

Centre interuniversitaire de recherche sur les réseaux d'entreprise, la logistique et le transport

Interuniversity Research Centre on Enterprise Networks, Logistics and Transportation

# Solving a Dynamic Facility Location Problem with an Application in Forestry

Sanjay Dominik Jena Jean-François Cordeau Bernard Gendron

December 2014

CIRRELT-2014-71

Bureaux de Montréal : Université de Montréal Pavillon André-Aisenstadt C.P. 6128, succursale Centre-ville Montréal (Québec) Canada H3C 3J7 Téléphone: 514 343-7575 Télépcone: 514 343-7121 Bureaux de Québec : Université Laval Pavillon Palasis-Prince 2325, de la Terrasse, bureau 2642 Québec (Québec) Canada G1V 0A6 Téléphone: 418 656-2073 Télécopie : 418 656-2624

www.cirrelt.ca





ÉTS UQÀM

HEC MONTRĒAL





# Solving a Dynamic Facility Location Problem with an Application in Forestry

# Sanjay Dominik Jena<sup>1,†,\*</sup>, Jean-François Cordeau<sup>2,3</sup>, Bernard Gendron<sup>2,4</sup>

- <sup>1</sup> Singapore-MIT Alliance for Research and Technology (SMART), 1 CREATE Way, #10-01 CREATE Tower, Singapore, 138602
- <sup>2</sup> Interuniversity Research Centre on Enterprise Networks, Logistics and Transportation (CIRRELT)
- <sup>3</sup> Department of Logistics and Operations Management, HEC Montréal, 3000 Côte-Sainte-Catherine, Montréal, Canada H3T 2A7
- <sup>4</sup> Department of Computer Science and Operations Research, Université de Montréal, P.O. Box 6128, Station Centre-Ville, Montréal, Canada H3C 3J7

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

**Keywords**. Facility location, dynamic capacity adjustment, Lagrangian relaxation, mixed-integer programming, industrial application.

**Acknowledgements.** The authors would like to thank Antonio Frangioni and Enrico Gorgone for providing the implementation of the bundle method, as well as their valuable advice on its use. The authors are also grateful to MITACS, the Natural Sciences and Engineering Research Council of Canada (NSERC) and the Fonds de recherche du Québec - Nature et technologies (FRQNT) for their financial support. Finally, they also thank Calcul Québec and Compute Canada for providing the computing infrastructure used for the experiments.

<sup>†</sup> This research was carried out while Sanjay Dominik Jena was doing his Ph.D. at Department of Computer Science and Operations Research, Université de Montréal.

Results and views expressed in this publication are the sole responsibility of the authors and do not necessarily reflect those of CIRRELT.

Les résultats et opinions contenus dans cette publication ne reflètent pas nécessairement la position du CIRRELT et n'engagent pas sa responsabilité.

<sup>\*</sup> Corresponding author: sanjay@smart.mit.edu

Dépôt légal – Bibliothèque et Archives nationales du Québec Bibliothèque et Archives Canada, 2014

<sup>©</sup> Jena, Cordeau, Gendron and CIRRELT, 2014

# 1 Introduction

Classical facility location aims at striking a balance between facility construction costs and transportation costs to satisfy customer demands. Operations research practitioners therefore contributed a considerable variety of extensions to classical models to represent real world applications in a more realistic manner, involving the location of hospitals (Vahidnia et al., 2009), telecommunication hubs (Chardaire et al., 1996), schools (Antunes and Peeters, 2001), manufacturing and distributing systems (Min and Melachrinoudis, 1999), and many others. The dynamic adjustment of the capacities over a planning horizon has often been a central issue. Problem extensions have been proposed to allow for the expansion and the reduction of capacity along time (Luss, 1982; Antunes and Peeters, 2001), temporary facility closing (Chardaire et al., 1996; Dias et al., 2006) and the relocation of capacities from one location to another. Other important extensions acknowledged uncertainty in the customer demands (Schütz et al., 2008) or the production capacities themselves (for references, see, e.g., Snyder, 2006). Given the difficulty to solve those problems for real world sized instances, many solution algorithms have been suggested. Exact methods have been proposed for classical variants (Wentges, 1996; Görtz and Klose, 2012), whereas heuristics have proved to be effective for more complex problem variants. Due to the complicated structure of the latter, only a few works have applied methods that provide a bound on the solution quality, such as Benders decomposition and Lagrangian relaxation (Dias et al., 2006; Kim and Kim, 2013). More complex problem variants have therefore been solved by methods such as sophisticated local search (Lee and Dong, 2008; Melo et al., 2011). which, by themselves, do not allow for an assessment of the solution quality.

In this paper, we consider a multi-period facility location problem with multiple commodities and multiple capacity levels that has recently been introduced and applied in the forestry sector by Jena et al. (2012). In the application considered by the authors, a logging company must locate camps to host its workers. The problem involves several different ways to adjust capacity, namely, the expansion of capacity, the temporary closing of parts of the facility and the relocation of facilities from one location to another. Many of these features have already been discussed in early literature. The first multi-period models include those by Ballou (1968) and Wesolowsky (1973). Multiple commodities have also been considered by authors such as Geoffrion (1974) and Warszawski (1973). Modular capacity levels have often been treated by offering a choice of facility size (Lee, 1991; Shulman, 1991; Correia and Captivo, 2003; Gouveia and Saldanha da Gama, 2006), whereas capacity expansion has been discussed in detail by Luss (1982) and has been found to be a crucial feature in many applications (Antunes and Peeters, 2001; Canel et al., 2001; Melo et al., 2006). Wesolowsky and Truscott (1975) have been among the first to consider simple relocation of facilities, followed by several others (Min and Melachrinoudis, 1999; Brotcorne et al., 2003; Melo et al., 2006). While the temporary closing of entire facilities has been modeled in several studies (Van Roy and Erlenkotter, 1982; Chardaire et al., 1996; Canel et al., 2001; Dias et al., 2006), the problem introduced by Jena et al. (2012) was the first to consider the partial closing and reopening of facilities along time. The authors propose a flow based formulation that uses a network structure for each facility location to manage the size of available capacity and the size of temporarily closed capacity of the facility. An integer flow, representing the number of open and closed capacity levels, allows for the closing of open capacity and the reopening of closed capacity. Another feature of the problem is the use of the so-called round-up capacity (RUC) constraints, which imply integer rounding of the total demand for each commodity allocated to the same facility. While this characteristic may correspond to the practice in many industries, to the best of our knowledge, the authors were the first to explicitly model this type of capacity constraints. In the following, we will refer to this problem as the Dynamic Facility Location Problem with Relocation and Partial Facility Closing (DFLP\_RPC) with RUC constraints.

The contribution of this paper is twofold. We first propose a new formulation for the DFLP\_RPC with RUC constraints that is based on a recently introduced modeling technique (Jena et al., 2014a) and yields integrality gaps more than ten times smaller than those of the formulation proposed by Jena et al. (2012). The use of this modeling technique also enables the formulation to make use of a more detailed cost structure. We then develop an efficient heuristic based on Lagrangian relaxation. The heuristic extends the one presented by Jena et al. (2014b) and adapts it to the new problem features, in particular to the partial closing of facilities, the relocation of facilities, and the RUC constraints. The heuristic consists of two optimization stages. In the first stage, Lagrangian relaxation is applied to provide lower and upper bounds for the problem. Then, a restricted mixed-integer programming (MIP) model, based on the Lagrangian solutions,

is solved to improve the final solution quality. Computational results show that the proposed heuristics are capable of finding high quality solutions in short computing times, even for large-scale instances for which general-purpose MIP solvers do not find feasible solutions. Furthermore, given the strength of the proposed formulation, the heuristics provide significant bounds on the quality of the obtained solution.

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

# 2 Problem Description

We consider the problem introduced by Jena et al. (2012), which extends the Capacitated Facility Location Problem in several respects: multiple time periods, multiple (modular) capacity levels and multiple commodity types. Given a set of customers with independent demands for each commodity and time period, the objective is to find the optimal locations and opening schedules for facilities that provide sufficient capacity to satisfy the customer demands at minimal costs. New facilities may be constructed and existing facilities may expand their capacity at any time period. Since a facility may not always require its entire capacity, parts of the facility may be temporarily closed, while other parts remain open. Given that the temporary closing and reopening of capacity is usually much cheaper than the complete shut-down and construction of a facility, this feature may result in a very dynamic opening schedule of the facilities. Facilities may also be relocated from one location to another, assuming: 1) a facility can only be relocated as a whole, not partially; 2) before it is relocated, the entire capacity of a facility has to be closed; 3) facilities cannot be merged at the same location.

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

**Application in Forestry.** The DFLP\_RPC with RUC constraints was motivated by an industrial application in the forestry sector introduced by Jena et al. (2012), where a logging company needs to locate camps to host its workers. Facilities represent logging camps, while customers represent logging regions that specify a total demand for two different commodities: the workforce for wood logging and the workforce for the construction and maintenance of access roads. Demands are specified over a time horizon of five years, each year divided into a summer and a winter season. Logging camps are composed by trailers and therefore have a very flexible structure. The capacity level of a facility thus represents the number of trailers at the camp. The hosting capacity of a logging camp can easily be expanded by adding new trailers. Some trailers may be closed, while others remain open. Trailers are only available for use when they are open. The total number of trailers of a camp, i.e., the sum of open and closed trailers, is also referred to as the number of existing trailers. Demands are specified as the average number of crews working throughout the entire season. It is likely that a crew will only work part of the season in a given region, which leads to a fractional demand. Given that crews always work together, the logging camp must ensure sufficient hosting capacity for the entire crew. The RUC constraints therefore ensure that capacity is modeled in a realistic manner. As an example, let us assume that one crew will work the entire season, while another one will work only

40% of the season at a given site. The total demand for the two crews is thus 1.4 and a logging camp would need to ensure a total capacity for at least [1.4] = 2 crews in order to host the workers of both crews in the same time period.

#### 3 Mathematical Formulations

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

We now present a new formulation for the DFLP\_RPC with RUC constraints, based on a modeling technique referred to as the Generalized Modular Capacities (GMC) formulation (Jena et al., 2014a). This formulation uses binary facility location variables  $y_{\ell_1\ell_2}^{jt}$  to explicitly represent a capacity change from level  $\ell_1$  to  $\ell_2$  at the beginning of time period t at location j. Even though the use of such variables results in larger models than classical flow based modeling techniques, it has been shown that GMC based formulations provide significantly stronger LP relaxation bounds, and therefore facilitate the solution of the problem when using general-purpose MIP solvers.

In the DFLP\_RPC, one needs to simultaneously manage capacity on two levels: the existing capacity and the open capacity. We therefore extend this modeling technique and use binary variables  $y_{\ell_1\ell_2n_1n_2}^{jt}$  that take value 1 if a facility at location j changes its existing capacity from level  $n_1$  to  $n_2$  and its open capacity from level  $\ell_1$  to  $\ell_2$  at the beginning of time period t. Clearly, variables are defined only for  $\ell_1 \leq n_1$  and  $\ell_2 \leq n_2$ . Integer variables  $z_{\ell p}^{jt} \in \mathbb{Z}^+$  represent the total number of blocks of commodity type p reserved at a facility with  $\ell$  open capacity levels, located at  $j \in J$ . Binary variables  $\hat{w}_{\ell n}^{jt}$  indicate whether a facility of size n, open at capacity level  $\ell$ , is closed and relocated from location j to another location before period t. Binary variables  $\check{w}_n^{jt}$  indicate whether a facility of size n is relocated to location j before period t. The continuous variables  $x_{\ell p}^{ijt} \in [0,1]$  denote the fraction of the demand  $d_p^{it}$  satisfied by a facility at location jopen at facility level  $\ell$ .

The objective function coefficients associated to the capacity change decisions  $y_{\ell_1\ell_2n_1n_2}^{jt}$  are given by  $f_{\ell_1\ell_2n_1n_2}^{jt}$  and describe the aggregated costs to change the open capacity of a facility at location j from level  $\ell_1$  to  $\ell_2$  and the existing capacity from level  $n_1$  to  $n_2$  at the beginning of period t, as well as the costs to operate the facility at levels  $\ell_2$  and  $n_2$  throughout time period t. Clearly, this cost structure is much more general than the one of the flow formulation used by Jena et al. (2012) and could therefore represent the capacity transitions in a more realistic manner. However, to obtain a formulation equivalent to that of Jena et al. (2012), we set the cost matrix  $f_{\ell_1\ell_2n_1n_2}^{jt}$  as follows. The number of capacity levels constructed (nFC), the number of capacity levels reopened (nRE) and the

number of capacity levels closed (nCL) that are represented by a decision variable  $y_{\ell_1\ell_2n_1n_2}^{jt}$  can be computed

by:

$$nFC = \max \{0, (n_2 - n_1)\}$$
  

$$nRE = \max \{0, (\ell_2 - \ell_1) - nFC\}$$
  

$$nCL = \max \{0, (\ell_1 - \ell_2) + nFC\}.$$

The cost coefficients are then defined as:

$$f_{\ell_1\ell_2n_1n_2}^{jt} = c_{\ell_2t}^M + c_{(nFC)t}^C + c_{(nRE)t}^{TO} + c_{(nCL)t}^{TC}.$$

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

$$\min \sum_{j \in J} \sum_{\ell_1 \in L} \sum_{\ell_2 \in L} \sum_{n_1 \in L} \sum_{n_2 \in L} \sum_{t \in T} f_{\ell_1 \ell_2 n_1 n_2}^{jt} y_{\ell_1 \ell_2 n_1 n_2}^{jt} \\ + \sum_{i \in I} \sum_{j \in J} \sum_{\ell \in L} \sum_{p \in P} \sum_{t \in T} g_{\ell p}^{ijt} d_p^{it} x_{\ell p}^{ijt} \\ + \sum_{i \in I} \sum_{j \in J} \sum_{\ell \in L} \sum_{p \in P} \sum_{t \in T} (c_\ell^{TC} + \frac{c_n^R}{2}) \hat{w}_{\ell n}^{jt} + \sum_{i \in I} \sum_{j \in J} \sum_{\ell \in L} \sum_{p \in P} (c_\ell^{TC} + \frac{c_n^R}{2}) \hat{w}_{\ell n}^{jt}$$
(1)

$$\sum_{j \in J} \sum_{\ell \in L} \sum_{n \in L} \sum_{t \in T} \sum_{t \in T} \sum_{j \in J} \sum_{n \in L} \sum_{t \in T} \sum_{t \in T} \sum_{i \in T} \sum_{i \in T} \sum_{t \in T} \sum_{i \inT} \sum_{$$

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

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

$$(3)$$

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

$$\tag{4}$$

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

$$\sum_{\ell_1 \in L} \sum_{n_1 \in L} y_{\ell_1 0 n_1 n}^{j(t-1)} + \check{w}_n^{jt} = \sum_{\ell_2 \in L} \sum_{n_2 \in L} y_{0\ell_2 n n_2}^{jt} + \hat{w}_{0n}^{jt}$$

$$\forall j \in J, \ \forall n \in L \setminus \{0\}, \ \forall t \in T \setminus \{1\}$$
(6)

$$\sum_{\ell_2 \in L} \sum_{n_2 \in L} y_{\ell^j \ell_2(n_1 = \ell^j) n_2}^{j_1} = 1 \ \forall j \in J^0$$
(7)

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

$$\tag{8}$$

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

$$\tag{9}$$

$$\begin{aligned} x_{\ell p}^{ijt} \ge 0 \quad \forall i \in I, \quad \forall j \in J, \quad \forall \ell \in L, \quad \forall p \in P, \quad \forall t \in T \\ y_{\ell p}^{it} = \{0, 1\}, \quad \forall i \in I, \quad \forall i \in J, \quad \forall \ell_1 \in L, \quad \forall \ell_2 \in L \end{aligned}$$
(10)

$$y_{\ell_1\ell_2n_1n_2}^{jt} \in \{0,1\} \quad \forall i \in I, \quad \forall j \in J, \quad \forall \ell_1 \in L, \quad \forall \ell_2 \in L,$$

$$\forall n \in I, \quad \forall n \in I, \quad \forall t \in T.$$

$$(11)$$

$$\forall n_1 \in L, \ \forall n_2 \in L, \ \forall t \in I \tag{11}$$

$$\hat{w}_{\ell n}^{J^{\iota}} \in \{0, 1\} \quad \forall j \in J, \quad \forall \ell \in L, \quad \forall n \in L, \quad \forall t \in T$$

$$(12)$$

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

$$\tag{13}$$

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

$$(14)$$

The objective function (1) minimizes the total costs for changing the capacity levels and allocating the demand. Note that the relocation costs  $c_n^R$  are equally split on variables  $\hat{w}$  and  $\check{w}$ . Hereby, we intend to better use both variables within the Lagrangian relaxation. Constraints (2) are the demand constraints for the customers. Constraints (3) and (4) are the round-up capacity constraints at the facilities that first round

up the total demand (specified in number of blocks) and then use the rounded number of blocks to determine the total capacity necessary at the facility. Constraints (5) and (6) are the flow conservation constraints that link the capacity change variables in consecutive time periods. These constraints also allow the use of the relocation variables  $\hat{w}_{\ell n}^{jt}$  and  $\check{w}_{n}^{jt}$  to remove flow from one location and add it to another location, respectively. Constraints (7) are the flow initialization constraints that specify that exactly one capacity level is chosen at the beginning of the planning horizon. Constraints (8) guarantee that exactly one capacity change variable is selected at each time period. Finally, constraints (9) are the relocation linking constraints that match the outgoing and incoming relocations of facilities of the same size. The flow conservation constraints (5) – (7) represent a network flow structure, which is illustrated in Figure 1 for a small example with four time periods and two capacity levels. Each node represents the number of open and existing capacity (*open capacity level* / *existing capacity level*). The binary capacity change variables are represented by arcs, which allow for the construction of new capacity, as well as the closing and reopening of existing capacity.



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

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

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

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

The Strong Inequalities (SI) (15), typically used in facility location and network design problems (Van Roy, 1986; Gendron and Crainic, 1994), are known to provide a tight upper bound for the demand assignment variables. The valid inequalities (16) are 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.

# 4 Lagrangian Heuristics

When applying Lagrangian relaxation to capacitated facility location problems, it is common to relax either the capacity constraints (Van Roy and Erlenkotter, 1982; Barcelo et al., 1990) or the demand constraints (Shulman, 1991; Beasley, 1993; Wu et al., 2006). Jena et al. (2014b) applied the latter Lagrangian relaxation to the GMC formulation. By relaxing the demand constraints, the Lagrangian subproblem decomposes into independent subproblems, one for each candidate facility location. These independent subproblems can then be efficiently solved by dynamic programming.

When the relocation of facilities is allowed, as it is the case for our problem, the relocation constraints (9) are an additional link between the candidate facility locations. Therefore, the relaxation of the demand constraints is not sufficient to decompose the Lagrangian subproblem by location. There are two possibilities to overcome this issue. One can relax both the demand constraints (2) and the relocation constraints (9) in order to obtain a subproblem that can be decomposed by location. Alternatively, one can relax only the demand constraints (2). The remaining Lagrangian subproblem then still includes the relocation linking constraints (9) and, therefore, cannot be decomposed by location. Instead, the subproblem may be transformed into a pure integer program and be solved by a generic MIP solver. In our computational experiments, the latter approach has not been competitive with the former one, given that the MIP solver may still take considerable time to solve the resulting integer program. We therefore report on the first approach, relaxing both the demand constraints (2) and the relocation linking constraints (9).

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

$$\begin{split} L(\alpha,\beta) &= \min \sum_{j \in J} \sum_{\ell_1 \in L} \sum_{\ell_2 \in L} \sum_{n_1 \in L} \sum_{n_2 \in L} \sum_{t \in T} f_{\ell_1 \ell_2 n_1 n_2}^{jt} y_{\ell_1 \ell_2 n_1 n_2}^{jt} \\ &+ \sum_{j \in J} \sum_{\ell \in L} \sum_{n \in L} \sum_{t \in T} (c_\ell^{TC} + \frac{c_n^R}{2} - \beta_{nt}) \hat{w}_{\ell n}^{jt} + \sum_{j \in J} \sum_{n \in L} \sum_{t \in T} (\frac{c_n^R}{2} + \beta_{nt}) \check{w}_n^{jt} \\ &+ \sum_{i \in I} \sum_{j \in J} \sum_{\ell \in L} \sum_{p \in P} \sum_{t \in T} \tilde{c}_{\ell p}^{ijt} x_{\ell p}^{ijt} + \sum_{i \in I} \sum_{p \in P} \sum_{t \in T} \alpha_{ipt} \\ s.t. \ (3) - (8), (10) - (14). \end{split}$$

#### 4.1 Solution of the Lagrangian Subproblem

We separate the Lagrangian subproblem into |J| independent subproblems, one for each potential facility location for a fixed set of Lagrange multipliers  $\alpha$  and  $\beta$ . The Lagrangian subproblem is then defined as  $L(\alpha,\beta) = \sum_{j\in J} L_j(\alpha,\beta) + \sum_{j\in J} \sum_{p\in P} \sum_{t\in T} \alpha_{ipt}$ , where  $L_j(\alpha,\beta)$  corresponds to the problem of finding the optimal opening schedule for the capacities of facility j with modified demand allocation costs  $\tilde{c}_{\ell p}^{ijt}$ . To solve this problem, we extend the dynamic programming algorithm presented by Jena et al. (2014b). Let  $O_j^{\alpha,\beta}(\ell,n,t)$  denote the value of the optimal opening schedule from time period 1 to t, including the costs to satisfy the customer demand during these time periods and assuming that a facility of size n, open at capacity level  $\ell$  is available at the end of time period t. To compute these values, we need to evaluate the following four combinations of incoming and outgoing relocations:

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

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

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

$$\hat{C}_{RelocIN}(\ell, n, t) = \min_{1 \le n_1 \le n} \{ \frac{c_{n_1}^R}{2} + c_{(nRE)t}^{TO} + c_{(nCL)t}^{TC} + \beta_{n_1t} + c_{j(n-n_1)}^c | \\ nRE = \max\{0, \ell - n + n_1\}, nCL = \max\{0, n - n_1 - \ell\} \}$$
$$\hat{C}_2(\ell, n, t) = \hat{C}_{RelocIN}(\ell, n, t) + c_{\ell t}^M + O_j^{\alpha, \beta}(0, 0, t - 1).$$

3. No incoming relocation, outgoing relocation. A facility has been relocated to another location and a facility of size  $\langle \ell, n \rangle$  has been constructed afterwards:

$$\hat{C}_{RelocOUT} = \min_{1 \le n_1 \le q, 0 \le \ell_1 \le n_1} \{ c_{\ell_1 t}^{TC} + \frac{c_{n_1}^R}{2} - \beta_{n_1 t} + O_j^{\alpha, \beta}(\ell_1, n_1, t-1) \}$$
$$\hat{C}_3(\ell, n, t) = \hat{C}_{RelocOUT} + f_{j0\ell 0nt}.$$

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

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

Based on these four cases, the optimal value for  $O_i^{\alpha}(\ell, n, t)$  is computed as:

$$O_{j}^{\alpha,\beta}(\ell,n,t) = L_{j}^{\alpha}(\ell,t) + \min\{\hat{C}_{1}(\ell,n,t), \hat{C}_{2}(\ell,n,t), \hat{C}_{3}(\ell,n,t), \hat{C}_{4}(\ell,n,t)\},$$
(17)

where  $L_j^{\alpha}(\ell, t)$  is the cost of the optimal demand allocation at facility j with  $\ell$  open capacity levels at period t, which is computed as shown below.

We solve the subproblem for location j by selecting the minimum among all possible facility sizes:

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

Note that, using the RUC constraints, the Lagrangian subproblem does not have the integrality property. The lower bound provided by the Lagrangian subproblem may thus be better than the bound provided by the LP relaxation of the original problem.

**Computation of the Optimal Demand Allocation.** When using common capacity constraints, as used in the Capacitated Facility Location Problem, the optimal demand allocation  $L_j^{\alpha}(\ell, t)$ , taking into account the adjusted demand allocation costs  $\tilde{c}_{\ell p}^{ijt}$  for a given  $\alpha$ , j,  $\ell$  and t, can be obtained by solving a continuous knapsack problem (Shulman, 1991). However, when using the RUC constraints, finding the optimal demand allocation is identical to solving the following MIP:

$$\min \sum_{i \in I} \sum_{p \in P} \tilde{c}_{\ell p}^{ijt} x_{\ell p}^{ijt}$$
  
s.t. 
$$\sum_{i \in I} d_p^{it} x_{\ell p}^{ijt} \leq z_{\ell p}^{jt} \quad \forall p \in P$$
$$\sum_{p \in P} s_p z_{\ell p}^{jt} \leq u_{\ell}^{j}$$
$$x_{\ell p}^{ijt} \leq 1 \quad \forall i \in I, \quad \forall p \in P$$
$$x_{\ell p}^{ijt} \in \mathbb{R}^+ \quad \forall i \in I, \quad \forall p \in P$$
$$z_{\ell p}^{jt} \in \mathbb{Z}_0^+ \quad \forall p \in P.$$

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

1. First, all feasible integer values for  $z_{\ell p}^{jt}$  are identified. For each p, the feasible integer values that  $z_{\ell p}^{jt}$  may take are given by set  $\Omega_p = \left\{0, 1, 2, ..., \left\lfloor \frac{u_{\ell}^j}{s_p} \right\rfloor\right\}$ .

- 2. The costs for each of the integer values of  $z_{\ell p}^{jt}$  are computed by solving a continuous knapsack problem for each  $p \in P$ , by rearranging the demand nodes  $d_{ipt}$  in increasing order of their ratio of adjusted transportation costs and demand quantity, i.e.,  $\frac{\tilde{c}_{\ell p}^{ijt}}{d_p^{it}}$ , and by serving demands until either the entire capacity, given by the integer value of  $z_{\ell p}^{jt}$ , is filled or a demand node with a positive adjusted transportation cost is met.
- 3. A Multiple-Choice Knapsack Problem (Martello and Toth, 1990) is solved by using dynamic programming, in which each integer value for a  $z_{\ell p}^{jt}$  variable represents an object with a cost coefficient given by the solution of the previously solved continuous knapsack. The weight of object  $z_{\ell p}^{jt}$  is given by  $s_p z_{\ell p}^{jt}$ and the total knapsack capacity is given by  $u_{\ell}^{j}$ . Exactly one object for each  $p \in P$  is selected.

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

### 4.2 Solution of the Lagrangian Dual

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

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

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

$$\begin{split} \gamma_{ipt}^{k} &= 1 - \sum_{j \in J} \sum_{\ell \in L} x_{\ell p}^{ijt} \ \forall i \in I, \ \forall p \in P, \ \forall t \in T \\ \mu_{nt}^{k} &= \sum_{j \in J} \check{w}_{n}^{jt} - \sum_{j \in J} \sum_{\ell \in L} \hat{w}_{\ell n}^{jt} \ \forall n \in L \backslash \left\{0\right\}, \ \forall t \in T. \end{split}$$

Due to its strong convergence properties, we chose to use a bundle method to solve the Lagrangian dual. The implementation based on Frangioni (2005) uses a subset of the tuples  $\langle L(\alpha^s), \gamma^s \rangle$  with  $s \in B$ , where B is referred to as the bundle of subgradients  $\gamma^s$ . From the primal view point, the following quadratic problem has to be solved at each iteration (Frangioni and Gallo, 1999):

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

where R is the so-called trust region for the tentative ascent direction, 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  $\theta^s$ , given for each bundle member, hold valuable information and can be used to construct feasible integer solutions (see Section 4.4). The tentative ascent direction is then computed by the convex combination of the subgradients, using the convex multipliers  $\theta$ . 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.

#### 4.3 Generation of Upper Bounds

At each iteration, a feasible solution is generated based on the current Lagrangian primal solution. This feasible solution serves as an upper bound for the optimal integer solution and directly impacts the convergence of the 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. To generate feasible solutions, we extend the approach followed by Jena et al. (2014b) by also considering partial closing and reopening of facilities, relocation of facilities and the RUC constraints. The Lagrangian solution provides a facility opening schedule for the entire planning horizon, defined by a capacity level pair  $(\ell'_{jt}/n'_{jt})$  for each j and t, as well as the corresponding demand allocation. As the demand constraints (2) have been relaxed, the set of demands  $d_{ipt}$  can be separated into three subsets:  $\Sigma_1$ ,  $\Sigma_2$  and  $\Sigma_3$  denote the demands defined by triplets  $\langle i, p, t \rangle$ , which are exactly met, over-served and under-served, respectively:

$$\Sigma_{1} = \left\{ \langle i, p, t \rangle : \sum_{j \in J} x_{(\ell'_{jt})p}^{ijt} = 1 \right\}, \Sigma_{2} = \left\{ \langle i, p, t \rangle : \sum_{j \in J} x_{(\ell'_{jt})p}^{ijt} > 1 \right\}$$
  
and  $\Sigma_{3} = \left\{ \langle i, p, t \rangle : \sum_{j \in J} x_{(\ell'_{jt})p}^{ijt} < 1 \right\}.$ 

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

- 1. Identify feasible relocation pairs: Feasible pairs of relocation decisions are identified by matching the outgoing relocations  $\hat{w}_{\ell n}^{jt}$  and incoming relocations  $\tilde{w}_{n}^{jt}$  from the Lagrangian solution. For each pair of facility size n and time period t, we choose the maximum number of facility matches j' and j'' (with  $j' \neq j''$ ) among the  $\hat{w}_{j'\ell nt}$  and  $\tilde{w}_{j''nt}$  decisions made in the Lagrangian solution. The procedure to find these pairs considers facilities without a specific order and excludes infeasible configurations, i.e., no outgoing relocation of a facility is smaller than a previously incoming relocation and subsequent incoming relocations at the same location are separated by outgoing relocation. For locations for which no relocation pairs have been selected, we generate the optimal opening schedule without relocations.
- 2. Reduce demand allocation: For each  $\langle i, p, t \rangle \in \Sigma_2$ , all facility/size pairs  $(j, (\ell'_{jt}))$  are sorted in decreasing order of their allocation costs  $g_{\ell p}^{ijt}$ . The allocated flow is removed until the total allocated demand for  $\langle i, p, t \rangle$  equals 1.
- 3. 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  $\ell'_{jt} < n'_{jt}$ , we incrementally increase  $\ell'_{jt}$  until the missing capacity is covered or  $\ell'_{jt} = n'_{jt}$ ; if no facility exists, we incrementally increase both  $\ell'_{jt}$  an  $n'_{jt}$  until the missing capacity is covered or the maximum capacity level for this facility is reached. For any time period t' > t, the existing capacity level  $n'_{it'}$  is increased to the new level  $n'_{jt}$  if  $n'_{jt'} < n'_{jt}$ .
- 4. Increase the demand allocation: For each  $\langle i, p, t \rangle \in \Sigma_3$ , all facility/size pairs  $(j, (\ell'_{jt}))$  with remaining capacity are sorted in increasing order of their allocation costs  $g_{\ell p}^{ijt}$ . Demand is allocated to these pairs until the total allocated demand for  $\langle i, p, t \rangle$  equals 1. Note that, due to the rounding in the capacity constraints, certain demand may be allocated to a facility without consuming additional capacity, if the facility has a commodity p block that is not yet completely filled. Furthermore, allocating a new commodity p block is subject to capacity availability at the facility for the entire lot.
- 5. Reduce unused capacities of open facilities: For each facility, a dynamic programming algorithm, similar to the one used to solve the Lagrangian subproblem, computes the optimal opening schedule that guarantees sufficient capacity to satisfy the demand allocated to that facility. The algorithm takes into account the total lot size for each commodity reserved at the facility, not only the allocated customer demands.

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

#### 4.4 Restricted MIP model

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

As explained in Section 4.2, the bundle method provides a multiplier  $\theta^s$  for each Lagrangian solution s such that  $\sum_s \theta^s = 1$ . The value  $\theta^s$  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  $y_{j\ell nt}^P = \sum_s \theta^s y_{j\ell nt}^s$ , where  $y_{j\ell nt}^s$  is 1 if solution s selects capacity level pair  $(\ell, n)$  for location j at period t. Furthermore, let  $L_{jt}^R$  be the set of  $(\ell, n)$  pairs for location j and period t available in the restricted MIP. The restricted MIP is then defined as follows:

- Selection of capacity levels. If the decision for location j and period t is not fixed,  $L_{jt}^R$  is composed by the  $n^S$  capacity level pairs  $(\ell, n)$  that have the highest  $y_{i\ell nt}^P$  values, with  $y_{i\ell nt}^P > 0.001$ .
- Defining the set of capacity transitions. Decision variables  $y_{\ell_1\ell_2n_1n_2}^{jt}$  are defined for all combinations between  $(\ell_1, n_1)$  and  $(\ell_2, n_2)$ , with  $(\ell_1, n_1) \in L_{jt}^R$  and  $(\ell_2, n_2) \in L_{j(t+1)}^R$ , if available in the original RPCr-GMC formulation.

Relocation decisions are added to the restricted MIP by computing the probabilities for outgoing and incoming relocations  $\hat{w}_{j\ell nt}^P$  and  $\check{w}_{jnt}^P$ , respectively. We set  $\hat{w}_{j\ell nt}^P = \sum_s \theta^s \hat{w}_{j\ell nt}^s$ , where  $\hat{w}_{j\ell nt}^s$  is 1 if solution s relocates a facility of size n open at level  $\ell$  from location j to another location at period t. In the same way, we set  $\check{w}_{jnt}^P = \sum_s \theta^s \check{w}_{jnt}^s$ , where  $\check{w}_{jnt}^s$  is 1 if solution s relocates a facility of size n open at level  $\ell$  from location s relocates a facility of size n open at level  $\ell$  from location s relocates a facility of size n open at level  $\ell$  from location j to another location at period t. All relocation variables with their corresponding  $\hat{w}_{j\ell nt}^P$  and  $\check{w}_{jnt}^P$  greater than or equal to 0.001 are added to the restricted model. To ensure their feasibility with respect to the flow conservation constraints, certain capacity levels are added to the sets of available capacity levels  $L^R$ . To be precise, when adding a relocation decision  $\check{w}_n^{jt}$  to the restricted MIP, capacity level pair (0, n) is added to  $L_{j(t-1)}^R$  and capacity level pairs (0, n) and (n, n) are added to  $L_{jt}^R$  to ensure that the flow conservation constraints contain the capacity transition variables  $y_{\ell_1 \ell_2 n_1 n_2}^{jt}$  that either maintain the facility closed or reopen it at its maximum capacity level n.

# 5 Computational Experiments

In this section, the performance of the Lagrangian heuristic and that of the MIP solver CPLEX are evaluated and compared by means of computational experiments. 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.

In addition to the DFLP\_RPC with RUC constraints, we consider two simplified problem variants, which are defined next. Then, we explain how test instances are generated. The integrality gaps of the different problem variants and formulations are then analyzed. Finally, computational results are presented to compare the performance of the Lagrangian heuristics and that of CPLEX.

#### 5.1 Problem Variants

The RUC constraints are a particular characteristic of the industrial application considered here. The majority of facility location models in the literature uses classical capacity constraints such as the following:

$$\sum_{i \in I} d_p^{it} x_{\ell p}^{ijt} \le \sum_{\ell_1 \in L} \sum_{n_1 \in L} \sum_{n_2 \in L} u_\ell^j y_{\ell_1 \ell n_1 n_2}^{jt} \quad \forall j \in J, \ \forall \ell \in L, \ \forall p \in P, \ \forall t \in T.$$

$$(18)$$

Computational results are therefore reported for the DFLP\_RPC with RUC constraints and two simplified problem variants, both using classical capacity constraints (18), to illustrate the impact of the RUC constraints on the difficulty of solving the problem. The DFLP\_RPC is defined as the DFLP\_RPC with RUC constraints, but with classical capacity constraints (18). The GMC based formulation for this problem, referred to as the RPC-GMC formulation, is therefore defined by objective function (1) and constraints (2), (5) – (13) and (18). The Dynamic Facility Location Problem with Partial Facility Closing (DFLP\_PC) is defined in the same way as the DFLP\_RPC, but without the relocation of facilities. Its GMC based formulation is referred to as the PC-GMC formulation and is defined by objective function (1) and constraints (2), (5) – (8), (10) – (13) and (18), but without the relocation variables  $\hat{w}_{\ell n}^{jt}$  and  $\check{w}_{n}^{jt}$  and the RUC integer variables  $z_{\ell p}^{jt}$ . We refer to the flow formulation proposed by Jena et al. (2012) as the RPCr-2i formulation. Based on

We refer to the flow formulation proposed by Jena et al. (2012) as the RPCr-2i formulation. Based on this formulation, we also define formulations for the DFLP\_PC and the DFLP\_RPC with classical capacity constraints. We denote the corresponding formulations as the PC-2i and RPC-2i, respectively.

When applying Lagrangian relaxation to the simplified problem variants with classical capacity constraints, the optimal demand allocation in the subproblem becomes significantly easier to compute. Instead of solving a multiple-choice integer knapsack problem when RUC constraints are used, classical capacity constraints involve only the solution of simple continuous knapsack problems. This reduces the computational complexity to solve the Lagrangian subproblem. Furthermore, for the DFLP\_PC, the dynamic programming algorithm to solve the optimal opening schedule does not consider relocation decisions. In the same way, when generating feasible upper bounds, the first step in Section 4.3, matching incoming and outgoing relocation decisions, is ignored.

#### 5.2 Test Instances

Test instances have been generated by following a scheme similar to that described in Jena et al. (2014a). 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: 300 km, 380 km and 450 km. We consider two different demand scenarios. In both scenarios, the demand for each of the customers is randomly generated and randomly distributed over time. The two scenarios differ in their total demand summed over all customers in each time period. In the first scenario, the total demand is similar in each time period. The second scenario assumes that the total demand follows strong variations along time and therefore varies at each time period. The number of commodities |P| has been selected such that  $|P| \in \{1,3,5\}$ . The demands for the second to fifth commodities are based on the demand for the first commodity. To be precise, the demand  $d_{ipt}$  for  $p \ge 2$ is computed as  $d_p^{it} = d_1^{it} \cdot r_p \cdot avgDem_p/avgDem_1$ , where  $avgDem_1 = 10$ ,  $avgDem_2 = 6$ ,  $avgDem_3 = 9$ ,  $avgDem_4 = 5$ ,  $avgDem_5 = 8$ , and  $r_p \sim N(1.0, 0.2)$  is a normally distributed random variable with mean of 1.0 and standard deviation of 0.2. Construction and operational costs follow concave cost functions, i.e., they involve economies of scale. 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 A for a detailed description of the parameters used to generate the instances.

#### 5.3 Analysis of the Integrality Gaps

We now assess the strength of the proposed formulations for each of the three problem variants by means of their integrality gaps. Table 1 summarizes, for each of the three problem variants, the integrality gaps provided by the 2i and GMC based formulations. The reported integrality gaps are the average over the instances of the same size for which all six formulations have solved the LP relaxation within 12 hours of computing time and for which the optimal integer solution is known within 0.1%. This number is indicated in column "# inst". The results demonstrate the same trend for all three problem variants. The GMC based formulations provide significantly lower integrality gaps, on average 13 to 30 times lower than those provided by the 2i formulations. Note, however, that the strong bound comes with the price of higher computing

times when solving the LP relaxation with CPLEX. For example, for the DFLP\_RPC with RUC constraints, solving the LP relaxation (limited to 12hs) of the GMC based formulation takes on average over all 540 instances about 17,085 seconds, while the 2i formulation takes only 4,879 seconds. Furthermore, CPLEX does not provide any bound for 177 instances when using the GMC formulation, mostly due to memory limitations, while this problem occurs for only 52 instances when using the 2i formulation.

Even though CPLEX struggles to solve the LP relaxation for many of the instances, the Lagrangian relaxation may provide bounds at least as good as the LP relaxation in a more efficient manner. In fact, for the DFLP\_RPC with RUC constraints, the Lagrangian subproblem does not have the integrality property when RUC constraints are used. The lower bound provided by the Lagrangian relaxation may therefore be better than the LP relaxation bound of the original formulation. An analysis showed that, for the 380 instances for which the LP relaxation has been solved with CPLEX within a total of 12 hours, the Lagrangian relaxation, limited to 500 iterations and 2 hours of computing time, provided a better bound for all instances. The bound provided by the Lagrangian relaxation has been on average 1.98% and for some instances up to 23.99% higher than the LP relaxation bound. The last column of Table 1 indicates the deviation of the lower bound provided by the bundle method of the Lagrangian heuristic from the upper bounds for the same set of instances. While the average integraliy gap provided by the 2i formulation has been assessed as 2.89%, the Lagrangian relaxation was able to decrease the integrality of the GMC formulation from 0.22% to an average deviation of 0.13%. As a result, the Lagrangian heuristic is likely to provide significant bounds on the quality of the solutions obtained.

	Instance	#	DFL	$P_PC$	DFLF	$P_RPC$	DFLP_RPC w		/ RUC
q	size	inst.	2i	GMC	2i	GMC	2i	GMC	LRH
3	50/50	13	3.34	0.57	4.96	0.59	4.97	0.60	0.62
	50/200	18	1.06	0.04	1.52	0.05	1.52	0.05	0.09
	100/100	17	2.50	0.10	3.93	0.09	3.93	0.09	0.12
	100/400	18	0.98	0.01	1.45	0.01	1.45	0.01	0.03
	150/150	17	2.08	0.07	3.46	0.05	3.46	0.05	0.06
	150/600	14	0.95	0.01	1.46	0.01	1.46	0.01	0.04
	200/200	17	1.89	0.05	3.16	0.03	3.16	0.03	0.06
	200/800	9	0.84	0.02	1.40	0.02	1.40	2.77	0.05
	250/250	17	1.69	0.02	2.97	0.02	2.97	0.02	0.05
	250/1000	5	0.76	0.02	1.26	0.03	1.27	0.03	0.05
	Avg All	145	1.68	0.09	2.67	0.08	2.67	0.26	0.11
5	50/50	4	4.98	0.95	6.22	0.81	6.22	0.81	0.88
	50/200	11	1.80	0.11	2.29	0.10	2.30	0.11	0.16
	100/100	9	3.47	0.18	4.86	0.22	4.86	0.22	0.29
	100/400	11	1.58	0.01	2.07	0.01	2.07	0.01	0.04
	150/150	6	3.07	0.10	4.55	0.07	4.55	0.07	0.09
	150/600	4	1.49	0.01	2.11	0.01	2.11	0.01	0.03
	200/200	7	2.67	0.07	3.95	0.02	3.95	0.02	0.07
	200/800	1	1.64	0.01	2.34	0.01	2.34	0.01	0.02
	250/250	5	2.24	0.02	3.60	0.03	3.60	0.02	0.07
	250/1000	2	1.47	0.02	2.16	0.03	2.16	0.03	0.04
	Avg All	60	2.45	0.13	3.41	0.12	3.41	0.12	0.16
10	100/400	1	3.01	0.03	3.10	0.00	3.14	0.04	0.07
	Avg All	1	3.01	0.03	3.10	0.00	3.14	0.04	0.07
All	Avg All	<b>206</b>	1.91	0.10	2.89	0.09	2.89	0.22	0.13

Table 1: Comparison of average integrality gaps (in %) for the three problem variants for instances where the optimal integer solution is known and the LP relaxation of all formulations has been solved.

## 5.4 CPLEX Optimization

Even though the LP relaxation of the 2i formulations is solved much faster than that of the GMC formulations, the latter provide a substantially better bound. The GMC formulations therefore perform much better than the 2i formulations when the MIP solver is used to find the optimal integer solutions. We now compare the performance of the two formulations for the instances for which both formulations found feasible solutions within the given time limit. The time limit has been set to three hours. The results for the DFLP\_RPC with RUC constraints are given in Table 2. Comparing on the same set of instances, the GMC formulation clearly outperforms the 2i formulation regarding the average and maximum deviations from the best known lower bound, as well as the computing times. In addition to the better solution quality, the GMC based formulation finds feasible solutions for more instances. Feasible solutions were found for 131 instances when using the RPCr-2i formulation, while the the RPCr-GMC formulation found feasible solutions for 251 instances.

For the simplified problem variants, similar results have been observed. Here, the time limit has been set to two hours. The PC-2i formulation for the DFLP\_PC reports an average optimality gap of 1.09%, a maximum gap of 90.83% and an average solution time of 2,768 seconds, whereas the the PC-GMC formulation results in a lower average gap of 0.16%, a maximum gap of 3.78% and an average solution time of 1,613 seconds. For the DFLP\_RPC, the RPC-2i formulation resulted in an average gap of 0.44%, a maximum gap of 4.83% and an average solution time of 1,956 seconds. Again, the RPC-GMC formulation reported superior performance with an average gap of 0.21% and a maximum gap of 1.39% in an average solution time of only 592 seconds. Feasible solutions were found for 323 instances when using the PC-2i formulation and for 291 instances when using the PC-GMC formulation. For the DFLP\_RPC, the RPC-2i formulation led to feasible solutions for 179 instances, whereas the RPC-GMC formulation found feasible solutions for 285 instances. This illustrates the increasing difficulty to solve the different problem variants as facility relocation is allowed and RUC constraints are used.

			RPCr-2i			RPCr-GMC			
q	Instance	# inst.	Avg %	Max %	Time	Avg %	Max %	Time	
	size		$Gap^*$	$\operatorname{Gap}^*$	(sec)	$Gap^*$	$\operatorname{Gap}^*$	(sec)	
3	50/50	16	0.20	1.48	2,561.5	0.17	1.50	2,393.1	
	50/200	18	0.00	0.01	1,398.4	0.00	0.01	733.3	
	100/100	16	0.08	1.22	1,795.3	0.02	0.21	854.9	
	100/400	12	0.16	1.92	2,968.2	0.00	0.01	529.3	
	150/150	13	0.01	0.02	2,324.5	0.00	0.01	459.8	
	150/600	6	1.65	4.75	7,200.5	0.01	0.02	2,682.5	
	200/200	13	0.04	0.20	3,547.5	0.22	2.79	1,721.2	
	250/250	7	0.33	1.29	5,403.0	0.00	0.01	1,443.4	
	Avg All	101	0.19	4.75	2,850.1	0.06	2.79	1,248.2	
5	50/50	6	0.52	2.32	3,819.7	0.19	0.61	2,607.5	
	50/200	8	0.00	0.01	1,762.8	0.00	0.01	347.1	
	100/100	5	0.01	0.01	1,972.6	0.01	0.01	369.6	
	100/400	3	0.12	0.35	3,744.3	0.00	0.00	745.0	
	150/150	4	0.06	0.22	4,819.3	0.04	0.15	2,395.5	
	200/200	2	0.00	0.00	3,705.0	0.00	0.01	1,245.5	
	Avg All	<b>28</b>	0.13	2.32	3,028.7	0.05	0.61	1,234.9	
10	50/200	1	0.28	0.28	7,201.0	0.34	0.34	7,200.0	
	Avg All	1	0.28	0.28	$7,\!201.0$	0.34	0.34	7,200.0	
All	Avg All	130	0.18	4.75	2,922.0	0.06	2.79	1,291.1	

Table 2: CPLEX results comparing the two formulations for the DFLP\_RPC with RUC constraints, considering instances where both formulations found feasible solutions.

## 5.5 Peformance of the Lagrangian Heuristic

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

performed in the initialization phase). In a final optimization phase, we solve the restricted MIP to improve the solution quality.

The following experiments allow for a total of 3 hours of computing time. For all experiments, a 0.01% optimality stopping criterion has been used. Table 3 presents the results for the Lagrangian heuristic applied to the DFLP\_RPC with RUC constraints. The results are given for two configurations of the heuristic. The first configuration uses only the bundle method, whereas the second configuration adds the restricted MIP model afterwards. The restricted MIP has been used with parameter  $n^S = 10$ . For the second configuration, the restricted MIP is started after 2 hours at the latest. The last column for each configuration indicates the number of instances for which the heuristic proved optimality within 1%. Given the strong lower bounds provided by the heuristic, the heuristic proves optimality within 1% for 379 of the 540 instances.

			Bund	le only		Bundle + R-MIP				
q	Instance	Avg	Max	Time	# 1 %	Avg	Max	Time	# 1 %	
	size	Gap %	$\operatorname{Gap}$ %	(sec)	$_{\mathrm{opt}}$	Gap %	$\operatorname{Gap}$ %	(sec)	$_{\mathrm{opt}}$	
3	50/50	3.77	11.64	325.4	1	0.31	1.22	4,142.9	9	
	50/200	0.78	1.54	499.1	15	0.53	0.96	268.3	18	
	100/100	1.75	6.14	858.7	8	0.40	0.87	273.2	18	
	100/400	0.85	2.82	914.9	16	0.59	0.92	650.8	18	
	150/150	1.86	5.13	1,559.9	8	0.29	0.99	610.8	18	
	150/600	1.02	2.86	$1,\!291.0$	14	0.70	0.98	760.5	18	
	200/200	1.89	6.40	2,576.7	10	0.42	0.94	1,098.2	18	
	200/800	1.47	4.70	2,479.1	11	0.51	1.00	1,707.8	18	
	250/250	1.54	6.44	$2,\!686.9$	10	0.40	0.93	1,381.2	18	
	250/1000	1.88	4.67	5,942.3	6	0.29	1.00	4,081.3	18	
	Avg All	1.68	11.64	1,913.4	99	0.44	1.22	$1,\!497.5$	171	
5	50/50	9.18	20.01	639.5	1	1.70	4.57	6,763.3	2	
	50/200	1.80	4.31	3,259.6	5	0.49	1.32	1,907.4	16	
	100/100	6.21	20.14	2,382.9	1	0.73	2.26	3,825.7	10	
	100/400	1.16	3.30	2,798.1	13	0.50	0.99	$1,\!605.0$	18	
	150/150	5.21	9.84	4,834.0	1	0.60	1.50	1,943.1	15	
	150/600	1.09	2.98	3,419.6	10	0.43	0.95	2,339.4	18	
	200/200	3.99	9.31	6,061.1	2	0.45	1.20	2,948.4	17	
	200/800	1.71	4.96	4,154.0	9	0.53	0.98	2,900.0	18	
	250/250	3.15	8.80	$5,\!849.7$	5	0.35	0.98	3,313.8	18	
	250/1000	2.05	4.35	$7,\!494.4$	5	0.30	0.97	5,024.0	18	
	Avg All	3.56	20.14	4,089.3	52	0.60	4.57	3,224.8	150	
10	50/50	15.93	27.86	1,770.1	0	6.38	20.19	7,404.5	0	
	50/200	9.80	18.18	5,830.8	0	3.45	18.18	8,189.3	2	
	100/100	17.78	29.99	$5,\!134.7$	0	4.99	20.86	$8,\!427.6$	0	
	100/400	5.85	13.43	7,882.2	2	2.24	13.43	6,213.4	12	
	150/150	18.42	29.05	7,207.3	0	6.04	29.05	9,140.5	1	
	150/600	5.03	10.94	8,013.6	3	2.88	10.94	6,707.2	11	
	200/200	17.15	26.92	8,040.9	0	5.16	26.26	9,010.6	1	
	200/800	4.65	11.91	9,568.7	0	2.96	11.91	$7,\!356.3$	12	
	250/250	15.21	27.29	8,357.2	0	6.26	27.29	8,666.2	4	
	250/1000	3.05	5.68	$10,\!893.3$	1	0.86	5.68	$7,\!598.8$	15	
	Avg All	11.29	29.99	7,269.9	6	4.11	29.05	7,864.3	58	
All	Avg All	5.51	29.99	$4,\!424.2$	157	1.72	29.05	4,194.1	379	

Table 3: Results for the Lagrangian heuristic for all 540 instances for the DFLP\_RPC with RUC constraints.

Table 4 provides a direct comparison between the results of CPLEX and the Lagrangian heuristic. The MIP solver does not find feasible solutions for about half of the instances, in particular for those with a high number of capacity levels, i.e., q = 10. For the instances where CPLEX finds feasible solutions, the Lagrangian heuristic provides more reliable results, having a smaller maximum deviation of the provided solution value from the best known lower bound, while the computing times are significantly lower. Furthermore, given the strong lower bounds provided by the heuristic, the latter is capable to prove optimality for almost the same number of instances as CPLEX (218 vs. 227).

-		CPLEX					Bundle + R-MIP			
q	Instance	#	Avg	Max	Time	# 1%	Avg	Max	Time	# 1%
	size	ns	$\operatorname{Gap}\%$	Gap $\%$	(sec)	$_{\mathrm{opt}}$	Gap %	$\mathrm{Gap}~\%$	(sec)	$_{\rm opt}$
3	50/50	0	0.23	1.21	4,327.7	16	0.31	1.22	4,142.9	9
	50/200	0	0.00	0.01	751.2	18	0.53	0.96	268.3	18
	100/100	0	0.01	0.18	1,325.3	18	0.40	0.87	273.2	18
	100/400	0	0.00	0.01	719.8	18	0.59	0.92	650.8	18
	150/150	0	0.02	0.26	1,665.1	17	0.29	0.99	610.8	18
	150/600	1	0.01	0.02	3,017.2	17	0.69	0.98	756.5	17
	200/200	0	0.12	2.04	2,949.2	17	0.42	0.94	1,098.2	18
	200/800	9	0.40	3.48	6,120.1	8	0.22	0.81	1,411.7	9
	250/250	0	0.10	1.79	2,978.1	17	0.40	0.93	1,381.2	18
	250/1000	15	0.84	2.51	8,152.7	2	0.05	0.11	1,400.7	3
	Avg All	<b>25</b>	0.10	3.48	2,553.1	148	0.43	1.22	$1,\!170.5$	146
5	50/50	3	2.15	8.39	8,005.7	9	1.19	3.28	5,831.7	2
	50/200	0	0.45	3.43	4,956.9	16	0.49	1.32	1,907.4	16
	100/100	4	0.36	1.48	4,955.4	11	0.42	1.29	2,104.2	10
	100/400	7	0.01	0.03	2,504.8	11	0.48	0.97	653.5	11
	150/150	7	0.92	7.56	6,093.3	8	0.34	0.89	812.4	10
	150/600	12	0.04	0.19	5,291.7	5	0.20	0.77	651.0	6
	200/200	10	0.09	0.39	4,031.4	8	0.16	0.50	800.8	8
	200/800	17	0.01	0.01	4,005.0	1	0.84	0.84	$1,\!145.0$	1
	250/250	12	0.04	0.14	5,347.5	5	0.18	0.73	557.7	6
	Avg All	90	0.63	8.39	5,259.5	<b>74</b>	0.50	3.28	1,950.8	70
10	50/50	12	1.78	4.07	10,800.2	3	1.85	2.81	611.5	0
	50/200	15	0.71	1.51	$10,\!110.0$	2	0.60	1.21	264.3	2
	100/100	13	2.88	8.82	10,800.4	0	1.92	3.14	2,254.6	0
	Avg All	166	1.94	8.82	$10,\!652.4$	<b>5</b>	1.61	3.14	$1,\!123.9$	<b>2</b>
All	Avg All	281	0.38	8.82	3,931.3	227	0.52	3.28	$1,\!433.1$	218

Table 4: Comparison of solution quality for CPLEX and the Lagrangian heuristic considering instances for the DFLP\_RPC with RUC constraints where CPLEX found feasible solutions.

## 6 Conclusions

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

In this paper, we proposed a new formulation for this problem, as well as for two simplified variants without relocation and with classical capacity constraints. For our test instances, the proposed formulations provide significantly lower integrality gaps than previous formulations, on average 13 to 30 times lower. As a consequence, MIP solvers perform much better when using the new formulation. Next to computational advantages, the proposed modeling technique also allows for a better representation of the cost structure of the problem.

Based on the proposed formulations, we developed efficient Lagrangian heuristics to find high quality solutions for the problems. They consist of two optimization phases. In the first phase, the Lagrangian dual is solved by a bundle method, providing lower and upper bounds. Then, a restricted MIP model is solved to improve the final solution quality. Even though the relocation of facilities, as well as the particular capacity constraints represent an additional obstacle when decomposing the problem, we demonstrate how to efficiently construct feasible facility relocations after relaxing the relocation linking constraints and how to solve the round-up capacity constraints in a combinatorial manner. Computational results show that the proposed heuristics outperform state-of-the-art MIP solvers, providing better average and maximum deviations from the best known lower bounds in significantly shorter computing times. While the MIP solver does not find feasible solutions for about half of the instances, the heuristics are able to provide high quality solutions for all instances. The average deviation from the best known lower bounds for all 540 test instances is 1.37% for the original problem considering relocation and round-up capacity constraints. The Lagrangian

heuristics also provide relatively accurate bounds on the solution quality, since the proposed formulations produce low integrality gaps.

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

# References

- Antunes, A. P., D. Peeters. 2001. On solving complex multi-period location models using simulated annealing. European Journal of Operational Research 130(1) 190–201.
- Ballou, R. H. 1968. Dynamic Warehouse Location Analysis. Journal of Marketing Research 5(3) 271–276.
- 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.
- 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., A. Sutter, M.-C. Costa. 1996. Solving the dynamic facility location problem. *Networks* 28(2) 117–124.
- Correia, I., M. E. Captivo. 2003. A Lagrangean heuristic for a modular capacitated location problem. Annals of Operations Research 122 141–161.
- 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.
- 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., Publication CRT-945, Centre de recherche sur les transports, Université de Montréal.
- Geoffrion, A. M. 1974. Lagrangean relaxation for integer programming. *Mathematical Programming Study* 2 **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.
- Gouveia, L., F. Saldanha da Gama. 2006. On the capacitated concentrator location problem: a reformulation by discretization. *Computers & Operations Research* **33**(5) 1242–1258.
- 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. 2014a. Dynamic Facility Location with Generalized Modular Capacities. Transportation Science doi:http://dx.doi.org/10.1287/trsc.2014.0575.
- Jena, S. D., J.-F. Cordeau, B. Gendron. 2014b. Lagrangian Heuristics for Large-Scale Dynamic Facility Location with Generalized Modular Capacities. Tech. rep., CIRRELT-2014-21, Centre interuniversitaire de recherche sur les reseaux 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, C. Y. 1991. An optimal algorithm for the multiproduct capacitated facility location problem with a choice of facility type. Computers & Operations Research 18(2) 167–182.
- 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.
- Luss, H. 1982. Operations Research and Capacity Expansion Problems: A Survey. *Operations Research* **30**(5) 907–947.
- Martello, S., P. Toth. 1990. *Knapsack problems: algorithms and computer implementations*. Interscience Series in Discrete Mathematics and Optimization, John Wiley & Sons.
- 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. 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.
- Min, H., E. Melachrinoudis. 1999. The relocation of a hybrid manufacturing/distribution facility from supply chain perspectives: a case study. *Omega* **27**(1) 75–85.
- Schütz, P., L. Stougie, A. Tomasgard. 2008. Stochastic facility location with general long-run costs and convex short-run costs. Computers & Operations Research 35(9) 2988–3000.
- Shulman, A. 1991. An Algorithm for Solving Dynamic Capacitated Plant Location Problems with Discrete Expansion Sizes. Operations Research 39(3) 423–436.
- Snyder, L. V. 2006. Facility location under uncertainty: a review. IIE Transactions 38(7) 547-564.
- Vahidnia, M. H., A. A. Alesheikh, A. Alimohammadi. 2009. Hospital site selection using fuzzy AHP and its derivatives. *Journal of environmental management* **90**(10) 3048–56.
- Van Roy, T. J. 1986. A Cross Decomposition Algorithm for Capacitated Facility Location. Operations Research 34(1) 145–163.
- Van Roy, T. J., D. Erlenkotter. 1982. A Dual-Based Procedure for Dynamic Facility Location. Management Science 28(10) 1091–1105.
- Warszawski, A. 1973. Multi-Dimensional Location Problems. Operational Research Quarterly (1970-1977) 24(2) 165–179.
- Wentges, P. 1996. Accelerating Benders' decomposition for the capacitated facility location problem. Mathematical Methods of Operations Research 44(2) 267–290.
- Wesolowsky, G. O. 1973. Dynamic facility location. Management Science 19(11) 1241–1248.
- Wesolowsky, G. O., W. G. Truscott. 1975. The Multiperiod Location-Allocation Problem with Relocation of Facilities. *Management Science* **22**(1) 57–65.
- Wu, L., X. Zhang, J. Zhang. 2006. Capacitated facility location problem with general setup cost. Computers & Operations Research 33(5) 1226–1241.

# A Test Instances

Due to the lack of openly available instance sets that include the input data required by the DFLP\_PC and DFLP\_RPC with classical and with RUC constraints, we generated a total of 540 instances, 180 for each capacity level, to test the presented models. These instances extend those used by Jena et al. (2014b) to account for partial facility closing, facility relocation and the RUC constraints. In the following we briefly describe how these instance properties are generated and which parameters are used. We refer to Jena et al. (2014b) for details.

## 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). Instances have been 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 candidate facility locations and the customer demand points have been randomly generated following a continuous uniform distribution. The networks were generated on squares of the following three sizes: 300km, 380km and 450km. 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.2 Demand allocation and fixed costs

Costs are divided into fixed and variable costs and include economies of scale in function of the size of the facility. 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 generation of the total demand allocation costs  $g_{ij\ell t}$  to serve the customer demands, the facility construction costs  $f_{j\ell}^o$  and the facility maintenance costs  $F_{i\ell}^o$  is described in Jena et al. (2014b).

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 1,..., 10 closed capacity levels 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 1,..., 10 open capacity levels 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.

Once all capacity levels are closed, a facility can be relocated to another location. The costs to relocate a facility of  $1, \ldots, 10$  capacity levels from one location to another are set to 12,823.70, 19,431.90, 24,077.00, 30,247.00, 42,639.10, 48,854.50, 50,579.90, 56,314.10, 67,804.10 and 73,558.50, respectively.

## A.3 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, 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 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. The demands for the second to fifth commodity are computed based on the demand of the first commodity. For further details, we refer to Jena et al. (2014b). When round-up capacity constraints are used, demands  $d_{ipt}$  are divided by the corresponding lot size  $s_p$ , which have been set to  $s_1 = 6$ ,  $s_2 = 3$ ,  $s_3 = 5$ ,  $s_4 = 2$  and  $s_5 = 7$ .