, Canel et al. 2001, Dias et al. 2006) and the relocation of facilities (Melo et al. 2006). Although mathematical models often take into account complex environments such as complete supply chains, the cost structure to adjust capacities along time is commonly modeled in less detail. Economies of scale are often represented on the level of the total capacity involved in each operation, but do not take into consideration the capacity level before the change. A more detailed representation of the cost structure is necessary in a number of applications, especially in the domains of transportation, logistics and telecommunications, where additional capacity gets cheaper (or more expensive) when approaching the maximum capacity limit. For instance, in the problem introduced by Jena et al. (2012), logging camps are located to host workers in the forest industry. In this problem, the total capacity of a camp is represented by its number of different hosting units, while additional units provide supporting infrastructure. As the relation between the number of different units cannot be captured by a simple function, the costs for capacity changes need to be described in a transition matrix.

Jena et al. (2013) recently introduced the Dynamic Facility Location Problem with Generalized Modular Capacities (DFLPG), in which the costs for capacity changes are based on a cost matrix. The mixed-integer programming (MIP) model presented by the authors therefore allows taking into account not only the total capacity involved in the capacity change, but also the current capacity level. This model generalizes several multi-period facility location problems: the problem with facility closing and reopening, the problem with capacity expansion and reduction, and the combination of the two. In addition, the DLFPG provides a strong linear programming (LP) relaxation bound. Compared to alternative MIP formulations, the DFLPG based models can often be solved twice as fast using a general-purpose MIP solver. Although it is possible to solve the models for small and medium size instances, they usually grow too large when considering more complex problem variants or larger instances. In this case, heuristics are an interesting alternative. They also provide an advantage when performing "what-if" analysis, which requires repeatedly solving the problem with different scenarios. Heuristics are usually capable of using solutions for a certain scenario to quickly find solutions for a different one.

Metaheuristics such as tabu search, simulated annealing and genetic algorithms have been frequently applied to several families of location problems, from classical facility location problems (Arostegui et al. 2006) to logistics network design that model entire supply chains (Lee and Dong 2008, Melo et al. 2011). Lagrangian relaxation based heuristics have been developed for several variants of single-period facility location problems (Barcelo et al. 1990, Sridharan 1991, Beasley 1993, Sridharan 1995, Holmberg and Ling 1997, Agar and Salhi 1998, Holmberg and Yuan 2000, Correia and Captivo 2003, Wu et al. 2006), some of which combined Langrangian relaxation and local search (Correia and Captivo 2006, Li et al. 2009). Lagrangian bounds have also been used within exact methods (Görtz and Klose 2012). For multi-period facility location, approaches based on Lagrangian relaxation have been proposed for problems without capacities (Chardaire and Sutter 1996), with fixed capacities (Shulman 1991), and for multi-echelon problems in the context of supply chain design (Diabat et al. 2011). Furthermore, Lagrangian based methods have been successfully applied to other location problems such as dynamic hub location (Elhedhli and Wu 2010, Contreras et al. 2011).

In this paper, we present an extension of the DFLPG in which customers have demands for different commodities. We propose Lagrangian based heuristics that find good quality solutions in reasonable computing times. Two methods are introduced to solve the Lagrangian dual: a subgradient method and a bundle method. After this process, a second optimization step is used to improve the solution quality. This step consists of solving a restricted MIP model, taking into consideration only decisions that have been part of a significant number of the previous Lagrangian solutions. To the best of our knowledge, this work is the first to present a Lagrangian relaxation approach to solve large-scale instances of a multi-period facility location problem of this nature, i.e., including modular capacity levels and multiple commodities. Given the strength of the formulation used to model the DFLPG, the Lagrangian heuristics are capable of providing relatively tight bounds on the optimal value. The results are stable even for large instances, where general-purpose MIP solvers either consume too much memory or do not solve the problem in reasonable time. Given the generality of the DFLPG, the proposed heuristic can handle an entire class of problems, consisting of all those that can be modeled by the DFLPG.

The remainder of the paper is organized as follows. Section 2 reviews and extends the MIP formulation for the DFLPG and shows how it can be used to model three different special cases. Section 3 explains how the problem is decomposed via Lagrangian relaxation and outlines the resulting heuristics. Section 4 then discusses how the final solution quality can be improved in a second optimization phase, using information from the solution of the Lagrangian dual to generate a restricted MIP model. The Lagrangian heuristics are then compared by means of computational experiments in Section 5. First, general results are presented for each of the different problem variants. Then, the advantages of the Lagrangian heuristics are illustrated with more detailed results comparing their performance to a state-of-the-art MIP solver. Finally, conclusions are drawn and future research directions are discussed in Section 6.

2. Mixed Integer Programming Formulation

This section first introduces a general formulation for the DFLPG and then explains how it can be used to model three special cases.

2.1. General Model

We consider the mixed-integer programming formulation introduced by Jena et al. (2013) and extend it to include multiple commodities. We denote by J the set of candidate 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 $T=\{1,2,\dots,|T|\}$ 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. Additionally, we denote by $P=\{1,2,\dots,|P|\}$ the set of different commodities. The demand of customer i for commodity p in period t is denoted by d_{ipt}, while the cost to serve one unit of commodity p from facility j operating at capacity level ℓ to customer i during period t is denoted by $g_{ij\ell pt}$. The capacity of a facility of size ℓ at location j is given by $u_{j\ell}$ (with $u_{j0}=0$). The cost matrix $f_{j\ell_1\ell_2 t}$ describes the combined cost to change the capacity level of a facility at location j from ℓ_1 to ℓ_2 at the beginning of period t and to operate the facility at capacity level ℓ_2 throughout time period t. Furthermore, we let ℓ^j be the initial capacity level of an existing facility at location j.

To formulate the problem, we use binary variables $y_{j\ell_1\ell_2 t}$ equal to 1 if and only if the facility at location j changes its capacity level from ℓ_1 to ℓ_2 at the beginning of period t. The allocation variables $x_{ij\ell pt}$ denote the fraction of the demand of customer i for commodity p in period t that is served from a facility of size ℓ located at j. Using this notation, we define the following MIP model, referred to as the *Generalized Modular Capacities (GMC)* formulation:

$$\text{(GMC)} \quad \min \sum_{j\in J}\sum_{\ell_1\in L}\sum_{\ell_2\in L}\sum_{t\in T} f_{j\ell_1\ell_2 t}y_{j\ell_1\ell_2 t} + \sum_{i\in I}\sum_{j\in J}\sum_{\ell\in L}\sum_{p\in P}\sum_{t\in T} g_{ij\ell t}d_{ipt}x_{ij\ell pt} \tag{1}$$

$$s.t. \sum_{j\in J}\sum_{\ell\in L} x_{ij\ell pt} = 1 \quad \forall i\in I,\ \forall p\in P,\ \forall t\in T \tag{2}$$

$$\sum_{i\in I}\sum_{p\in P} d_{ipt}x_{ij\ell pt} \le \sum_{\ell_1\in L} u_{j\ell}y_{j\ell_1\ell t} \quad \forall j\in J,\ \forall \ell\in L,\ \forall t\in T \tag{3}$$

$$\sum_{\ell_1\in L} y_{j\ell_1\ell(t-1)} = \sum_{\ell_2\in L} y_{j\ell\ell_2 t} \quad \forall j\in J,\ \forall \ell\in L,\ \forall t\in T\setminus\{1\} \tag{4}$$

$$\sum_{\ell_2\in L} y_{j\ell^j\ell_2 1} = 1 \quad \forall j\in J \tag{5}$$

$$x_{ij\ell pt} \geq 0 \quad \forall i \in I,\ \ \forall j \in J,\ \ \forall \ell \in L,\ \ \forall p \in P,\ \ \forall t \in T \tag{6}$$

$$y_{j\ell_1\ell_2 t} \in \{0,1\} \quad \forall j \in J,\ \ \forall \ell_1 \in L,\ \ \forall \ell_2 \in L,\ \ \forall t \in T. \tag{7}$$

The objective function (1) minimizes the total cost for changing the capacity levels and allocating the demand. Constraints (2) are the demand constraints for the customers. Constraints (3) are the capacity constraints at the facilities. Constraints (4) link the capacity change variables in consecutive time periods. Finally, constraints (5) specify that exactly one capacity level must be chosen at the beginning of the planning horizon. Note that the flow constraints (4) and (5) further guarantee that, in each time period, exactly one capacity change variable is selected.

We may also adapt two types of valid inequalities to be used in the GMC formulation:

$$x_{ij\ell pt} \leq \sum_{\ell_1 \in L} y_{j\ell_1\ell t} \quad \forall i \in I,\ \ \forall j \in J,\ \ \forall \ell \in L,\ \ \forall p \in P,\ \ \forall t \in T. \tag{8}$$

$$\sum_{j \in J} \sum_{\ell_1 \in L} \sum_{\ell_2 \in L} u_{j\ell_2} y_{j\ell_1\ell_2 t} \geq \sum_{i \in I} \sum_{p \in P} d_{ipt} \quad \forall t \in T. \tag{9}$$

The *Strong Inequalities (SI)* (8), typically used in facility location and network design problems (see, for instance, Gendron and Crainic 1994), are known to provide a tight upper bound for the demand assignment variables. The SIs may be added to the model either *a priori* or in a branch-and-cut manner only when they are violated in the solution of the LP relaxation. The set of valid inequalities (9) is referred to as the *Aggregated Demand Constraints (ADC)*. Although they are redundant for the LP relaxation, adding them to the model enables MIP solvers to generate cover cuts that further strengthen the formulation.

2.2. Special Cases

We now illustrate how special cases can be modeled by using the GMC formulation. As will be explained in Section 4, our solution approach can be tailored to take advantage of the special structure of each problem variant. Jena et al. (2013) explicitly show how to model two problem variants, using the GMC formulation: facility location with closing and reopening of facilities and facility location with capacity expansion and reduction. In the first problem, the size of the facility is chosen from a discrete set of capacity levels. Existing facilities may then be closed and reopened multiple times. In the second problem considered, capacities can be adjusted by the use of a single facility at each location. At each facility, the capacity can be expanded or reduced from one capacity level to another. It is assumed that an expansion of ℓ capacity levels has always the same cost, regardless of the previous capacity level. These two problems are denoted as the *Dynamic Modular Capacitated Facility Location Problem with Closing and Reopening (DMCFLP_CR)* and the *Dynamic Modular Capacitated Facility Location Problem with Capacity Expansion and Reduction (DMCFLP_ER)*, respectively.

A subset of capacity change variables $y_{j\ell_1\ell_2t}$ is chosen to model these special cases. The cost coefficients $f_{j\ell_1\ell_2t}$ for these variables are based on the following fixed costs, defined to characterize the special cases:

- $c^c_{j\ell}$ and $c^o_{j\ell}$ are the costs to temporarily close and reopen a facility of size ℓ at location j, respectively;
- $f^c_{j\ell}$ and $f^o_{j\ell}$ are the costs to reduce and to expand the capacity of a facility at location j by ℓ capacity levels, respectively;
- $F^o_{j\ell}$ is the cost to maintain an open facility of size ℓ at location j throughout one time period.

For the problem variant involving facility closing and reopening, we create an artificial capacity level $\overline{\ell}$ for each capacity level $\ell \in L\backslash\{0\}$. Capacity level $\overline{\ell}$ represents the state in which a facility of size ℓ is temporarily closed. At each time period $t \in T$ and location $j \in J$, we may find capacity transition decisions $y_{j\ell_1\ell_2t}$ that represent different types of operations (note that the costs for these decisions are usually composed by the cost to perform the capacity transition, as well as the maintenance cost for the new capacity level):

1. Facility construction and capacity expansion. The expansion of the capacity is represented by a capacity transition from capacity level ℓ_1 to any other capacity level $\ell_2 > \ell_1$. If the decision represents a facility construction, then ℓ_1 is 0. The capacity is thus expanded by $\ell_2 - \ell_1$ capacity levels. The cost for this decision is set to $f_{j\ell_1\ell_2t} = f^o_{j(\ell_2-\ell_1)} + F^o_{j\ell_2}$.

2. Capacity reduction. The reduction of the capacity is represented by a transition from capacity level ℓ_1 to any other capacity level $\ell_2 < \ell_1$. The capacity is thus reduced by $\ell_1 - \ell_2$ capacity levels. The cost for this decision is set to $f_{j\ell_1\ell_2t} = f^c_{j(\ell_1-\ell_2)} + F^o_{j\ell_2}$.

3. Maintaining the current capacity level. A facility may neither expand nor reduce the current capacity level. The cost of this transition is thus only composed of the maintenance cost, i.e., $f_{j\ell_1\ell_1t} = F^o_{j\ell_1}$ if the capacity level represents an open facility, $f_{j\bar{\ell}_1\bar{\ell}_1t} = 0$ if the capacity level represents a temporarily closed facility and $f_{j00t} = 0$ if no facility exists.

4. Temporary closing. An open facility of size ℓ_1 can be temporarily closed, i.e., it changes to capacity level $\bar{\ell}_1$. The total cost is $f_{j\ell_1\bar{\ell}_1t} = c^c_{j\ell_1}$.

5. Reopening a closed facility. A temporarily closed facility of size ℓ_1 can be reopened, i.e., it changes its capacity level from $\bar{\ell}_1$ to ℓ_1. The total cost for this decision is $f_{j\bar{\ell}_1\ell_1t} = c^o_{j\ell_1} + F^o_{j\ell_1}$.

The DMCFLP_CR is represented by transition decisions of type 1 (for construction only), 3, 4 and 5. We denote the resulting model as the *CR-GMC* formulation. The DMCFLP_ER is represented by transition decisions of type 1, 2 and 3. The resulting model is denoted as the *ER-GMC* formulation.

Jena et al. (2013) also refer to a third problem variant, which combines both features of the two special cases. It is denoted as the *Dynamic Modular Capacitated Facility Location Problem with Closing/Reopening and Capacity Expansion/Reduction (DMCFLP_CR_ER)*. The problem variant is modeled by using the transition decisions of type $1-5$ presented above. However, these decisions allow only one single operation, for example either capacity reduction or facility closing, at each time period. In practice, it is very likely that one may want to reduce or expand the capacity before closing or after reopening a facility at the same time period. We may therefore consider four additional decision types that represent combinations of such operations:

(a) A facility is reopened at level ℓ_1 and its capacity is expanded to level $\ell_2 > \ell_1$ at the same time period.

(b) A facility is reopened at level ℓ_1 and its capacity is reduced to level $\ell_2 < \ell_1$ at the same time period.

(c) The capacity of a facility at level ℓ_1 is expanded to level $\ell_2 > \ell_1$ and the facility is closed right after.

(d) The capacity of a facility at level ℓ_1 is reduced to level $\ell_2 < \ell_1$ and the facility is closed right after.

By making the realistic assumption that the costs for closing and reopening a facility are non-decreasing as the size of the facility increases, we may discard two of the four possibilities.

PROPOSITION 1. *Let $c^o_{j\ell} \leq c^o_{j(\ell+1)}$ and $c^c_{j\ell} \leq c^c_{j(\ell+1)}$ for $\ell = 0, 1, 2, \ldots, (q-1)$, then there is at least one optimal solution that does neither use decisions of type (b) nor of type (c).*

PROOF. Note that case (c) may only occur in two situations: either the facility stays closed until the end of the planning horizon or the facility is reopened at a later moment. If the facility stays closed, then closing it at level ℓ_1 is at most as expensive as combined capacity expansion and closing as suggested in case (c): $c^c_{j\ell_1} \leq f^o_{j(\ell_2-\ell_1)} + c^c_{j\ell_2}$. If the facility is closed at the beginning of time period t_1, but it will be reopened at the beginning of period $t_2 > t_1$, then the corresponding costs using case (c) are given by: $C^c = c^c_{j\ell_1} + f^o_{j(\ell_2-\ell_1)} + c^o_{j\ell_2} + F^o_{j\ell_2}$. However, the same solution may be reproduced by closing the facility at level ℓ_1 and expanding its capacity only after it has been reopened using case (a), which corresponds to the following costs: $C^a = c^c_{j\ell_1} + c^o_{j\ell_1} + f^o_{j(\ell_2-\ell_1)} + F^o_{j\ell_2}$. Now, because $c^o_{j\ell_1} \leq c^o_{j\ell_2}$, we have: $C^a \leq C^c$. Therefore, a solution using case (a) is at most as expensive as a solution using case (c).

The same can be shown for the relation between cases (d) and (b), where reducing the capacity before temporary closing is as most as costly as reducing the capacity after temporary closing. Q.E.D.

We thus add only the transition decisions given by the cases (a) and (d) to the model:

6. Reopening and capacity expansion. A closed facility of capacity level ℓ_1 is reopened and its capacity is expanded to level ℓ_2 (with $\ell_1 < \ell_2$). The cost for this decision, including the maintenance costs at capacity level ℓ_2 is thus set to $f_{j\overline{\ell_1}\ell_2 t} = c^c_{j\ell_1} + f^o_{j(\ell_2-\ell_1)} + F^o_{\ell_2}$.

7. Capacity reduction and facility closing. An open facility reduces its capacity from level ℓ_1 to level ℓ_2 (with $\ell_1 > \ell_2$) and is temporarily closed afterwards. The cost for this decision, including the maintenance costs at capacity level ℓ_2 is thus set to $f_{j\ell_1\overline{\ell_2}t} = f^c_{j(\ell_1-\ell_2)} + c^c_{j\ell_2}$.

3. Lagrangian Relaxation

When applying Lagrangian relaxation to capacitated facility location problems, it is common to relax either the capacity constraints or the demand constraints. Since relaxing the capacity constraints results in a subproblem that is NP-hard (Van Roy and Erlenkotter 1982, Barcelo et al. 1990), a more promising and popular choice in the literature (e.g., Shulman 1991, Beasley 1993, Wu et al. 2006) is to relax the demand constraints (2), which yields a Lagrangian subproblem that can be solved efficiently. Let α be the vector of Lagrange multipliers. After relaxing the demand constraints (2) and rearranging the terms in the objective function, we obtain the following Lagrangian subproblem:

$$
\begin{aligned}
L(\alpha) = \quad & \min \sum_{j\in J}\sum_{\ell_1\in L}\sum_{\ell_2\in L}\sum_{t\in T} f_{j\ell_1\ell_2t}y_{j\ell_1\ell_2t} \\
& + \sum_{i\in I}\sum_{j\in J}\sum_{\ell\in L}\sum_{p\in P}\sum_{t\in T}(g_{ij\ell pt}d_{ipt} - \alpha_{ipt})x_{ij\ell pt} + \sum_{i\in I}\sum_{p\in P}\sum_{t\in T}\alpha_{ipt} \\
& s.t.\ (3)-(8).
\end{aligned}
$$

Note that the Strong Inequalities (8) are included in the Lagrangian subproblem, since they are easy to handle, as shown next.

3.1. Solution of the Lagrangian Subproblem

Let $\tilde{c}_{ij\ell pt} = g_{ij\ell pt}d_{ipt} - \alpha_{ipt}$ denote the modified costs for the $x_{ij\ell pt}$ variables. We separate the Lagrangian subproblem into $|J|$ independent subproblems, one for each candidate facility location for a fixed set of Lagrangian multipliers α. The Lagrangian subproblem is solved as $L(\alpha) = \sum_{j\in J} L_j(\alpha) + \sum_{i\in I}\sum_{p\in P}\sum_{t\in T}\alpha_{ipt}$, where $L_j(\alpha)$ is defined as follows:

$$
L_j(\alpha) = \quad \min \sum_{\ell_1\in L}\sum_{\ell_2\in L}\sum_{t\in T} f_{j\ell_1\ell_2t}y_{j\ell_1\ell_2t} + \sum_{i\in I}\sum_{\ell\in L}\sum_{p\in P}\sum_{t\in T}\tilde{c}_{ij\ell pt}x_{ij\ell pt}
$$

$$s.t. \sum_{i \in I} \sum_{p \in P} d_{ipt} x_{ij\ell pt} \leq \sum_{\ell_1 \in L} u_{j\ell} y_{j\ell_1 \ell t} \quad \forall \ell \in L, \ \ \forall t \in T$$

$$\sum_{\ell_1 \in L} y_{j\ell_1 \ell (t-1)} = \sum_{\ell_2 \in L} y_{j\ell\ell_2 t} \quad \forall \ell \in L, \ \ \forall t \in T \setminus \{1\}$$

$$\sum_{\ell_2 \in L} y_{j\ell^j \ell_2 1} = 1$$

$$x_{ij\ell pt} \leq \sum_{\ell_1 \in L} y_{j\ell_1 \ell t} \quad \forall i \in I, \ \ \forall \ell \in L, \ \ \forall p \in P, \ \ \forall t \in T$$

$$x_{ij\ell pt} \geq 0 \quad \forall i \in I, \ \ \forall \ell \in L, \ \ \forall p \in P, \ \ \forall t \in T$$

$$y_{j\ell_1 \ell_2 t} \in \{0,1\} \quad \forall \ell_1 \in L, \ \ \forall \ell_2 \in L, \ \ \forall t \in T.$$

Each of these subproblems (one for each location $j \in J$) is concerned with finding the optimal capacity planning over time, i.e., an optimal schedule to open facilities of a certain size such that the total cost composed by demand allocation costs (considering the modified costs $\tilde{c}_{ij\ell pt}$) and the costs to change capacity levels is minimal. We can solve this problem using dynamic programming by adapting the approach presented by Shulman (1991). Let $L_j^\alpha(\ell,t)$ denote the cost for an optimal demand allocation at period t assuming that a facility of size ℓ is available. For a given set of multipliers α, let $O_j^\alpha(\ell,t)$ denote the optimal cost to serve all demands by facility j throughout the time periods $0,\ldots,t$, with a facility of size ℓ at the end of period t. For $\ell > 0, t > 0$, the optimal value of $O_j^\alpha(\ell,t)$ is composed of the costs for demand allocation in period t, the capacity transition to level ℓ, the facility maintenance at level ℓ, and the optimal cost to serve all demands in previous time periods at the capacity level that minimizes the total cost. They can be computed by the following recurrence formula:

$$O_j^\alpha(\ell,t) = L_j^\alpha(\ell,t) + \min_{0 \leq \ell_1 \leq q} \{ f_{j\ell_1 \ell t} + O_j^\alpha(\ell_1, t-1) \}.$$

Note that $L_j^\alpha(0,t) = 0$ since demand cannot be allocated to a facility with capacity level 0. Furthermore, for $t = 0$ the size of the facility that exists at the beginning of the planning horizon is ℓ^j. We therefore have: $O_j^\alpha(\ell,0) = f_{i\ell^j \ell t} + L_j^\alpha(\ell,0)$.

The subproblem is then solved by selecting the facility size at the last time period that has the lowest total cost:

$$L_j(\alpha) = \min_{0 \leq \ell \leq q} \left\{ O_j^\alpha(\ell, |T|) \right\}.$$

Note that, without the use of the SIs, the Lagrangian subproblem does not possess the integrality property (Geoffrion 1974), since facility capacities will only be opened as much as forced by the capacity constraints, i.e., $\sum_{\ell_1 \in L} y_{j\ell_1\ell t} = \sum_{i\in I}\sum_{p\in P}(d_{ipt}x_{ij\ell pt})/u_{j\ell}$, which may be fractional. Adding the SIs to the problem strengthens the dependence between the opening decisions and the demand allocation: $\sum_{\ell_1\in L} y_{j\ell_1\ell t} = \max_{i\in I,p\in P}\{x_{ij\ell pt}\}$. The variables $x_{ij\ell pt}$ (and therefore also one of the corresponding $y_{j\ell_1\ell_2 t}$ variables) will take value 1 if their modified costs $\tilde{c}_{ij\ell pt}$ compensate the costs for the open facility. As a consequence, using the SIs, the Lagrangian subproblem also has the integrality property. The lower bound provided by the Lagrangian dual will therefore never be better than the bound provided by the LP relaxation of the original problem using the SIs.

Computation of the Optimal Demand Allocation. The optimal demand allocation $L_j^\alpha(\ell, t)$ at location j assumes that a facility of size ℓ is available and can be computed by solving a fractional knapsack problem (subject to the capacity constraints and the SIs):

$$\begin{aligned} L_j^\alpha(\ell,t) = \quad & \min \sum_{i\in I}\sum_{p\in P} \tilde{c}_{ij\ell pt} x_{ij\ell pt} \\ & s.t. \ \sum_{i\in I}\sum_{p\in P} d_{ipt} x_{ij\ell pt} \leq u_{j\ell} \\ & \qquad 0 \leq x_{ij\ell pt} \leq 1 \ \ \forall i \in I, \forall p \in P. \end{aligned}$$

This problem can be solved by sorting all x variables in increasing order of their ratio $\tilde{c}_{ij\ell pt}/d_{ipt}$, selecting those with the most negative ratio until the capacity is completely filled or all variables with negative ratios have been selected. To be precise, we repeatedly select the variables with the most negative ratio for $< i, p >$ and increase the variable value to the maximum value possible, updating the remaining knapsack capacity $u'_{j\ell}$ after each variable selection:

$$< i^*, p^* > = \operatorname*{argmin}_{i\in I, p\in P} \left\{ \frac{\tilde{c}_{ij\ell pt}}{d_{ipt}} \right\}, \qquad x_{ij^*\ell p^* t} = \min\left\{1, \frac{u'_{j\ell}}{d_{ipt}}\right\}.$$

Clearly, all other x variables are set to 0.

3.2. Solution of the Lagrangian Dual

The solution of the Lagrangian subproblem, for any choice of the Lagrange multipliers α, provides a lower bound to the DFLPG. To obtain the best possible lower bound, one must solve the Lagrangian dual:

$$z^* = \max_{\alpha} \; L(\alpha).$$

The Lagrangian function $L(\alpha)$ is non-differentiable. However, a subgradient direction can be easily computed. We consider two different methods to solve the Lagrangian dual: a subgradient method and a bundle method.

Subgradient Method. The subgradient direction γ_{ipt} at the $k-th$ iteration is computed as the violation of the relaxed constraints when x is fixed to the values found by solving the Lagrangian subproblem:

$$\gamma_{ipt}^k = 1 - \sum_{j\in J}\sum_{\ell\in L} x_{ij\ell pt} \quad \forall i\in I, \;\; \forall p\in P, \;\; \forall t\in T.$$

We choose the step size λ^k at iteration k as suggested by Held et al. (1974) and often used in other works (Shulman 1991, Sridharan 1991, Correia and Captivo 2003):

$$\lambda^k = \delta^k \frac{\widehat{Z} - L^k(\alpha)}{\sum_{i\in I}\sum_{p\in P}\sum_{t\in T}(\gamma_{ipt}^k)^2},$$

where δ^k is a scalar, $L^k(\alpha)$ equals the value of $L(\alpha)$ at iteration k and $\widehat{Z}$ is the cost of the best feasible solution found so far. The Lagrange multipliers for the $(k+1)-th$ iteration are then updated by:

$$\alpha_{ipt}^{(k+1)} = \alpha_{ipt}^k + \lambda^k\gamma_{ipt}^k \quad \forall i\in I, \;\; \forall p\in P, \;\; \forall t\in T.$$

Bundle Method. The second method used to solve the Langrangian dual is an implementation of the bundle method (Frangioni 2005). The method uses a subset of the tuples $< L(\alpha^s), \gamma^s >$

with $s \in B$ and B is referred to as the bundle of subgradients γ^s. From the primal view point, the following quadratic problem has to be solved at each iteration (Frangioni and Gallo 1999):

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

where R is the so called trust region, and $E_s = L(\alpha) + \gamma(\hat{\alpha} - \alpha) - L(\hat{\alpha})$ is the linearization error from the current point $\hat{\alpha}$. 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 4.2). The tentative ascent direction is then computed by the convex combination of the subgradients, using the convex multipliers θ. Alternatively, the dual problem can be solved to compute the ascent direction, or directly the new point. Frangioni and Gallo (1999) elaborate on this relationship in detail.

Bundle methods usually possess stronger convergence properties than the subgradient method. However, they also tend to require more time to compute the Lagrange multipliers. They are therefore beneficial when a small number of iterations is performed to reach the desired accuracy.

3.3. Upper Bound Generation

At each iteration, a feasible solution is generated based on the Lagrangian solution obtained by solving the Lagrangian subproblem. This solution provides an upper bound for the optimal integer solution of the problem that directly impacts the convergence of the subgradient and bundle methods. Even though high quality upper bounds are desirable, it is important that they are generated in an efficient manner, as the solution of the Lagrangian dual typically involves hundreds of iterations.

The solution of the Lagrangian subproblem provides a facility opening schedule for the entire planning horizon. This schedule is defined by capacity levels ℓ'_{jt} indicating the facility size at locaton j at time period t. In addition to the schedule, the Lagrangian solution provides a demand allocation. As the demand constraints (2) have been relaxed, the customer demands d_{ipt} are either exactly met, under-served or over-served.

The set of all customer demands can therefore be separated into three subsets, where Σ_1, Σ_2 and Σ_3 denote the demands defined by triplets $< i,p,t >$, which are exactly met, over-served and under served, respectively:

$$\Sigma_1 = \left\{ < i,p,t >: \sum_{j \in J} x_{ij(\ell'_{jt})pt} = 1 \right\}, \Sigma_2 = \left\{ < i,p,t >: \sum_{j \in J} x_{ij(\ell'_{jt})pt} > 1 \right\}$$

$$\text{and } \Sigma_3 = \left\{ < i,p,t >: \sum_{j \in J} x_{ij(\ell'_{jt})pt} < 1 \right\}.$$

To obtain an integer feasible solution, we heuristically reduce redundant demand allocation for the pairs in Σ_2 and increase missing demand allocation for the pairs in Σ_3. Note that the heuristic to increase available capacity is very simple. The difficulty here is to find general rules that perform well on the different problem variants that may be modeled by the use of the GMC formulation. The heuristic procedure used to obtain a feasible solution is composed of the following steps:

1. **Reduce demand allocation**: For each $< i,p,t > \in \Sigma_2$, all facility/size pairs $(j,(\ell'_{jt}))$ are sorted in decreasing order of their allocation costs $d_{ipt}g_{ij\ell pt}$. The allocated flow is removed until the total allocated demand for $< i,p,t >$ equals 1.

2. **Increase capacities**: If the total remaining capacity is smaller than the total remaining demand, we increase the capacity sequentially for 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 a facility is already open at any moment in the planning horizon, we increase the capacity for the current time period to its maximum capacity level throughout the planning; if no facility exists, we increase the capacity level until the missing capacity is covered or the maximum capacity level for this facility is reached.

3. **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 $d_{ipt}g_{ij\ell pt}$. Demand is allocated to these pairs until the total allocated demand for $< i,p,t >$ equals 1.

4. **Reduce unused capacities of open facilities**: For each facility, we use a dynamic programming algorithm, similar to the one used to solve Lagrangian subproblem, to compute the

optimal opening schedule (i.e., the one with the lowest costs) that guarantees sufficient capacity to satisfy the demand allocated to that facility.

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. Upper Bound Improvement: Restricted MIP Model

The previous section outlined the heuristic procedure to generate integer feasible solutions. This heuristic focuses on efficiency rather than on the quality of the upper bound. However, the objective of the Lagrangian heuristic is to provide high quality solutions. It is therefore beneficial to add an optimization phase that aims at finding solutions of higher quality than those already found during the solution of the Lagrangian dual. Either one tries to improve promising solutions that have been found during the Lagrangian dual method, or one constructs new solutions based on information gathered during the process.

Local improvement heuristics, based on already available solutions, have been successfully applied in a second optimization phase after performing a Lagrangian relaxation method (e.g., Correia and Captivo 2006, Li et al. 2009). However, they require a detailed knowledge of the problem structure. As seen in Section 2.2, the GMC is a fairly general model, capable of representing different facility location problems. In some cases, certain capacity levels represent open facilities, whereas other capacity levels represent closed facilities. Given the flexibility regarding the usage of capacity levels, it is beneficial to use a more general mechanism to find high quality solutions.

4.1. MIP Model Based on Lagrangian Solutions

The Lagrangian heuristic proposed in this work involves a second optimization phase using information collected during the solution of the Lagrangian dual. We solve a restricted MIP, taking into consideration the decisions made by the Lagrangian solutions. One would expect that the larger the decision space is, the better the quality of the final solution will be. However, this is only true without memory and computing time limitations. Given those limitations, a large MIP may result in a low overall performance, as the model is too large to be solved with the available time and memory resources. We therefore filter the decisions considered in the restricted MIP to sufficiently reduce the size of the model.

Let n^{Iter} denote the number of iterations performed by the subgradient or by the bundle method. Let $n^C_{j\ell t}$ be the number of Lagrangian solutions where capacity level ℓ has been selected for location j at time period t (note that we have $\sum_{\ell \in L} n^C_{j\ell t} = n^{Iter}$ for each j and t). Furthermore, let L^R_{jt} be the set of capacity levels for location j and period t available in the restricted MIP. The restricted MIP is then defined as follows:

- **Decision fixing.** For each j and t, a decision is fixed to capacity level ℓ if it appears in at least $100 \times pFix$ (with $pFix \in]0.5, 1]$) percent of all iterations, i.e, $L^R_{jt} = \{\ell\}$, if $n^C_{j\ell t}/n^{Iter} \geq pFix$.
- **Selection of available capacity levels.** If the capacity level for location j and time period t is not fixed, L^R_{jt} is composed by the n^S capacity levels that appear the most often in the Lagrangian solutions (i.e., have the highest $n^C_{j\ell t}$) and appear in at least one Lagrangian solution (i.e., $n^C_{j\ell t} \geq 1$).
- **Defining the set of capacity transitions.** Decisions $y_{j\ell_1\ell_2 t}$ are defined for all combinations between ℓ_1 and ℓ_2, with $\ell_1 \in L^R_{jt}$ and $\ell_2 \in L^R_{j(t+1)}$, if available in the original GMC formulation.

Using appropriate values for the parameters $pFix$ and n^S, the original GMC model can be reduced to a restricted version with reasonable memory and computing time requirements, taking into consideration only decisions that have been found to be significant by the Lagrangian solutions.

4.2. MIP Model Based on Convexified Bundle Solutions

When using the bundle method to solve the Lagrangian dual, we may take advantage of the information the method holds concerning the set of solutions that are linked to the subgradients in the bundle, as demonstrated by Borghetti et al. (2003).

As explained in Section 3.2, 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 a probability that solution s provides a good opening schedule. We may therefore derive probabilities for each of the opening decisions $\tilde{y}_{j\ell t} = \sum_s \theta^s y^s_{j\ell t}$, where $y^s_{j\ell t}$ is 1 if solution s selects capacity level ℓ for location j at period t.

We may now construct a restricted MIP, as previously shown based on the Lagrangian solutions. Instead of using the number of occurrences $n^C_{j\ell t}$ in Lagrangian solutions, we use the value of $\tilde{y}_{j\ell t} \in [0,1]$, defining its importance according to the multipliers θ^s provided by the bundle method. In this case, a capacity level ℓ is fixed at location j and period t if $\tilde{y}_{j\ell t} \geq pFix$, where $pFix \in]0.5, 1]$. Otherwise, L^R_{jt} is composed by the n^S capacity levels with the highest $\tilde{y}_{j\ell t}$ values, with $\tilde{y}_{j\ell t} \geq 0.001$. Note that the Lagrangian solutions linked to the subgradients that are stored in the bundle are only a subset of those generated in all iterations. The set of decisions considered in the restricted MIP based on the convexified bundle solution is therefore very likely to be much smaller than the restricted MIP based on all Lagrangian solutions.

5. Computational Results

In this section, the performance of different configurations for the Lagrangian heuristics and that of the MIP solver CPLEX will be evaluated and compared by means of computational experiments. First, we discuss how test instances were generated. Then, we elaborate on the integrality gap of the different problems. Finally, computational results are presented to explore the impact of parameter choices for the Lagrangian heuristics and to compare different configurations with each other and with CPLEX.

Test instances have been generated by following a scheme similar to that described in Jena et al. (2013). However, the instances used in this previous work included only one commodity, up to 100 candidate facility locations and up to 1000 customer locations. In this work, we use instances that are significantly larger with respect to the number of candidate facility locations and the number of commodities. 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: $300km$, $380km$ and $450km$. 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 (*regular*), the total demand is similar in each time period. The second scenario (*irregular*) 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 computed based on the demand for the first commodity. To be precise, the demand d_{jpt} for $p \geq 2$ is computed as $d_{jpt} = d_{j1t} \cdot rand(1.0, 0.2) \cdot avgDem_p / avgDem_1$, where $avgDem_1 = 10$, $avgDem_2 = 6$, $avgDem_3 = 9$, $avgDem_4 = 5$, $avgDem_5 = 8$, and $rand(1.0, 0.2)$ is a random variable with normal distribution, mean value of 1.0 and standard deviation of 0.2. Construction and operational costs follow concave cost functions, i.e., they involve economies of scale. Jena et al. (2013) also tested a second cost scenario in which the transportation costs are five times higher. The authors found that these instances are significantly easier to solve. In this work, we only consider the instances that are more difficult to solve, i.e., the ones with their original level of transportation costs. 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 and small enough to not increase the size of the models too much. Note that we assume that the problem instances do not contain initially existing facilities. We refer to Appendix A for a detailed description of the parameters used to generate the instances.

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 (2.67GHz), limited to 24GB of RAM.

5.1. Integrality Gaps of the Test Instances

The integrality gap is defined as the difference between the optimal LP relaxation solution value and the cost of an optimal integer solution, divided by the latter. For many instances, the GMC models are very large and exceed the available memory of 24GB. It was therefore not possible to find all of these optimal values. The integrality gap has been exactly determined only for a subset of the 540 instances. Considering the best lower and upper bounds obtained throughout all computational experiments, optimality has been proved for 302, 388, 384 and 382 instances for the DFLPG, the DMCLFP_CR, the DMCFLP_ER and the DMCFLP_CR_ER, respectively.

As observed in Jena et al. (2013), the integrality gap for the GMC based formulations tend to be very small. This turns out to be useful for two reasons. First, when using Lagrangian relaxation, the provided bounds are more meaningful. Low integrality gaps may help to prove optimality within a certain tolerance. Second, the input data for multi-period facility location problems usually comes from forecasts, and it is very likely that the real data will slightly deviate from the forecast, especially for the last time periods. An optimal solution may therefore not be more relevant in practice than a solution that guarantees optimality within a certain tolerance. Melo et al. (2011) therefore aim at a finding solutions within 1% from the optimal solution. On the instances used in this work, the integrality gap has been found to be smaller than or equal to 1% for a fairly large part of the instances. To be precise, the integrality gap is smaller than or equal to 1% for at least 413, 397, 410 and 397 instances for each the four problems, respectively. The Lagrangian relaxation may therefore prove optimality within a deviation of 1% for a large part of the instances if its lower bounds are close to the LP relaxation bounds and its generated upper bounds (i.e., the feasible solutions generated throughout the Lagrangian relaxation) are close to optimal.

5.2. Comparison of Different Configurations for the Lagrangian Heuristics

We now compare the performance of different configurations for the Lagrangian relaxation based heuristics. Section 3 discussed two different methods to solve the Lagrangian dual, the subgradient method and the bundle method. These methods can be used to generate feasible solutions at each iteration. Furthermore, it has been shown in Section 4 how information from the Lagrangian solutions and the convexified bundle solutions can be collected throughout the solution of the Lagrangian dual, and then be used to generate a restricted MIP to find solutions of even better quality.

Parameter Settings. The subgradient method is used with an initial scalar $\delta^k = 2.0$. This scale factor halves every 25 consecutive iterations without improvement in the lower bound. The algorithm terminates if δ^k falls below 0.005. For the bundle method, an implementation similar to the one described by Frangioni (2005) has been used as a black box. The bundle implementation has four principal internal performance and termination criteria, which are set as follows. Parameters *tStar*, *EpsLin* have been set to 10^4 and 10^{-6}, respectively. The long-term *t-strategy* has been set to *"soft"* with a parameter value of *0.1*. In addition to the stopping criteria mentioned above, a 1% optimality stopping criterion has been used, i.e., the algorithms stop as soon as the best lower and upper bounds found are within 1%. All experiments have been limited to a maximum of 2 hours of computing time.

5.2.1. Combining the Lagrangian Dual Solution Methods with a Restricted MIP

After performing the subgradient method, a restricted MIP can be solved based on the Lagrangian solutions (see Section 4.1). When using the bundle method, the restricted MIP can be generated based on either the Lagrangian solutions or on the convexified bundle solution (see Section 4.2). We now compare the performance of different combinations for the heuristic, i.e., the use of the subgradient method and the bundle method to solve the Lagrangian dual, and the use of the restricted MIP based on Lagrangian solutions and the convexified solutions to further improve the solution quality. The bundle method has shown significantly faster convergence than the subgradient method. We therefore stop the method when a maximum of 500 iterations has been performed. For the subgradient method, due to its slower convergence, we also tested configurations with a maximum of 1000 iterations.

Table 1 summarizes the results for seven different solution strategies: the subgradient method without (*"only"*) and with a restricted MIP based on the Lagrangian Solutions (*"w/ LS R-MIP"*), as well as the bundle method without (*"only"*) and with a restricted MIP, based either on the Lagrangian solutions (*"w/ LS R-MIP"*) or on the convexified bundle solution (*"w/ CS R-MIP"*). As mentioned above, the subgradient method has been tested in two variants, stopping either after a maximum of 500 iterations or after a maximum of 1000 iterations. When using the restricted MIP based on the Lagrangian solutions, we use parameter values that have led to good performance (see Section 5.2.2): $pFix = 70\%$ and $n^S = 3$. For the bundle method with the restricted MIP based on the convexified solutions, we used $pFix = 0.85$ and $n^S = 4$, which led to smaller average and maximum optimality gaps than setting $pFix$ to 0.7, 0.8 or 0.9. Note that for the restricted MIP based on the convexified solutions, we only tested n^S values of 2, 3 and 4.

	Subgradient method				Bundle method		
	500 max iter		1000 max iter		500 max iter		
	only	w/ LS R-MIP	only	w/ LS R-MIP	only	w/ LS R-MIP	w/ CS R-MIP
DFLPG							
Avg Gap %	3.01	0.81	1.67	0.72	1.64	0.76	0.72
Max Gap %	17.31	17.31	13.06	7.28	8.83	7.85	2.88
Avg Time (sec)	374.8	1,140.0	486.5	714.5	258.9	844.6	359.3
# prov. 1% gap	211	386	329	407	331	405	407
DMCFLP_CR							
Avg Gap %	3.85	0.82	2.28	0.88	2.10	0.86	0.81
Max Gap %	22.26	12.96	14.38	9.73	10.39	10.39	3.78
Avg Time (sec)	846.9	1,532.2	1,130.1	1,447.2	617.8	1,370.0	884.6
# prov. 1% gap	160	378	287	385	292	392	397
DMCFLP_ER							
Avg Gap %	3.18	0.78	1.72	0.70	1.64	0.74	0.69
Max Gap %	17.58	14.68	12.45	6.63	10.47	8.56	2.57
Avg Time (sec)	379.5	1,137.0	484.4	712.9	247.7	833.9	348.8
# prov. 1% gap	205	391	317	405	325	410	411
DMCFLP_CR_ER							
Avg Gap %	3.77	0.87	2.11	0.82	1.96	0.83	0.77
Max Gap %	21.00	17.76	15.92	8.37	10.14	9.15	3.44
Avg Time (sec)	840.3	1,703.7	1,091.7	1,493.7	569.9	1,335.4	857.6
# prov. 1% gap	174	373	295	389	310	399	395

Table 1 Comparison of different configurations for the Lagrangian based heuristics for the four problems.

The results take into account all 540 instances and are reported for each of the four problem variants. We indicate the average and maximum gap (when compared to the best lower bounds known for the instances), the average computing time and the number of instances for which a 1% optimality has been proved (*"# 1% gap proved"*).

The results are consistent for the four different problem variants. Solving only the Lagrangian dual, the bundle method clearly stays ahead of the subgradient method. Given its stronger convergence properties, it finishes, on average, in significantly shorter computing times. For the subgradient method, allowing 1000 instead of 500 iterations strongly improves the solution quality. After this first phase, a 1% optimality has been proved for more than half of the instances.

Adding the Lagrangian solution based restricted MIP to the subgradient method significantly improved the optimality gap when up to 1000 iterations are performed. With only 500 iterations, the improvement is less significant. This illustrates the importance of reasonably solving the Lagrangian dual before constructing a restricted MIP, because "high-quality" decisions tend to appear in the later stage of the subgradient method.

For the bundle method, a larger improvement of the maximum optimality gap can be observed. Both versions of the restricted MIP result in very competitive results. The maximum optimality gap is always kept below 4.25%, while the average computing time is very reasonable. Using the restricted MIP to improve the solution quality, the number of instances where a 1% optimality gap could be proved increased to over 75% of all instances. While both approaches show similar maximum optimality gaps, the convexified solution presents better average gaps and is capable of proving a 1% gap for more instances.

The results based on the bundle method are clearly better than those based on the subgradient method, as the subgradient method itself already takes a significant portion of the available computing time. Therefore, there is often not enough time left to solve the restricted MIP. However, a heuristic based on the latter could still be effective. Tuning the maximum number of subgradient iterations and the parameters used to define the restricted MIP will hereby make the crucial difference. Such tuning is exemplified in the next section.

5.2.2. Restricted MIP Parameter Tuning

The restricted MIP, performed after the solution of the Lagrangian dual, has to be sufficiently restricted in a way in that it can be reasonably solved within the remaining time. This is done by appropriately setting the two parameters n^S and $pFix$, indicating the maximum number of decisions considered for each location and time period, and the percentage necessary to fix a decision, respectively.

$pFix$		$n^S = 2$	$n^S = 3$	$n^S = 4$	$n^S = 10$
51%	Avg / Max gap %	1.48 / 8.83	1.48 / 8.83	1.48 / 8.83	1.48 / 8.83
	Avg time	233.3 sec	231.5 sec	231.2 sec	228.4 sec
70%	Avg / Max gap %	0.82 / 5.37	0.80 / 5.37	0.80 / 5.37	0.80 / 5.37
	Avg time	235.6 sec	239.3 sec	243.6 sec	242.3 sec
80%	Avg / Max gap %	0.75 / 3.47	0.74 / 3.36	0.73 / 3.15	0.73 / 3.20
	Avg time	286.6 sec	296.2 sec	301.1 sec	302.3 sec
90%	Avg / Max gap %	0.74 / 7.85	0.72 / 3.35	0.71 / 3.35	0.71 / 3.35
	Avg time	385.7 sec	415.0 sec	422.7 sec	419.6 sec
100%	Avg / Max gap %	0.74 / 7.85	0.72 / 7.85	0.72 / 7.85	0.72 / 7.85
	Avg time	597.7 sec	620.0 sec	623.3 sec	614.1 sec
No	Avg / Max gap %	0.74 / 7.85	0.72 / 7.85	0.72 / 7.85	0.72 / 7.85
Fixing	Avg time	582.2 sec	595.9 sec	613.6 sec	601.6 sec

Table 2 Comparison of results for different parameters for the bundle method with MIP based on Lagrangian solutions, applied to the DFLPG.

Table 2 summarizes the results for different parameter values, using the bundle method with a restricted MIP based on the Lagrangian solutions applied to the DFLPG. The results are given for all combinations between different $pFix$ and n^S values, reporting the average and maximum optimality gap, as well as the average computation time. The average computation times increase due to two factors: more capacity level decisions in the MIP (i.e., higher values of n^S), and less variable fixing (i.e., higher values of $pFix$). For the given time limit of 2 hours, well performing values can be found by balancing these two parameters. Setting n^S to 3, 4 or even 10, and $pFix$ between 80% and 90% results in a maximum optimality gap of around 3.36%, while other parameter values may result in gaps of up to 8.83%. Clearly, if more computing time is available, one may allow higher values for these parameters, which may further improve the solution quality.

Similar experiments were performed for different parameter values for the restricted MIP based on the convexified bundle solutions. Not restricting the MIP at all resulted in significantly better results than for the non-restricted MIP based on the Lagrangian solutions. Furthermore, it was found that the restricted MIP based on the convexified bundle solution is less sensitive to changes in the parameter value $pFix$ than the one based on the Lagrangian solutions. These results suggest that the decisions that are part of solutions selected by the bundle are those which are also present in high quality solutions.

5.3. Comparisons with CPLEX

The performance of one of the Lagrangian based heuristics is now compared to CPLEX. We chose the configuration that provided the lowest average and maximum optimality gaps: the bundle method with restricted MIP based on its convexified solution, with $n^S = 4$ and $pFix = 0.85$. CPLEX has been used with standard parameters. As in the previous experiments, a 1% optimality stopping criterion and a time limit of 2 hours have been applied.

Computational Results. Tables 3, 4, 5 and 6 summarize the results for CPLEX, as well as for the Lagrangian based heuristic outlined above for the four different problems DFLPG, DMCLFP_CR, DMCFLP_ER and DMCFLP_CR_ER, respectively. All results are grouped by the number of capacity levels q and the problem dimension defined by the number of candidate facility locations and the number of customers. Each group given by such a combination includes 18 instances. The tables report the average and maximum gaps of the best feasible integer solutions found by the algorithm when compared to the best lower bound known for the corresponding problem instance, as well as the average computing times. Note that the results shown in the Tables 3 – 6 only take into account the instances where CPLEX found a feasible integer solution within the time limit of 2 hours. The number of instances for which CPLEX did not find any feasible solution is indicated by column *"#ns"*. Furthermore, column *"# prov. 1% gap"* gives the number of instances (out of those for which CPLEX found a feasible solution) where a 1% optimality gap has been proven by the algorithm. For the Lagrangian heuristic, the number in brackets to the right represents the same count, but for all 540 instances.

The observations made for the results of CPLEX and the Lagrangian based heuristics are similar for all four problems. The number of instances where CPLEX did not find feasible solutions is fairly high, at least 25% of the instances for each of the four problems. In most of the cases, this happens due to memory limitations when the number of capacity levels or the number of candidate facility locations is high. Even though the average quality of solutions found by CPLEX is quite good, the solver provides large optimality gaps on many instances. This is mostly the case when a large number of capacity levels ($q = 10$) is available. As the solver constantly improves its bounds, the optimality gaps proven by the algorithm (shown in brackets) are very close to the gaps when compared to the best known lower bound for the instances. CPLEX is capable of proving a 1% optimality gap for at least 342 out of the 540 instances for each of the four problems.

The Lagrangian based heuristic provides stable results for each of the four problems. When compared to the same instances, it provides an average gap lower than that of CPLEX in computing times that are, on average, significantly lower. For the DFLPG and the DMCFLP_ER, the Lagrangian heuristic is, on average, twelve times faster than CPLEX. For the DMCFLP_CR and the DMCFLP_CR_ER, the heuristic is, on average, five times faster. Most importantly, the maximum optimality gap is at most 3.78%. Due to the strength of the GMC formulation, the maximum optimality gap proven by the Lagrangian heuristic is 4.87%. Furthermore, considering the same set of instances, the heuristic proves a 1% gap for almost the same number of instances as CPLEX. When considering all 540 instances (even those for which CPLEX does not find feasible solutions), the Lagrangian heuristic proves a 1% gap for 395 or more of the 540 instances for each of the four problems.

Interestingly, the difficulty of a problem is not always linked to its dimension. Instances where the number of customers is close to the number of candidate facility locations are significantly harder to solve than those where the number of customers is higher. In particular, this can be observed for instances of dimension (50/50). An analysis showed that these instances tend to possess larger integrality gaps, which may be linked to the fact that the more customers are available, the easier it is to make efficient use of a facility (in terms of allocation costs and capacity usage) in an integer solution.

A Note on the Model Size. As the previous results show, general-purpose MIP solvers such as CPLEX may perform very well on small instances, i.e., when the number of capacity levels is low

		CPLEX (with SIs a priori)					Lagrangian Heuristic			
	Instance	Avg	Max	Avg	# prov.	#	Avg	Max	Avg	# prov.
q	size	Gap %	Gap %	Time	1% gap	ns	Gap %	Gap %	Time	1% gap
3	50/50	0.26	0.99	531.7	18	0	0.44	1.32	6.7	7 [7]
	50/200	0.02	0.12	15.1	18	0	0.43	0.92	7.6	18 [18]
	100/100	0.11	0.51	54.1	18	0	0.43	0.95	13.0	17 [17]
	100/400	0.04	0.37	63.9	18	0	0.58	0.95	39.7	18 [18]
	150/150	0.13	0.76	82.4	18	0	0.38	0.87	33.3	18 [18]
	150/600	0.06	0.64	179.7	18	0	0.66	0.96	104.6	18 [18]
	200/200	0.17	0.86	116.3	18	0	0.51	0.98	49.2	18 [18]
	200/800	0.09	0.52	370.1	12	6	0.67	0.90	184.1	12 [18]
	250/250	0.04	0.37	179.7	18	0	0.44	0.92	88.8	18 [18]
	250/1000	0.15	0.86	373.5	6	12	0.52	0.94	262.3	6 [18]
	All	0.10	0.99	177.1	162	18	0.50	1.32	61.5	150 [168]
		[0.18]	[1.00]				[0.67]	[2.32]		
5	50/50	0.71	2.11	3,122.8	13	0	0.88	2.06	28.4	3 [3]
	50/200	0.17	0.79	90.7	18	0	0.48	0.89	17.5	16 [16]
	100/100	0.46	1.30	1,268.3	16	0	0.64	1.26	36.8	9 [9]
	100/400	0.04	0.16	145.8	18	0	0.55	0.87	58.3	18 [18]
	150/150	0.39	1.13	1,106.5	15	1	0.61	1.24	98.5	12 [13]
	150/600	0.08	0.67	255.0	12	6	0.63	0.96	116.2	12 [18]
	200/200	0.22	0.84	762.6	16	2	0.52	0.92	89.3	15 [16]
	200/800	0.06	0.20	552.0	6	12	0.53	0.89	243.8	6 [18]
	250/250	0.15	0.52	885.7	17	1	0.46	0.95	151.2	17 [17]
	250/1000	0.13	0.75	683.3	6	12	0.49	0.94	348.5	6 [18]
	All	0.27	2.11	957.8	137	34	0.59	2.06	90.2	114 [146]
		[0.39]	[2.11]				[0.86]	[4.59]		
10	50/50	23.11	92.72	6,472.0	2	0	1.90	2.88	282.7	0 [0]
	50/200	0.86	2.19	2,823.1	12	3	0.73	1.28	108.0	8 [9]
	100/100	3.03	14.82	5,312.7	4	7	1.26	2.44	131.1	2 [2]
	100/400	0.30	1.44	991.3	10	7	0.55	0.91	123.4	11 [18]
	150/150	2.59	11.93	5,014.6	3	11	0.85	1.31	105.0	2 [2]
	150/600	0.07	0.17	541.2	6	12	0.43	0.67	125.2	6 [17]
	200/200	0.88	1.66	3,400.7	4	12	0.80	1.62	193.3	3 [4]
	200/800	0.12	0.12	1,743.0	1	17	0.15	0.15	681.0	1 [18]
	250/250	0.20	0.36	1,052.7	3	15	0.27	0.40	171.3	3 [5]
	250/1000	-	-	-	0	18	-	-	-	- [18]
	All	6.28	92.72	3,741.6	45	102	1.02	2.88	171.1	36 [93]
		[6.36]	[92.72]				[1.18]	[3.63]		
All	All	1.42	92.72	1,192.7	344	154	0.64	2.88	94.5	300 [407]
		[1.51]	[92.72]				[0.90]	[4.59]		

Table 3 **Comparison of CPLEX and Lagrangian based heuristics for the DFLPG: average and maximum optimality gap when compared to the best known lower bound.**

($q \in \{3, 5\}$) and the number of candidate facility locations is small ($|J| \leq 100$). Clearly, adding the SIs (8) a priori to the model significantly increases the number of constraints and, therefore, the memory requirements of the model. As noted by Jena et al. (2013), the addition of the SIs to the

		CPLEX (with SIs a priori)					Lagrangian Heuristic			
	Instance	Avg	Max	Avg	# prov.	#	Avg	Max	Avg	# prov.
q	size	Gap %	Gap %	Time	1% gap	ns	Gap %	Gap %	Time	1% gap
3	50/50	0.29	1.14	650.6	17	0	0.51	1.54	13.9	7 [7]
	50/200	0.08	0.67	18.5	18	0	0.61	0.99	15.6	18 [18]
	100/100	0.11	0.55	41.2	18	0	0.44	0.84	27.3	17 [17]
	100/400	0.08	0.66	90.6	18	0	0.62	0.94	86.8	18 [18]
	150/150	0.04	0.23	102.4	18	0	0.52	0.99	62.4	18 [18]
	150/600	0.10	0.85	275.5	18	0	0.72	0.98	243.4	18 [18]
	200/200	0.13	0.92	198.0	18	0	0.56	0.95	137.8	16 [16]
	200/800	0.19	0.93	589.3	12	6	0.57	0.95	530.9	12 [18]
	250/250	0.06	0.34	496.3	18	0	0.56	0.95	219.1	18 [18]
	250/1000	0.15	0.73	791.0	6	12	0.67	0.96	596.7	6 [18]
	All	0.12	1.14	281.1	161	18	0.57	1.54	151.0	148 [166]
		[0.19]	[1.27]				[0.74]	[2.75]		
5	50/50	0.67	2.44	1,973.7	14	0	0.91	2.24	32.2	3 [3]
	50/200	0.26	0.69	86.4	18	0	0.54	1.04	46.9	16 [16]
	100/100	0.37	0.93	1,144.2	17	0	0.45	1.05	91.8	13 [13]
	100/400	0.11	0.89	203.6	18	0	0.61	0.97	136.9	18 [18]
	150/150	0.39	1.00	1,104.5	17	0	0.54	1.19	191.7	15 [15]
	150/600	0.09	0.85	413.8	12	6	0.69	0.95	280.6	12 [18]
	200/200	0.23	0.88	992.8	18	0	0.52	0.96	322.6	18 [18]
	200/800	0.16	0.60	868.0	6	12	0.36	0.94	597.0	6 [18]
	250/250	0.48	3.88	1,473.8	16	1	0.54	1.12	457.3	16 [17]
	250/1000	0.16	0.81	1,214.5	6	12	0.64	0.98	734.3	6 [18]
	All	0.32	3.88	950.4	142	31	0.59	2.24	227.7	123 [154]
		[0.47]	[3.95]				[0.89]	[3.70]		
10	50/50	4.27	19.79	5,779.7	4.00	1	2.32	3.78	977.8	0 [0]
	50/200	0.74	1.51	3,308.0	9	5	0.82	1.53	202.8	4 [4]
	100/100	7.84	66.50	5,449.2	4	5	1.67	3.19	376.4	0 [0]
	100/400	0.92	8.29	1,151.1	11	6	0.66	0.99	370.7	11 [15]
	150/150	17.03	88.13	5,172.8	4	7	1.26	2.40	331.3	1 [1]
	150/600	0.26	0.89	1,010.3	6	12	0.55	0.88	440.5	6 [15]
	200/200	23.45	89.67	4,217.1	4	10	1.09	1.73	506.6	3 [3]
	200/800	0.22	0.63	3,467.8	6	12	0.44	0.95	1,395.0	6 [17]
	250/250	1.02	2.91	3,064.0	3	14	0.52	0.80	664.8	4 [6]
	250/1000	-	-	-	0	18	-	-	-	- [16]
	All	6.41	89.67	3,951.9	51	90	1.23	3.78	555.2	35 [77]
		[7.08]	[100.00]				[1.53]	[4.50]		
All	All	1.61	89.67	1,353.7	354	139	0.72	3.78	270.2	306 [397]
		[1.84]	[100.00]				[1.05]	[4.50]		

Table 4 Comparison of CPLEX and Lagrangian based heuristics for the DMCFLP_CR: average and maximum optimality gap when compared to the best known lower bound.

GMC based models significantly facilitates the solution of the problems. In fact, for the instances used in this work, without the use of the SIs, CPLEX provides very low solution quality even for small instances. Other studies, such as the one by Gendron and Larose (2014) applied to a network

		CPLEX (with SIs a priori)					Lagrangian Heuristic			
q	Instance size	Avg Gap %	Max Gap %	Avg Time	# prov. 1% gap	# ns	Avg Gap %	Max Gap %	Avg Time	# prov. 1% gap
3	50/50	0.25	0.94	779.8	17	0	0.25	0.86	6.7	11 [11]
	50/200	0.01	0.10	20.5	18	0	0.59	0.97	6.9	18 [18]
	100/100	0.17	0.67	40.7	18	0	0.37	0.94	13.8	17 [17]
	100/400	0.04	0.37	89.0	18	0	0.65	0.99	40.7	18 [18]
	150/150	0.20	0.74	120.3	18	0	0.46	0.96	26.4	18 [18]
	150/600	0.01	0.17	263.9	18	0	0.67	0.95	105.0	18 [18]
	200/200	0.06	0.43	189.7	18	0	0.31	0.86	46.3	18 [18]
	200/800	0.11	0.76	466.3	12	6	0.75	0.96	190.3	12 [18]
	250/250	0.04	0.40	393.0	18	0	0.50	0.91	97.1	18 [18]
	250/1000	0.31	0.91	538.8	6	12	0.50	0.91	247.3	6 [18]
	All	0.11	0.94	265.3	161	18	0.50	0.99	61.4	154 [172]
		[0.20]	[1.39]				[0.68]	[2.11]		
5	50/50	0.56	1.60	2,521.3	15	0	0.85	1.79	34.7	3 [3]
	50/200	0.19	0.65	100.7	18	0	0.48	0.86	18.3	15 [15]
	100/100	0.44	1.83	1,376.8	15	0	0.58	1.35	35.1	10 [10]
	100/400	0.10	0.45	240.1	18	0	0.57	0.83	58.2	18 [18]
	150/150	0.43	1.44	1,286.9	16	0	0.50	1.21	91.9	13 [13]
	150/600	0.01	0.11	393.4	12	6	0.67	0.97	118.4	12 [18]
	200/200	0.38	2.58	1,061.8	16	1	0.53	0.98	113.6	16 [16]
	200/800	0.05	0.11	796.5	6	12	0.51	0.94	239.0	6 [18]
	250/250	0.17	0.63	1,211.4	16	2	0.51	0.91	174.9	16 [17]
	250/1000	0.32	0.82	1,076.8	6	12	0.54	0.94	328.2	6 [18]
	All	0.29	2.58	1,039.8	138	33	0.58	1.79	94.2	115 [146]
		[0.43]	[2.60]				[0.87]	[3.84]		
10	50/50	6.31	87.99	5,595.0	6.00	0	1.44	2.57	149.9	1 [1]
	50/200	0.83	5.76	2,029.6	15	1	0.64	1.19	125.5	9 [10]
	100/100	8.18	91.36	4,867.1	6	3	1.22	2.43	157.3	2 [2]
	100/400	0.23	0.63	1,012.0	11	7	0.51	0.86	125.1	11 [17]
	150/150	11.78	95.92	5,188.2	4	9	1.06	1.92	570.4	2 [2]
	150/600	0.09	0.31	1,139.2	6	12	0.44	0.67	154.3	6 [16]
	200/200	0.61	1.91	2,713.2	4	13	0.80	1.48	166.0	2 [3]
	200/800	3.20	6.33	3,544.0	1	16	0.02	0.04	507.0	2 [18]
	250/250	0.10	0.19	1,570.0	3	15	0.40	0.84	165.3	3 [6]
	250/1000	-	-	-	0	18	-	-	-	- [18]
	All	4.30	95.92	3,468.0	56	94	0.91	2.57	197.3	38 [93]
		[4.62]	[100.00]				[1.18]	[3.83]		
All	All	1.09	95.92	1,250.8	355	145	0.62	2.57	103.2	307 [411]
		[1.25]	[100.00]				[0.91]	[3.84]		

Table 5 Comparison of CPLEX and Lagrangian based heuristics for the DMCFLP_ER: average and maximum optimality gap when compared to the best known lower bound.

design problem, confirm that it may be beneficial to add these inequalities in a branch-and-cut scheme. However, this only yields good performance if only a small number of SIs are violated and therefore added to the model. In the case of the DFLPG, a significant number of SIs are violated

		CPLEX (with SIs a priori)					Lagrangian Heuristic			
	Instance	Avg	Max	Avg	# prov.	#	Avg	Max	Avg	# prov.
q	size	Gap %	Gap %	Time	1% gap	ns	Gap %	Gap %	Time	1% gap
3	50/50	0.23	1.20	738.9	17	0	0.48	1.40	13.9	7 [7]
	50/200	0.05	0.44	20.6	18	0	0.46	0.94	16.0	18 [18]
	100/100	0.15	0.73	70.4	18	0	0.38	0.93	32.5	17 [17]
	100/400	0.13	0.87	99.0	18	0	0.53	0.90	84.5	18 [18]
	150/150	0.09	0.39	127.3	18	0	0.49	0.94	58.3	18 [18]
	150/600	0.03	0.28	302.9	18	0	0.74	0.97	217.0	18 [18]
	200/200	0.13	0.79	248.5	18	0	0.52	0.96	126.4	18 [18]
	200/800	0.15	0.93	666.7	12	6	0.64	0.98	422.9	12 [18]
	250/250	0.11	0.83	453.1	18	0	0.41	0.87	201.0	18 [18]
	250/1000	0.04	0.12	812.3	6	12	0.69	0.95	428.2	6 [18]
	All	0.12	1.20	308.4	161	18	0.52	1.40	130.5	150 [168]
		[0.19]	[1.23]				[0.68]	[2.39]		
5	50/50	0.71	2.96	2,951.6	13	0	0.91	2.42	48.4	3 [3]
	50/200	0.23	0.81	187.3	18	0	0.45	0.82	41.4	14 [14]
	100/100	0.47	1.85	1,286.9	16	0	0.54	1.42	88.8	10 [10]
	100/400	0.12	0.63	273.7	18	0	0.55	0.95	129.6	18 [18]
	150/150	0.41	1.13	1,242.1	16	1	0.52	0.87	164.8	15 [15]
	150/600	0.10	0.89	475.3	12	6	0.67	0.96	242.3	12 [18]
	200/200	0.26	0.84	1,193.9	18	0	0.40	0.86	319.4	17 [17]
	200/800	0.17	0.89	1,202.3	6	12	0.68	0.97	599.5	6 [18]
	250/250	0.47	4.61	1,230.1	15	2	0.52	0.98	358.8	16 [17]
	250/1000	0.03	0.15	1,347.3	6	12	0.66	0.86	640.3	6 [18]
	All	0.33	4.61	1,142.0	138	33	0.57	2.42	205.3	117 [148]
		[0.47]	[4.61]				[0.87]	[4.00]		
10	50/50	8.06	87.78	6,168.2	3.00	0	2.14	3.44	1,632.8	0 [0]
	50/200	0.63	1.55	2,561.8	11	5	0.70	1.23	172.5	6 [7]
	100/100	15.65	94.16	5,781.8	3	4	1.72	2.59	776.1	0 [0]
	100/400	0.28	0.68	705.1	10	8	0.66	0.96	194.3	10 [14]
	150/150	0.98	1.82	4,655.6	4	11	1.02	1.66	289.4	1 [1]
	150/600	0.04	0.19	1,394.0	6	12	0.53	0.99	492.5	6 [15]
	200/200	19.75	96.06	4,567.4	3	13	0.97	1.69	384.0	2 [2]
	200/800	-	-	-	0	18	-	-	-	- [17]
	250/250	0.23	0.50	3,733.7	3	15	0.40	0.57	453.3	3 [6]
	250/1000	-	-	-	0	18	-	-	-	- [17]
	All	6.34	96.06	4,043.6	43	104	1.24	3.44	693.4	28 [79]
		[6.74]	[100.00]				[1.44]	[4.87]		
All	All	1.43	96.06	1,364.1	342	155	0.68	3.44	270.2	295 [395]
		[1.59]	[100.00]				[1.00]	[4.87]		

Table 6 Comparison of CPLEX and Lagrangian based heuristics for the DMCFLP_CR_ER: average and maximum optimality gap when compared to the best known lower bound.

in its LP relaxation. Adding the inequalities as CPLEX user cuts to reduce the size of the model showed less competitive results. For more than 40% of the instances, the solver could not find feasible solutions. When feasible solutions were found, the average optimality gap was consistently high, on average, more than 10%.

We also note that, even though we use information from the Lagrangian solutions, other mechanism could be used to rate the importance of opening decisions to generate a MIP that is significantly restricted in its size. Theoretically, using the LP relaxation solution would be one alternative. However, as the LP relaxation cannot be efficiently solved (or not at all) for large instances, such a solution strategy would be applicable only to small and medium sized instances, or in computing environments with significantly larger memory and time resources.

We finally would like to remark that, even though the computational results are reported using CPLEX v12.6, all experiments had previously been performed with v12.4. We observed a significant improvement of the computing times: solving the models with CPLEX was, on average, about 20% to 50% faster for each of the four problem variants. The improvement for the Lagrangian heuristics has been found to be, on average, between 35% and 80%. This may be due to the fact that the solver improved particularly for small problems, as it is the case for the restricted MIP used in the Lagrangian heuristics.

6. Conclusions and Future Research

In this work, we have extended the Dynamic Facility Location Problem with Generalized Modular Capacities by considering demands for multiple commodities. We addressed the solution of large-scale instances and proposed a heuristic based on two optimization phases. First, the Lagrangian dual is solved, involving the iterated solution of the Lagrangian subproblem. In this phase, feasible solutions of reasonable quality are found in very short computing times. Then, a restricted MIP is generated taking into consideration only decisions that have been found important during the solution of the Lagrangian dual. Using this approach, the final solution quality is consistently within 3.78% from the best known lower bound, even for instances for which CPLEX does not find feasible solutions due to the large memory and solution time required by the model.

The general cost structure of this problem allows for representing several existing facility location problems. In addition to the DFLPG, in which the capacity change costs are based on a cost matrix, this has been exemplified on three special cases. Given the strength of the GMC formulation, the Lagrangian heuristic was able to prove optimality within 1% for most of the small and medium sized instances. The proposed model and solution method may be applied to other problems, especially to those where the model size passes the limits of state-of-the-art MIP solvers. It may also be applied to larger instances than those addressed in this work, as the method consumes very little memory.

The Lagrangian dual has been solved by the classical subgradient method and a bundle implementation. Although the bundle method requires more time to compute the Lagrangian multipliers, it consistently outperformed the subgradient approach due to its strong convergence properties. On average, it required half of the time and resulted in a higher solution quality.

While local improvement heuristics such as tabu search have been common as a second phase optimization, the use of a restricted MIP is an interesting alternative, as general-purpose MIP solvers constantly improve. The implementation of a restricted MIP is very simple. Furthermore, one can handle any kind of problem structure that can be defined as a MIP. Even though one does not have to worry about finding the right trade-off between size and inspection time of a neighborhood, the question of how to significantly restrict the size of the original MIP is crucial. The bundle method with restricted MIP resulted in very competitive results, especially since the use of the convexified solutions already limits the decisions to those stored in the bundle. For the subgradient method, a well performing filtering approach based on the Lagrangian solutions may be designed, for example, by better tuning the maximum number of subgradient iterations and the parameter values for the restricted MIP.

Acknowledgments

The authors are grateful to MITACS, the Natural Sciences and Engineering Research Council of Canada (NSERC) and the Fonds de recherche du Québec Nature et Technologies (FRQNT) for their financial support. The authors would also like to thank Antonio Frangioni and Enrico Gorgone for providing the implementation of the bundle method, as well as their valuable advice on its usage.

References

Agar, M. C., S. Salhi. 1998. Lagrangean heuristics applied to a variety of large capacitated plant location problems. *Journal of the Operational Research Society* **49**(10) 1072–1084.

Arostegui, M. A., S. N. Kadipasaoglu, B. M. Khumawala. 2006. An empirical comparison of Tabu Search, Simulated Annealing, and Genetic Algorithms for facilities location problems. *International Journal of Production Economics* **103**(2) 742–754.

Barcelo, J., Å. Hallefjord, E. Fernandez, K. Jörnsten. 1990. Lagrangean relaxation and constraint generation procedures for capacitated plant location problems with single sourcing. *OR Spektrum* **12**(2) 79–88.

Beasley, J. E. 1993. Lagrangean heuristics for location problems. *European Journal of Operational Research* **65**(3) 383–399.

Borghetti, A., Antonio Frangioni, F. Lacalandra, C.a. Nucci. 2003. Lagrangian heuristics based on disaggregated bundle methods for hydrothermal unit commitment. *IEEE Transactions on Power Systems* **18**(1) 313–323.

Brotcorne, L., G. Laporte, F. Semet. 2003. Ambulance location and relocation models. *European Journal of Operational Research* **147**(3) 451–463.

Canel, C., B. M. Khumawala, J. Law, A. Loh. 2001. An Algorithm for the Capacitated, Multi-Commodity Multi-Period Facility Location Problem. *Computers & Operations Research* **28**(5) 411–427.

Chardaire, P., M.-C. Sutter, A. Costa. 1996. Solving the dynamic facility location problem. *Networks* **28**(2) 117–124.

Contreras, I., J.-F. Cordeau, G. Laporte. 2011. The Dynamic Uncapacitated Hub Location Problem. *Transportation Science* **45**(1) 18–32.

Correia, I., E. M Captivo. 2006. Bounds for the single source modular capacitated plant location problem. *Computers & Operations Research* **33**(10) 2991–3003.

Correia, I., M. E. Captivo. 2003. A Lagrangean heuristic for a modular capacitated location problem. *Annals of Operations Research* **122** 141–161.

Diabat, A., J.-P. Richard, C. W. Codrington. 2011. A Lagrangian relaxation approach to simultaneous strategic and tactical planning in supply chain design. *Annals of Operations Research* **203**(1) 55–80.

Dias, J., M. E. Captivo, J. Clímaco. 2006. Capacitated dynamic location problems with opening, closure and reopening of facilities. *IMA Journal of Management Mathematics* **17**(4) 317–348.

Dias, J., M. E. Captivo, J. Clímaco. 2007. Dynamic Location Problems with Discrete Expansion and Reduction Sizes of Available Capacities. *Investigação Operacional* **27**(2) 107–130.

Elhedhli, S., H. Wu. 2010. A Lagrangean Heuristic for Hub-and-Spoke System Design with Capacity Selection and Congestion. *INFORMS Journal on Computing* **22**(2) 282–296.

Fleischmann, B., S. Ferber, P. Henrich. 2006. Strategic Planning of BMW's Global Production Network. *Interfaces* **36**(3) 194–208.

Frangioni, A. 2005. About Lagrangian Methods in Integer Optimization. *Annals of Operations Research* **139**(1) 163–193.

Frangioni, A., G. Gallo. 1999. A bundle type dual-ascent approach to linear multicommodity min-cost flow problems. *INFORMS Journal on Computing* **11**(4) 370–393.

Gendron, B., T. G. Crainic. 1994. Relaxations for multicommodity capacitated network design problems. Tech. rep., CRT-945, Centre de recherche sur les transports, Université de Montréal.

Gendron, B., M. Larose. 2014. Branch-and-price-and-cut for large-scale multicommodity capacitated fixed-charge network design. *EURO Journal on Computational Optimization* 1–21.

Geoffrion, Arthur M. 1974. Lagrangean Relaxation for Integer Programming. *Mathematical Programming Study 2* 82–114.

Görtz, S., A. Klose. 2012. A Simple but Usually Fast Branch-and-Bound Algorithm for the Capacitated Facility Location Problem. *INFORMS Journal on Computing* **24**(4) 597–610.

Held, M., P. Wolfe, H. P. Crowder. 1974. Validation of Subgradient Optimization. *Mathematical Programming* **6** 62–88.

Holmberg, K., J. Ling. 1997. A Lagrangean heuristic for the facility location problem with staircase costs. *European Journal of Operational Research* **97**(1) 63–74.

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

Jacobsen, S. K. 1990. Multiperiod Capacitated Location Models. *Discrete Location Theory*, chap. 4. Wiley-Interscience, 173–208.

Jena, S. D., J.-F. Cordeau, B. Gendron. 2012. Modeling and solving a logging camp location problem. *Annals of Operations Research* doi:10.1007/s10479-012-1278-z.

Jena, S. D., J.-F. Cordeau, B. Gendron. 2013. Dynamic Facility Location with Generalized Modular Capacities. Tech. rep., CIRRELT-2013-18, Centre interuniversitaire de recherche sur les réseaux d'entreprise, la logistique et le transport.

Kim, D.-G., Y.-D. Kim. 2013. A Lagrangian heuristic algorithm for a public healthcare facility location problem. *Annals of Operations Research* **206**(1) 221–240.

Lee, D.-H., M. Dong. 2008. A heuristic approach to logistics network design for end-of-lease computer products recovery. *Transportation Research Part E: Logistics and Transportation Review* **44**(3) 455–474.

Li, J., F. Chu, C. Prins. 2009. Lower and upper bounds for a capacitated plant location problem with multicommodity flow. *Computers & Operations Research* **36**(11) 3019–3030.

Luss, H. 1982. Operations Research and Capacity Expansion Problems: A Survey. *Operations Research* **30**(5) 907–947.

Melo, M. T., S. Nickel, F. Saldanha-da Gama. 2006. Dynamic multi-commodity capacitated facility location: a mathematical modeling framework for strategic supply chain planning. *Computers & Operations Research* **33**(1) 181–208.

Melo, M. T., S. Nickel, F. Saldanha-da Gama. 2009. Facility location and supply chain management A review. *European Journal of Operational Research* **196**(2) 401–412.

Melo, M. T., S. Nickel, F. Saldanha-da Gama. 2011. A tabu search heuristic for redesigning a multi-echelon supply chain network over a planning horizon. *International Journal of Production Economics* **136**(1) 218–230.

Peeters, D., A. P. Antunes. 2001. On solving complex multi-period location models using simulated annealing. *European Journal of Operational Research* **130**(1) 190–201.

Revelle, C., H. Eiselt, M. S. Daskin. 2008. A bibliography for some fundamental problem categories in discrete location science. *European Journal of Operational Research* **184**(3) 817–848.

Shulman, A. 1991. An Algorithm for Solving Dynamic Capacitated Plant Location Problems with Discrete Expansion Sizes. *Operations Research* **39**(3) 423–436.

Smith, H. K., G. Laporte, P. R. Harper. 2009. Locational analysis: highlights of growth to maturity. *Journal of the Operational Research Society* **60** S140–S148.

Sridharan, R. 1991. A Lagrangian Heuristic for the Capacitated Plant Location Problem with Side Constraints. *Journal of the Operational Research Society* **42**(7) 579–585.

Sridharan, R. 1995. The capacitated plant location problem. *European Journal of Operational Research* **87**(2) 203–213.

Thomas, D. J., P. M. Griffin. 1996. Coordinated supply chain management. *European Journal of Operational Research* **94**(1) 1–15.

Troncoso, J., R. Garrido. 2005. Forestry production and logistics planning: an analysis using mixed-integer programming. *Forest Policy and Economics* **7**(4) 625–633.

Van Roy, Tony J., D. Erlenkotter. 1982. A Dual-Based Procedure for Dynamic Facility Location. *Management Science* **28**(10) 1091–1105.

Wu, L., X. Zhang, J. Zhang. 2006. Capacitated facility location problem with general setup cost. *Computers & Operations Research* **33**(5) 1226–1241.

Appendix

A. Test Instances

Instances for multi-period facility location problems essentially contain information about the customer demand for each time period, construction costs of the facilities and the costs to allocate demand between customers and facilities. The DFLPG and the three special cases additionally involve a detailed cost structure for the capacity changes. Due to the lack of openly available instance sets that include these properties, we generated a total of 540 instances, 180 for each capacity level, to test the presented models. These essentially extend the instances used by Jena et al. (2013) by adding multiple commodities, the use of a cost matrix for capacity changes and a larger set of candidate facility locations. In the following we present how these instance properties are generated and which parameters are used.

A.1. Problem dimension

Instances were generated with different numbers of candidate facility locations $|J|$ and customers $|I|$, combining all pairs of $J \in \{50, 100, 150, 200, 250\}$ and $I \in \{|J|, 4 \cdot |J|\}$. To be precise, the instance dimensions are: *(10/20)*, *(50/50)*, *(50/200)*, *(100/100)*, *(100/400)*, *(150/150)*, *(150/600)*, *(200/200)*, *(200/800)*, *(250/250)* and *(250/1000)*.

A.2. Number of capacity levels

The number of capacity levels q also impacts on the size of the models. Instances are generated with a maximum of 3, 5 and 10 capacity levels, which are assumed to be reasonable values for a broad variety of different application contexts.

The capacities $u_{j\ell}$ are generated based on the total number of customers and are chosen such that a considerably large number of facilities (about half of the candidate locations) is selected. The larger the set of customers, the higher is the capacity of each level. To be precise, we set $u_{j1} = 300$ if the instance covers 50 customers, $u_{j1} = 600$ if the instance covers 100 customers, $u_{j1} = 800$ if the instance covers 150 customers, $u_{j1} = 1000$ if the instance covers 200 customers, $u_{j1} = 1200$ if the instance covers 250 customers, $u_{j1} = 2000$ if the instance covers 400 customers, $u_{j1} = 2500$ if the instance covers 600 customers, $u_{j1} = 3000$ if the instance covers 800 customers and $u_{j1} = 5000$ if the instance covers 1000 customers. The capacities of higher capacity levels $\ell \geq 2$ are set as multiples of the first capacity level, i.e., $u_{j\ell} = \ell \cdot u_{j1}$. Note that we assume that the problem instances do not contain initially existing facilities, i.e., the initial capacity level of each facility is 0.

A.2.1. Number of time periods

All generated instances contain ten time periods, which is found to be sufficient to demonstrate capacity changes along time and small enough to not increase the size of the models too much.

A.3. Customer/facility locations

For each of the different problem sizes, $|I|$ customer demand points have been randomly generated following a continuous uniform distribution, rounding the x and y coordinates to the next lowest integer value. The first $|J|$ points of $|I|$ customer locations have additionally been defined as candidate facility locations and therefore coincide with the customer demand points. The networks were generated on squares of the following three sizes: $300km$, $380km$ and $450km$.

A.4. Demand allocation costs

Costs are divided into fixed and variable costs. Fixed costs are given by the construction of facilities and the change of their capacity levels. Variable costs are composed of the costs to produce and transport the commodities.

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 transportation costs are composed of two components:

1. A cost that depends on the total distance, referred to as the vehicle cost. The vehicle cost is linear in function of the Euclidean distance between the two points on the network (5$/km).

2. A cost that depends on the travel time, referred to as the driver's payment. The driver's payment is 0 if the two points are within one-hour of transportation distance (assuming an average vehicle speed of 62km/h) and linear in function of the Euclidean distance if the two points are at more than one hour of driving distance (50$/h).

Let $dist_{ij}$ denote the distance between facility location j and customer i. The costs to transport one unit of demand from facility j to customer i is therefore set to:

$$g_{ij}^t = 5 \cdot dist_{ij} + 50 \cdot \max\left(0, \frac{dist_{ij}}{62} - 1\right)$$

The variable and fixed costs include economies of scale in function of the size of the facility. These costs are therefore described by concave cost functions, as explained in the following. The production costs for each unit served from a facility to a customer is defined as the cost to operate a facility and depends on the size of the facility. The cost to produce one commodity unit at capacity level 1 is set to 20.90$. At each higher capacity level, the production cost is 3% cheaper than at the previous level:

$$g^p_{j0} = 20.90$$

$$g^p_{j\ell} = 0.97 \cdot g^p_{j(\ell-1)}$$

Note that the production costs are added to the transportation costs to determine the total demand allocation costs $g_{ij\ell t}$ to serve the customer demands:

$$g_{ij\ell t} = g^t_{ij} + g^p_{j\ell}$$

In addition to the demand allocation costs as discussed above, a second set of instances was generated with five times higher transportation costs.

A.5. Fixed costs

The construction cost, also referred to as capacity expansion cost, is set to 100,000$ for a facility of level 1. Each additional capacity level is 10% cheaper than the previous one. The construction costs for facilities of different capacity levels are therefore computed according to the following formula:

$$f^o_{j0} = 100,000$$

$$f^o_{j1} = 190,000$$

$$f^o_{j\ell} = f^o_{j(\ell-1)} + 0.9 \cdot (f^o_{j(\ell-1)} - f^o_{j(\ell-2)})$$

The maintenance costs for a facility of a certain size are computed in a similar fashion. They are set relatively high to motivate capacity changes. The maintenance costs for a facility of capacity level 1 are set to 51,000$. The maintenance costs for each additional capacity level are 15% cheaper than the previous ones:

$$F^o_{j0} = 51,000$$

$$F^o_{j1} = 94,350$$

$$F^o_{j\ell} = F^o_{j(\ell-1)} + 0.85 \cdot (F^o_{j(\ell-1)} - F^o_{j(\ell-2)})$$

Fixed Costs for the Special Cases

For the three special cases, i.e., the DMCFLP_CR, DMCFLP_ER and the DMCFLP_CR_ER, the cost to reduce the capacity of a facility by ℓ capacity levels is set to 10% of the costs to expand the capacity of a facility by ℓ capacity levels.

Finally, the costs for reopening and closing existing facilities were taken from the input data of the previously mentioned industrial application introduced by Jena et al. (2012). Although being strictly increasing, these costs do not necessarily represent economies of scale. The costs to reopen a closed facility of capacity level $1,\ldots,10$ are 3,138.34, 4,084.69, 4,924.58, 5,693.26, 7,085.07, 7,727.50, 8,342.34, 8,933.68, 10,057.70 and 10,594.80, respectively. The costs to close an open facility of capacity level $1,\ldots,10$ are 8,624.93, 11,595.80, 14,305.60, 16,836.50, 21,524.10, 23,727.90, 25,858.30, 27,925.70, 31,901.10 and 33,820.70, respectively.

Fixed Costs for the DFLPG

For the DFLPG, the construction costs are as indicated above, i.e., the costs to construct a facility of size ℓ and its maintenance costs at time period t are set to: $f_{j0\ell t} = f^o_{j\ell} + F^o_{j\ell}$.

The costs to change capacity levels for this problem are based on a cost matrix, and, therefore, differ from the costs for capacity expansion and reduction shown above for the special cases. The cost to completely remove a facility are set to 25% of the construction costs of a facility of the same size: $f_{j\ell 0t} = f^o_{j\ell}/4$.

Finally, the cost to change the capacity level from $\ell_1 \geq 1$ to $\ell_2 \geq 1$ are set to the difference of their construction costs, scaled by 50%:

$$f_{j\ell_1\ell_2 t} = \begin{cases} 1.5 \cdot (f^o_{j\ell_2} - f^o_{j\ell_1}) \text{ , if } \ell_1 < \ell_2 \\ 1.5 \cdot (f^o_{j\ell_1} - f^o_{j\ell_2}) \text{ , if } \ell_1 > \ell_2. \end{cases}$$

A.6. Demand distribution

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 (*regular*), the total demand is similar in each time period. We set the average demand for a customer to 12 units per time period. The total demand for all customers is therefore approximately $10 \cdot |I|$ units at each time period. The second scenario (*irregular*) assumes that the total demand follows strong variations along time and therefore varies at each time period. In this scenario, the total demand for all customers is multiplied by a random distortion factor at each time period. This random distortion factor is set to the absolute value of a normal random variable with mean value 1.0 and standard deviation 0.6 (note that this procedure produced distortion factors from 0.14 to 2.24). Let $totDem_t$ be the total customer demand for time period t, computed as explained above for one of the two scenarios.

We now explain how the individual demands for each of the customers are generated and distributed on the different time periods such that its total sum equals approximately the value of $totDem_t$ at each of the time periods. For all customers and all time periods, the total demand covers approximately $12 \cdot |I| \cdot |T|$ units. In a first step, this total demand is randomly distributed on each of the customers. In a second step, each customer demand is distributed on different time periods:

1. Let $totRemDem$ denote the total demand for all customers and time periods that has not yet been allocated to any customer. Furthermore, let $numRemCust$ indicate the number of customers that have not yet been allocated any demand. For each customer, its total demand for all time periods, denoted to $totJDem_j$, is computed as a random normal variable with a mean $\mu = totRemDemand/numRemCust$ and standard deviation $\sigma = \mu/2$. Note that, throughout our instance generation, this method did not produce any negative value.

2. The total demand for each customer, $totJDem_j$ is then divided into four equal parts. One part of the demand is allocated to a time period that is randomly selected following a uniform distribution. Each of the other three parts is allocated to the time period t that has the highest gap between the total demand yet allocated to period t and its value $totDem_t$.

The demands for the second to fifth commodity are computed based on the demand of the first commodity. To be precise, the demand d_{ipt} for $p \geq 2$ is computed as $d_{ipt} = d_{i1t} \cdot rand(1.0, 0.2) \cdot avgDem_p/avgDem_1$, where $avgDem_1 = 10$, $avgDem_2 = 6$, $avgDem_3 = 9$, $avgDem_4 = 5$, $avgDem_5 = 8$ and $rand(1.0, 0.2)$ is a random variable with normal distribution, a mean of 1.0 and a standard deviation of 0.2.

Note that the choice of allocating demand to only a few of the time periods is motivated by the aforementioned industrial application in the forest industry, where each logging region is harvested, on average, about four seasons over the ten-period planning horizon. Furthermore, it results in a geographically more dispersed distribution of the demand which creates the need to adjust capacities at the facilities.