

Sauvey, Christophe; Melo, Teresa; Correia, Isabel

Research Report

Two-phase heuristics for a multi-period capacitated facility location problem with service-differentiated customers

Schriftenreihe Logistik der Fakultät für Wirtschaftswissenschaften der htw saar, No. 16

Provided in Cooperation with:

Saarland University of Applied Sciences (htw saar), Saarland Business School

Suggested Citation: Sauvey, Christophe; Melo, Teresa; Correia, Isabel (2019) : Two-phase heuristics for a multi-period capacitated facility location problem with service-differentiated customers, Schriftenreihe Logistik der Fakultät für Wirtschaftswissenschaften der htw saar, No. 16, Hochschule für Technik und Wirtschaft des Saarlandes, Fakultät für Wirtschaftswissenschaften, Saarbrücken

This Version is available at:

<https://hdl.handle.net/10419/195071>

Standard-Nutzungsbedingungen:

Die Dokumente auf EconStor dürfen zu eigenen wissenschaftlichen Zwecken und zum Privatgebrauch gespeichert und kopiert werden.

Sie dürfen die Dokumente nicht für öffentliche oder kommerzielle Zwecke vervielfältigen, öffentlich ausstellen, öffentlich zugänglich machen, vertreiben oder anderweitig nutzen.

Sofern die Verfasser die Dokumente unter Open-Content-Lizenzen (insbesondere CC-Lizenzen) zur Verfügung gestellt haben sollten, gelten abweichend von diesen Nutzungsbedingungen die in der dort genannten Lizenz gewährten Nutzungsrechte.

Terms of use:

Documents in EconStor may be saved and copied for your personal and scholarly purposes.

You are not to copy documents for public or commercial purposes, to exhibit the documents publicly, to make them publicly available on the internet, or to distribute or otherwise use the documents in public.

If the documents have been made available under an Open Content Licence (especially Creative Commons Licences), you may exercise further usage rights as specified in the indicated licence.

C. Sauvey | T. Melo | I. Correia

Two-phase heuristics for a multi-period capacitated facility
location problem with service-differentiated customers

**Schriftenreihe Logistik der Fakultät für Wirtschaftswissenschaften
der htw saar**

Technical reports on Logistics of the Saarland Business School

Nr. 16 (2019)

Logistik

© 2019 by Hochschule für Technik und Wirtschaft des Saarlandes, Fakultät für Wirtschaftswissenschaften,
Saarland Business School

ISSN 2193-7761

Two-phase heuristics for a multi-period capacitated facility location problem with service-differentiated customers

C. Sauvey | T. Melo | I. Correia

Bericht/Technical Report 16 (2019)

Verantwortlich für den Inhalt der Beiträge sind die jeweils genannten Autoren.

Alle Rechte vorbehalten. Ohne ausdrückliche schriftliche Genehmigung des Herausgebers darf der Bericht oder Teile davon nicht in irgendeiner Form – durch Fotokopie, Mikrofilm oder andere Verfahren – reproduziert werden. Die Rechte der öffentlichen Wiedergabe durch Vortrag oder ähnliche Wege bleiben ebenfalls vorbehalten.

Die Veröffentlichungen in der Berichtreihe der Fakultät für Wirtschaftswissenschaften können bezogen werden über:

Hochschule für Technik und Wirtschaft des Saarlandes
Fakultät für Wirtschaftswissenschaften
Campus Rotenbühl
Waldhausweg 14
D-66123 Saarbrücken

Telefon: +49 (0) 681/ 5867 - 519

E-Mail: fakultaet-wiwi@htwsaar.de

Internet: www.htwsaar.de/wiwi

Two-phase heuristics for a multi-period capacitated facility location problem with service-differentiated customers

Christophe Sauvey^a, Teresa Melo^{b*}, Isabel Correia^c

^a *Université de Lorraine, LGIPM, F-57000 Metz, France*

^b *Business School, Saarland University of Applied Sciences, D 66123 Saarbrücken, Germany*

^c *Centro de Matemática e Aplicações, Departamento de Matemática, Faculdade de Ciências e Tecnologia, Universidade Nova de Lisboa, P 2829-516 Caparica, Portugal*

Abstract

We investigate a recently introduced extension of the multi-period facility location problem that considers service-differentiated customer segments. Accordingly, some customers require their demands to be met on time, whereas the remaining customers accept delayed deliveries as long as lateness does not exceed a pre-specified threshold. In this case, late shipments can occur at most once over the delivery lead time, i.e. an order cannot be split over several time periods. At the beginning of the multi-period planning horizon, a number of facilities are in place with given capacities. A finite set of potential facility sites with multiple capacity levels is also available. The objective is to find the optimal locations and the opening, resp. closing, schedule for new, resp. existing, facilities that provide sufficient capacity to satisfy all customer demands at minimum cost. In this paper, we propose four heuristics that construct initial solutions to this problem and subsequently explore their neighborhoods via different local improvement mechanisms. Computational results with randomly generated instances demonstrate the effectiveness of the proposed heuristics. While a general-purpose mixed-integer programming solver fails to find feasible solutions to some instances within a given time limit, the heuristics provide good solutions to all instances already during the constructive phase and in significantly shorter computing times. During the improvement phase, the solution quality is further enhanced. For nearly one-fifth of the instances, the heuristic solutions outperform the best solutions identified by the solver.

Keywords: facility location, multi-period, delivery lateness, constructive heuristics, local improvements

*Corresponding author. *E-mail address:* teresa.melo@htwsaar.de

1 Introduction

Discrete facility location problems involve a finite set of potential sites at which new facilities can be located, and a finite set of customers, whose demands have to be satisfied from the new facilities. In solving these problems, a balance between fixed facility costs and variable transportation costs must be found. Stimulated by real-world problems, a wide variety of extensions to classical discrete location models have emerged in different contexts. The latter include, but are not limited to, supply chain network design (Melo et al., 2009), telecommunications (Fortz, 2015), location-routing (Prodhon and Prins, 2014), health care services (Ahmadi-Javid et al., 2017), and humanitarian operations (Martins et al., 2018). One important problem extension focuses on the time-phasing of location and demand allocation decisions over a multi-period finite planning horizon (Nickel and Saldanha da Gama, 2015).

In this paper, we consider a multi-period facility location problem with different time scales for strategic (location) decisions and tactical (demand allocation) decisions that was recently introduced by Correia and Melo (2016). Given a number of facilities that are available at the beginning of the planning horizon and a set of potential sites for new facilities, the problem involves determining the optimal locations and the opening, resp. closing, schedule for new, resp. existing, facilities in order to satisfy time-varying customer demands at minimum cost. Facility sizing decisions are also considered by offering a choice of discrete capacity levels in each potential location. While multi-period facility location has been a research topic of recurring interest (Owen and Daskin, 1998; Klose and Drexl, 2005; Arabani and Farahani, 2012), only recently a few authors have acknowledged the relevance of adopting different time scales for decisions belonging to distinct planning levels. This is the case of the location-routing problem addressed by Albareda-Sambola et al. (2012), in which strategic location decisions are allowed to be made over a pre-defined subset of periods in the time horizon, whereas routing decisions can be made in any time period. A similar approach is also followed by Bashiri et al. (2012) and Badri et al. (2013) in the context of supply chain network design. Moreover, in the multi-period location problem studied by Fattahi et al. (2015), each strategic period spans multiple, consecutive tactical periods. Some authors also integrate location and capacity acquisition decisions into a single model (Amiri, 2006; Elhedhli and Gzara, 2008; Amrani et al., 2011; Correia et al., 2013; Irawan and Jones, 2019) by assuming that multiple capacity levels are available at each potential location. This setting is appropriate when capacity is purchased in the form of equipment which is available at a few discrete sizes (Correia and Captivo, 2006).

While most multi-period location models enforce the satisfaction of customer demands on time, the problem introduced by Correia and Melo (2016) is the first to consider service-differentiated customer segments. To this end, two customer segments are modeled with distinct sensitivity with respect to delivery lead times. Customers in the first segment require their demands to be met in the same periods in which they occur, whereas customers in the second segment tolerate late deliveries as long as they do not exceed a pre-specified threshold. This setting is often encountered in inventory management for spare parts (Alvarez et al., 2015), repair services (Wang et al., 2002), and retail companies (Duran et al., 2008; Li et al., 2015). Typically, two demand classes are considered: customers in the first class receive priority service, while customers in the second class are given price incentives in exchange for longer response times (Hung et al., 2012; Wu and Wu, 2015). By offering different price and delivery time options to customers, companies such as Amazon remain competitive with other available services (e.g. brick-and-mortar stores), are able to segment the available market in an attempt to earn more profit from customers demanding preferred service, and can benefit from greater flexibility in managing their facility network. All these aspects impact location decisions as suggested by Escalona et al. (2015) for a location-inventory problem.

When those customers accepting late shipments do not allow an order to be split over multiple periods, thus requiring an order to be delivered as a single shipment, even if with some delay, Correia and Melo (2016) proposed two mixed-integer linear programming (MILP) formulations. Numerical experiments with a general-purpose MILP solver revealed that optimality could not be achieved within a time limit of 10 CPU hours for the majority of the instances. For a few instances, the solver even failed to find feasible solutions. These findings are not surprising due to the combinatorial nature of the problem at hand. In addition, when re-optimization is required for performing ‘what-if’ analysis and a model has to be solved repeatedly with different data sets, the computational burden can become prohibitively expensive. In this case, there is a need to develop tailored solution algorithms that can efficiently handle instances with significant size. Various exact and (meta)heuristic methods have been proposed to solve the discrete multi-period location problem as well as a number of challenging variants of the classical problem. Lagrangian relaxation (Shulman, 1991; Elhedhli and Gzara, 2008) and Benders decomposition (Torres-Soto and Üster, 2011; Marufuzzaman et al., 2016; Castro et al., 2017) have proved to be effective for some problems. Large instances are prone to be tackled by means of different kinds of heuristics, including Variable Neighborhood Search (Amrani et al., 2011), metaheuristics, (Arostegui Jr. et al., 2006; Melo et al., 2012), memetic algorithms (Dias

et al., 2008), and clustering techniques (Boujelben et al., 2016), among others.

In this paper, we make the following contributions to the problem studied by Correia and Melo (2016) and that we revisit: First, we propose several two-phase heuristic procedures for the problem. In the first phase, a feasible solution is constructed by choosing a subset of the initially existing facilities and a subset of the potential locations to operate over the planning horizon. Four alternative strategies are developed to explore the feasible space of the location decisions. For the facility choices made, customer demands are allocated according to a heuristic scheme. In the second phase, two different types of facility moves are applied which aim at finding improving configurations of the open facilities. The neighborhood structures are embedded in a local improvement algorithm which starts with a feasible solution and performs improving facility moves and customer demand reallocations until it obtains a locally optimal solution. The combination of the constructive and improving schemes yields in total 16 different possible ways to obtain feasible solutions to the problem at hand. Second, we assess the computational efficiency and the solution quality of the proposed heuristics for randomly generated instances. The testbed includes the instances created by Correia and Melo (2016) and is extended with larger instances. In contrast to CPLEX, all constructive heuristics are capable of finding feasible solutions to all instances in significantly shorter computing times. At the end of the two phases, the average heuristic objective value is less than 1% more costly than the objective value of the best solution identified by CPLEX. Moreover, for nearly one-fifth of the test instances, the heuristics even achieve higher solution quality than CPLEX.

The remainder of the paper is organized as follows. In Section 2, the problem is defined and one of the formulations proposed by Correia and Melo (2016) is revisited. Section 3 describes four constructive heuristics that provide initial feasible solutions. In Section 4, local improvement moves are presented. Section 5 reports the computational results obtained. Section 6 concludes the paper with a summary of our findings and an outline of opportunities for future research.

2 Formulation

For the problem revisited in this paper, Correia and Melo (2016) proposed two MILP formulations. For the sake of completeness, we present in this section one of the formulations, namely the formulation that exhibited a better computational performance in a set of numerical experiments. The heuristic procedures to be described in Sections 3 and 4 use the decision variables and various parameters of this model.

We consider a company that operates a set of facilities at fixed locations to serve the demands of customers (or customer zones) for a single product (or product family). The company differentiates its customers by their service requirements in terms of delivery times. Customers who receive preferred service have zero delivery lead time, i.e. their demands must be satisfied in the same periods in which they are placed. Customers who are not averse to waiting for their demands to be satisfied specify a maximum allowed delivery delay. These customers are compensated with a lower price which is translated into a tardiness penalty cost for delayed deliveries to reflect the negative impact on the company's profit margin. Due to projected variations in customer demands, it is anticipated that the company will not be able to provide adequate customer service in the future. Therefore, a finite set of potential locations at which new facilities can be located has been identified. At each potential site, a discrete set of capacity levels is also available. A planning horizon is considered which is divided into a finite number of time periods having equal durations. Selected time periods form the set of strategic periods at which existing facilities are closed, new facilities are opened and their capacity levels are installed. Shipment decisions from operating facilities to customers are made in any time period. The company needs to determine the number, location and capacity of new facilities, schedule their opening periods, and plan the removal of existing facilities so as to satisfy all customer demands at minimum cost. Due to the sizeable investment associated with location decisions, facilities cannot be temporarily closed and reopened. Accordingly, new facilities must remain in activity once they are initially opened. Analogously, if an existing facility is closed then it cannot be reopened in a later time period. All relevant data (i.e. costs, customer demands, and other parameters) are deterministic and assumed to be the outcome of forecasting methods and company-specific analyzes.

The following notation is used throughout the paper.

- T, T_L : Set of discrete time periods, resp. set of *strategic* time periods in which location and capacity acquisition decisions can be made, $T_L \subset T$
- I^e, I^n : Set of initially existing facilities, resp. set of potential locations for opening new facilities
- K_i : Set of capacity levels available in location i ($i \in I^e \cup I^n$); for each existing facility $i \in I^e$, $|K_i| = 1$
- J^0, J^1 : Set of customers that receive preferred service, resp. set of customers that tolerate delays in demand satisfaction

All facility locations are gathered in set I , i.e. $I = I^e \cup I^n$. Moreover, J denotes the set of all customers with $J = J^0 \cup J^1$ and $J^0 \cap J^1 = \emptyset$. Let $\ell = 1$, resp. ℓ_{max} , be the first, resp. last, strategic time period in which decisions on opening/closing facilities and installing capacity levels in potential locations can be made. The configuration of a facility in time period $\ell \in T_L$ is the outcome of the decisions taken until that period. For $\ell < \ell_{max}$, this configuration remains unchanged over all tactical periods between ℓ and the next strategic period ℓ' . For $\ell = \ell_{max}$, the configuration of the facility is maintained until the last period of the planning horizon. Let $\phi(\ell)$ denote the last tactical period between two consecutive strategic periods, ℓ and ℓ' . It follows that $\phi(\ell) = \max\{t \in T : t < \ell'\}$ for $\ell < \ell_{max}$, and $\phi(\ell) = |T|$ for $\ell = \ell_{max}$. Capacity and demand parameters are defined next.

- Q_{ik}, Q_{i1} : Capacity of level k that can be installed in potential location i ($i \in I^n; k \in K_i$), resp. capacity of initially existing facility i ($i \in I^e$)
- d_j^t : Demand of customer j for a given product at time period t ($j \in J; t \in T$)
- ρ_j : Maximum allowed delay (in number of time periods) to satisfy the demand of customer j ($j \in J$)

Customers that receive preferred service have $\rho_j = 0$ ($j \in J^0$), whereas customers tolerating late deliveries have $\rho_j > 0$ ($j \in J^1$). The time lag for demand satisfaction for customer j is also defined by ρ_j . This means that demand for period $t \in T$ must be filled over periods $t, t + 1, \dots, t + \rho_j$. In case $t + \rho_j > |T|$, then the last delivery must occur in period $|T|$, thus ensuring that demand is not carried over to future periods beyond the planning horizon. Moreover, we assume that for each customer $j \in J^1$, his demand d_j^t in period t cannot be split over multiple periods of time. In other words, the customer receives a single shipment even if it arrives with some delay. For the customer, the cost of handling a single shipment is proportionally less than the cost of processing several deliveries belonging to a particular order.

Fixed and variable costs are defined as follows:

- FO_{ik}^ℓ : Fixed cost of opening a new facility in potential location i with capacity level k at the *beginning* of time period ℓ ($i \in I^n; k \in K_i; \ell \in T_L$)
- FC_{i1}^ℓ : Fixed cost of closing the initially existing facility i at the *end* of time period ℓ ($i \in I^e; \ell \in T_L$)
- M_{ik}^t : Fixed cost of operating facility i with capacity level k in time period t ($i \in I; k \in K_i; t \in T$)

- c_{ij}^t : Cost of distributing one unit of product from facility i to customer j in time period t ($i \in I; j \in J; t \in T$)
- o_{ik}^t : Cost of processing one unit of product in facility i with capacity level k in time period t ($i \in I; k \in K_i; t \in T$)
- $p_j^{tt'}$: Tardiness penalty cost for satisfying one unit of demand of customer j in period t' that was originally demanded in period t ($j \in J^1; t \in T; t' = t, t+1, \dots, \min\{t + \rho_j, |T|\}$); in particular, $p_j^{tt'} = 0$ for $t' = t$

All cost parameters associated with the capacity levels ($FO_{ik}^\ell, FC_{i1}^\ell, M_{ik}^t, o_{ik}^t$) reflect economies of scale. In addition, economies of scale are also present in the variable cost for processing the product in a facility. By combining the fixed facility and operating costs over an appropriate number of time periods, we obtain the total fixed cost incurred over the planning horizon for opening or closing a facility in a given time period. For a new facility $i \in I^n$ that is opened in period $\ell \in T_L$ with capacity level $k \in K_i$, this total cost is calculated as $F_{ik}^\ell = FO_{ik}^\ell + \sum_{t=\ell}^{|T|} M_{ik}^t$. For an existing facility $i \in I^e$ that is closed at the end of period $\ell \in T_L$, the total fixed cost is given by $F_{i1}^\ell = FC_{i1}^\ell + \sum_{t=1}^{\ell} M_{i1}^t$.

The problem can be formulated as a MILP model using the following binary variables:

- z_{ik}^ℓ : 1 if a new facility is opened in potential location i with capacity level k at the beginning of time period ℓ , 0 otherwise ($i \in I^n; k \in K_i; \ell \in T_L$)
- z_{i1}^ℓ : 1 if the initially existing facility i is closed at the end of time period ℓ , 0 otherwise ($i \in I^e; \ell \in T_L$)
- $v_j^{tt'}$: 1 if all the demand of customer j for period t is delivered in period t' , 0 otherwise ($j \in J^1; t \in T; t' = t, \dots, \min\{t + \rho_j, |T|\}$)

Additionally, we use three sets of continuous variables that are related to the amount of product that is moved from operating facilities to customers.

- r_{ij}^t : Total quantity of product shipped from facility i to customer j in time period t ($i \in I; j \in J^0; t \in T$)
- $s_{ij}^{tt'}$: Amount of product distributed from facility i to customer j in time period t' to satisfy all demand of period t ($i \in I; j \in J^1; t \in T; t' = t, \dots, \min\{t + \rho_j, |T|\}$)
- w_{ik}^t : Total quantity of product shipped from facility i with capacity level k in time period t ($i \in I; k \in K_i; t \in T$)

The MILP formulation is as follows:

$$\begin{aligned}
\text{Min} \quad & \sum_{\ell \in T_L} \sum_{i \in I} \sum_{k \in K_i} F_{ik}^\ell z_{ik}^\ell + \sum_{t \in T} \sum_{i \in I^e} M_{i1}^t \left(1 - \sum_{\ell \in T_L} z_{i1}^\ell \right) \\
& + \sum_{t \in T} \sum_{i \in I} \sum_{j \in J^0} c_{ij}^t r_{ij}^t + \sum_{t \in T} \sum_{i \in I} \sum_{k \in K_i} o_{ik}^t w_{ik}^t \\
& + \sum_{t \in T} \sum_{i \in I} \sum_{j \in J^1} \sum_{t'=t}^{\min\{t+\rho_j, |T|\}} \left(p_j^{tt'} + c_{ij}^{t'} \right) s_{ij}^{tt'} \tag{1}
\end{aligned}$$

s.t.

$$\sum_{\ell \in T_L} \sum_{k \in K_i} z_{ik}^\ell \leq 1 \quad i \in I \tag{2}$$

$$\sum_{i \in I} r_{ij}^t = d_j^t \quad j \in J^0, t \in T \tag{3}$$

$$\sum_{i \in I} s_{ij}^{tt'} = d_j^t v_j^{tt'} \quad j \in J^1, t \in T, \tag{4}$$

$$\sum_{t'=t}^{\min\{t+\rho_j, |T|\}} v_j^{tt'} = 1 \quad j \in J^1, t \in T \tag{5}$$

$$w_{ik}^t \leq Q_{ik} \sum_{\ell \in T_L: \ell \leq t} z_{ik}^\ell \quad i \in I^n, k \in K_i, t \in T \tag{6}$$

$$w_{i1}^t \leq Q_{i1} \left(1 - \sum_{\ell \in T_L: \ell < t} z_{i1}^\ell \right) \quad i \in I^e, t \in T \tag{7}$$

$$\sum_{k \in K_i} w_{ik}^t = \sum_{j \in J^0} r_{ij}^t + \sum_{j \in J^1} \sum_{t'=\max\{1, t-\rho_j\}}^t s_{ij}^{t't} \quad i \in I, t \in T \tag{8}$$

$$z_{ik}^\ell \in \{0, 1\} \quad i \in I, k \in K_i, \ell \in T_L \tag{9}$$

$$v_j^{tt'} \in \{0, 1\} \quad j \in J^1, t \in T, \tag{10}$$

$$r_{ij}^t \geq 0 \quad i \in I, j \in J^0, t \in T \tag{11}$$

$$s_{ij}^{tt'} \geq 0 \quad i \in I, j \in J^1, t \in T, \tag{12}$$

$$w_{ik}^t \geq 0 \quad i \in I, k \in K_i, t \in T \tag{13}$$

The objective function (1) minimizes the total sum of the fixed and variable costs. The former include the costs incurred for opening new facilities and installing capacity levels in the new sites, removing initially existing facilities, and operating facilities in those periods in which

they are active. Variable costs account for processing the product in operating facilities and shipping it to customers. In addition, tardiness costs resulting from delayed deliveries are also incurred. For each potential location $i \in I^n$, constraints (2) state that at most one new facility can be opened with a given capacity level over the time horizon. Constraints (2) also allow each initially existing facility $i \in I^e$ to be closed at most once throughout the planning horizon. Constraints (3), resp. (4), guarantee the satisfaction of the demand over the time horizon for customer segment J^0 , resp. J^1 . Equalities (5) ensure that an order of customer $j \in J^1$ cannot be split over multiple time periods. Inequalities (6), resp. (7), are capacity constraints for new, resp. existing, facilities. Observe that since an existing facility can only be closed at the end of a given time period, say ℓ , its capacity is not available in any subsequent period. This is described in (7) by considering all strategic periods $\ell \in T_L$ such that $\ell < t$ for every $t \in T$. In contrast, if a new facility is opened in time period t then its capacity also becomes available in the same period. Therefore, in constraints (6) we consider all periods $\ell \in T_L$ such that $\ell \leq t$ for every $t \in T$. Constraints (8) state that the total product outflow from a facility in a given time period is split into deliveries to customers with preferred service and deliveries to customers accepting delays in demand satisfaction. Finally, non-negativity and binary conditions are given by (9)–(13).

Correia and Melo (2016) also enhanced the above formulation with additional inequalities that set a lower bound on the total number of facilities that must be available in each time period of the planning horizon. Finally, we remark that the problem belongs to the class of NP-hard problems as it generalizes the classical multi-period uncapacitated facility location problem (Jacobsen, 1990).

3 Constructive heuristics

The general framework of the proposed constructive schemes is provided in Algorithm 1. We start by selecting in the first strategic period the subset of facilities that should be available over the whole planning horizon (line 3). To this end, the set \hat{I}^e is identified which includes the existing facilities that are retained from time period 1 until period $|T|$. Analogously, the set \hat{I}^n is also specified with all new facilities that are opened in the first time period. We have developed four different procedures to obtain \hat{I}^e and \hat{I}^n . These will be detailed in Section 3.2. The location choices gathered in \hat{I}^e and \hat{I}^n can be later revoked during the improvement phase (cf. Section 4). After having selected the operating facilities, the allocation of customer demands is

then performed (line 7) over all time periods according to a mechanism that will be described in Section 3.1. Depending on the choice of facilities to operate, this step may fail to detect a feasible solution. Therefore, we have also developed additional strategies that enable us to revise our previous location decisions (see Section 3.2). A pre-specified stopping criterion rules the number of times this procedure is repeated. In lines 4 and 5, we initialize sets \widehat{I}_t^e , resp. \widehat{I}_t^n , with all existing, resp. new, facilities available at time period t ($t \in T$). Since an existing facility can only be closed at the end of a strategic period (recall the definition of variables z_{i1}^ℓ in Section 2), all existing facilities are available at time period $t = 1$ and therefore, $\widehat{I}_1^e = I^e$ (line 4). If $\widehat{I}^e \subset I^e$ (i.e. some, but not all, existing facilities have been selected) then in any other period t ($2 \leq t \leq |T|$) none of the facilities in $I^e \setminus \widehat{I}^e$ are operating. For the selected new facilities, it is not necessary to distinguish between $t = 1$ and $t > 1$ (line 5) because the time instant for opening a new facility coincides with the beginning of the period in which this action takes place.

Algorithm 1: General constructive scheme

Input : All instance data
Output: Solution \mathcal{S}

```

1 while stopping condition is not satisfied do
2   // Choice of facilities to operate throughout the planning horizon
3   SelectFacilities( $\widehat{I}^e, \widehat{I}^n$ )
4    $\widehat{I}_1^e := I^e$  and  $\widehat{I}_t^e := \widehat{I}^e$  for every  $t = 2, \dots, |T|$ 
5    $\widehat{I}_t^n := \widehat{I}^n$  for all  $t \in T$ 
6   // Allocation of customer demands to selected facilities
7   AllocateDemands( $\widehat{I}_1^e, \dots, \widehat{I}_{|T|}^e, \widehat{I}_1^n, \dots, \widehat{I}_{|T|}^n$ )
8 end
9 return feasible solution  $\mathcal{S} = (z, v, r, s, w)$ 

```

3.1 Allocation of customer demands

Given pre-selected sets of open facilities, \widehat{I}_t^e and \widehat{I}_t^n ($t \in T$), customer demands are satisfied over the planning horizon according to Procedure AllocateDemands. At each time period t , this process consists of two phases. In the first phase, so-called *priority demands*, gathered in set D_*^t , are satisfied (line 5). The set D_*^t includes all demand requirements of customer segment J^0 for period t and unsatisfied orders of customers in segment J^1 whose fulfillment cannot be further delayed. If at the end of this phase there is at least one open facility with positive residual

capacity (line 7) then additional demands from customer segment J^1 are allocated in the second phase (line 8). To keep track of the capacity that is available in each open facility in period t , the residual capacity vector $\mathbf{RQ}^t = (RQ_i^t, i \in \widehat{I}_t^e \cup \widehat{I}_t^n)$ is updated whenever the demand of a customer is partially or totally assigned to an operating facility. The sets U^t ($t \in T$), which appear in lines 6 and 8, will be defined below. Procedure AllocateDemands is also embedded in the local improvement heuristics to be described in Section 4.

Algorithm AllocateDemands: Demand allocation over the planning horizon

Input : $\widehat{I}_1^e, \dots, \widehat{I}_{|T|}^e$: existing facilities that are operated
 $\widehat{I}_1^n, \dots, \widehat{I}_{|T|}^n$: new facilities that are open
Output: Solution \mathcal{S}

- 1 **for** $t := 1$ **to** $|T|$ **do**
- 2 Determine total capacity available in period t , TQ^t
- 3 Determine set of *priority demands*, D_*^t , and the associated total demand, d_{min}^t
- 4 **if** $d_{min}^t \leq TQ^t$ **then**
- 5 AllocatePriorityDemand($\widehat{I}_t^e, \widehat{I}_t^n, D_*^t, \mathbf{RQ}^t$)
- 6 $U^t := J^1$
- 7 **if** $\sum_{i \in \widehat{I}_t^e \cup \widehat{I}_t^n} RQ_i^t > 0$ **and** $t < |T|$ **then**
- 8 AllocateNonPriorityDemand($\widehat{I}_t^e, \widehat{I}_t^n, \mathbf{RQ}^t, U^{t-\rho+1}, \dots, U^{t-1}, U^t$)
- 9 **end**
- 10 **else**
- 11 **return** \emptyset // unable to construct feasible solution
- 12 **end**
- 13 **end**
- 14 **return** solution $\mathcal{S} = (z, v, r, s, w)$ (if feasible)

In what follows, we assume without loss of generality that all customers in segment J^1 have the same maximum delay ρ , that is, $\rho_j = \rho$ for all $j \in J^1$ ($\rho > 0$). The procedures that detail the demand allocation can be easily adapted when this is not the case.

At the beginning of each time period t , there are various demand sources. Some of them must be served in period t , while others may or may not be considered for service in that period. The former group concerns two types of customers: all customers from the preferred segment, J^0 , and a subset of the customers of segment J^1 whose demands for period $t - \rho$ were not satisfied so far. Failing to serve the demands of these customers in period t would violate the maximum delay tolerated by them. This specific subset of customers is denoted by $U^{t-\rho}$. In addition, optional service can also be provided in period t to customers from J^1 with unfilled demands from periods $t - \rho + 1, \dots, t - 1$. These particular customers are gathered in the sets

$U^{t-\rho+1}, \dots, U^{t-1}$. Note that demands from J^1 for the current period t may or may not be satisfied in t . For consistency, we introduce $U^t = J^1$. All demands that cannot be delayed are called *priority demands* and form the set D_*^t . Accordingly, this set is defined as follows:

$$\begin{aligned} D_*^t &= \{d_j^t : j \in J^0\} && \text{if } t = 1, \dots, \rho \\ D_*^t &= \{d_j^t : j \in J^0\} \cup \{d_j^{t-\rho} : j \in U^{t-\rho}\} && \text{if } t = \rho + 1, \dots, |T| - 1 \\ D_*^t &= \{d_j^t : j \in J\} \cup \{d_j^{t-\rho} : j \in U^{t-\rho}\} \cup \\ &\quad \{d_j^{t-\rho+1} : j \in U^{t-\rho+1}\} \cup \dots \cup \{d_j^{t-1} : j \in U^{t-1}\} && \text{if } t = |T| \end{aligned}$$

Observe that in the last time period it is necessary to include the demand from both customer segments for that period as well as all unsatisfied demand from periods $|T| - \rho, \dots, |T| - 1$. In particular, customers $j \in U^{|T|-\rho+1} \cup \dots \cup U^{|T|-1}$ must also be given priority in the last period since they cannot be serviced after the end of the planning horizon.

Priority demands set the minimum service level that must be provided in each time period. Let d_{min}^t denote the minimum demand requirements in period $t \in T$. Clearly, d_{min}^t is given by:

$$d_{min}^t = \begin{cases} \sum_{j \in J^0} d_j^t & \text{if } t = 1, \dots, \rho \\ \sum_{j \in J^0} d_j^t + \sum_{j \in U^{t-\rho}} d_j^t & \text{if } t = \rho + 1, \dots, |T| - 1 \\ \sum_{j \in J} d_j^t + \sum_{t'=t-\rho}^{|T|-1} \sum_{j \in U^{t'}} d_j^{t'} & \text{if } t = |T| \end{cases}$$

Procedure `AllocatePriorityDemand` outlines how priority demands are allocated to facilities operating in time period t . The total capacity available in this period is determined by $TQ^t = \sum_{i \in \widehat{I}_t^e} Q_{i1} + \sum_{i \in \widehat{I}_t^n} Q_{ik_i^*}$, with k_i^* denoting the capacity level chosen for facility $i \in \widehat{I}_t^n$. Since Procedure `AllocatePriorityDemand` is only run when inequality $d_{min}^t \leq TQ^t$ holds (line 4, Procedure `AllocateDemands`), all priority demands can be satisfied. We start by initializing the residual capacity RQ_i^t of every facility $i \in \widehat{I}_t^e \cup \widehat{I}_t^n$ with the corresponding capacity size (line 2, Procedure `AllocatePriorityDemand`). After ranking the priority demands from highest to lowest, we decide on the facility or facilities that should serve each one of them (lines 7–23). To this end, we consider all facilities with positive residual capacity and sort them by non-decreasing total variable cost for serving the customer specified in line 4. This cost is the sum of the processing cost (which depends on the size of the facility) and the distribution cost. We give preference to choosing the least costly facility. If its currently free capacity is insufficient to

cover the customer order then demand will have to be split between two or more facilities (lines 16-20).

Algorithm AllocatePriorityDemand: Allocation of *priority demands* to facilities operating in time period t

Input : $\widehat{I}_t^e, \widehat{I}_t^n$: existing facilities, resp. new, facilities operating in period t
 D_*^t : demands to be served in period t , sorted by non-increasing value

Output: Complete allocation of demands from D_*^t , update of residual capacity vector,
 RQ^t

```

1 // Initialize the residual capacity of each facility with the corresponding
  available capacity level
2  $RQ_i^t := Q_{i1}$  for every  $i \in \widehat{I}_t^e$  and  $RQ_i^t := Q_{ik_i^*}$  for every  $i \in \widehat{I}_t^n$ 
3 repeat
4   Extract a demand from the top of  $D_*^t$ . Let  $d_j^{t'}$  be this demand
5    $rd_j^t := d_j^{t'}$  // (residual) demand of customer  $j$ 
6   Create list  $\widehat{I}_t$  of facilities  $\widehat{I}_t^e \cup \widehat{I}_t^n$  that have positive residual capacity; sort these
   facilities by non-decreasing cost  $o_{ik}^t + c_{ij}^t$  (with  $k = 1$  if  $i \in \widehat{I}_t^e$  and  $k = k_i^*$  if  $i \in \widehat{I}_t^n$ ) and
   set  $w_{ik}^t := 0$ 
7   repeat
8     Extract a facility from the top of  $\widehat{I}_t$ , say  $i$ 
9     if  $rd_j^t \leq RQ_i^t$  then
10      // Assign residual demand of customer  $j$  to facility  $i$ 
11      if  $j \in J^0$  then  $r_{ij}^t := rd_j^t$  else  $s_{ij}^{t't} := rd_j^t$ 
12       $RQ_i^t := RQ_i^t - rd_j^t$  // update residual capacity
13       $w_{ik}^t := w_{ik}^t + rd_j^t$ 
14       $rd_j^t := 0$ 
15    else
16      // Assign part of residual demand of customer  $j$  to facility  $i$ 
17      if  $j \in J^0$  then  $r_{ij}^t := RQ_i^t$  else  $s_{ij}^{t't} := RQ_i^t$ 
18       $w_{ik}^t := w_{ik}^t + RQ_i^t$ 
19       $rd_j^t := rd_j^t - RQ_i^t$  // update residual demand of customer  $j$ 
20       $RQ_i^t := 0$ 
21    end
22    if  $j \in J^1$  then  $v_j^{t't} := 1$ 
23  until  $rd_j^t = 0$ 
24 until  $D_*^t = \emptyset$ 
25 if  $t > \rho$  then  $U^{t-\rho} := \emptyset$ 
26 if  $t = |T|$  then  $U^\tau := \emptyset$  for every  $\tau = |T| - \rho + 1, \dots, |T|$ 
27 return  $v, r, s, w, RQ^t$ 

```

The demand allocation decisions are also translated into specific values for variables r_{ij}^t

($i \in \widehat{I}_t^c \cup \widehat{I}_t^n$; $j \in J^0$), $s_{ij}^{t't}$ and $v_j^{t't}$ ($i \in \widehat{I}_t^c \cup \widehat{I}_t^n$; $j \in J^1$), and w_{ik}^t ($i \in \widehat{I}_t^c \cup \widehat{I}_t^n$; $k \in K_i$) in lines 11, 13, 17, 18, and 22. The time period t' in line 4 of the algorithm identifies the period in which a particular priority demand occurs. For every customer belonging to segment J^0 , $t' = t$. If $j \in J^1$ then t' may be associated with different time periods. In this case, if $\rho < t < |T|$ then t' represents period $t - \rho$ according to the definition of the set D_*^t . Only in the last time period, t' is associated with multiple periods with unsatisfied demand for customers in J^1 , namely periods $t - \rho, t - \rho + 1, \dots, |T|$ (recall the definition of $D_*^{|T|}$).

Throughout Procedure AllocatePriorityDemand, the capacity available of each facility open in time period t is gradually used according to the demand allocation decisions that are made. After having satisfied all priority demands, the second phase of the demand allocation scheme is applied, provided that there is at least one facility with positive residual capacity (recall line 7 in Procedure AllocateDemands). In this case, Procedure AllocateNonPriorityDemand manages the unfilled demands of all customers gathered in sets $U^{t-\rho+1}, \dots, U^t$. Since the longer a customer order is delayed, the higher the tardiness penalty cost becomes, it would seem natural to rank these customers by giving the highest priority to set $U^{t-\rho+1}$ and the lowest priority to set U^t . Therefore, the demands of customers belonging to $U^{t-\rho+1}$ would be the first to be allocated, whereas the demands of customers in U^t would be the last to be considered, provided that sufficient residual capacity were available. However, depending on the quantities ordered by individual customers, demands would not necessarily be satisfied in the less costly manner using this strategy. This case is illustrated in Table 1 by means of a small example with two customers and assuming the maximum delay $\rho = 3$. Let us also suppose that the total residual capacity is 15. Customer j requests 6 units in period $t - 2$, while customer j' orders 10 units in period $t - 1$. The satisfaction of the demand of customer j in period t incurs a total tardiness cost of 18 monetary units and the residual capacity drops to 9. Clearly, there is not sufficient capacity to also meet the order of customer j' in period t , and so its demand will have to be further delayed. This will increase even more the tardiness cost of this customer (from 20 to

Customer	Demand	Unit tardiness cost		Total tardiness cost	
		period t	period $t + 1$	period t	period $t + 1$
$j \in U^{t-2}$	$d_j^{t-2} = 6$	$p_j^{t-2,t} = 3$	$p_j^{t-2,t+1} = 4$	18	24
$j' \in U^{t-1}$	$d_{j'}^{t-1} = 10$	$p_{j'}^{t-1,t} = 2$	$p_{j'}^{t-1,t+1} = 3$	20	30

Table 1: Example with two customers and their tardiness costs in periods t and $t + 1$

30). In fact, it would be less costly to meet the 10 units of customer j' in period t and postpone servicing customer j to period $t + 1$.

Hence, to avoid additional tardiness costs, we adopt an improved procedure for the allocation of unfilled demands which is based on the total tardiness penalty cost $(p_j^{t't} + 1) d_j^{t'}$ for all $j \in U^{t'}$ and $t' = t - \rho + 1, \dots, t$. We add one monetary unit to the original tardiness penalty cost $p_j^{t't}$ because no delay occurs when $t' = t$ (i.e. $p_j^{tt} = 0$). Without this perturbation, we would never consider allocating demand for customers in U^t since $p_j^{tt} d_j^t = 0$, and so these customers would always experience delays. Next, we create the list D of demands sorted by non-increasing values of $(p_j^{t't} + 1) d_j^{t'}$ (line 1, Procedure AllocateNonPriorityDemand). In this way, higher importance is given both to large orders subject to a relatively low unit tardiness penalty cost and to small demands incurring a high unit tardiness penalty cost. Lines 2–27 describe a procedure similar to the one used in algorithm AllocatePriorityDemand to satisfy each one of these demands. If there is enough residual capacity to serve a particular demand (line 5) then one or several facilities are selected and the associated customer is removed from set $U^{t'}$ (line 24). Since it is not allowed to split an order for customer $j \in U^{t'}$ over multiple periods of time, when the residual capacity is insufficient to cover an order, its fulfillment must be further delayed and the customer remains in set $U^{t'}$. As a result, for $t < |T|$, not all sets $U^{t'}$ may be empty when Procedure AllocateNonPriorityDemand terminates.

3.2 Facility selection

In this section, we propose four alternative schemes for the Procedure SelectFacilities and show how they are embedded in the framework of Algorithm 1. In all of them, we choose in the first strategic period the (sub)set of facilities that will operate throughout the planning horizon. To this end, the existing, resp. new, facilities are gathered in set \hat{I}^e , resp. \hat{I}^n . In two of the schemes, the facility selection process is based on the estimation of the mean demand \bar{D} that needs to be satisfied per period,

$$\bar{D} = \frac{\sum_{t \in T} \sum_{j \in J} d_j^t}{|T|} + \bar{m} \sigma \quad (14)$$

where σ is the standard deviation of all the demands and \bar{m} is an integer parameter that will be specified later. Demand fluctuations are captured by expression (14).

In the first scheme, that leads to Heuristic 1, we decide to keep all existing facilities.

Algorithm AllocateNonPriorityDemand: Allocation of non-priority demand from customer segment J^1 to facilities operating in time period t

Input : $\widehat{I}_t^e, \widehat{I}_t^n$: existing facilities, resp. new, facilities operating in period t
 RQ^t : residual capacity vector for facilities operating in period t
 U^t ($t' = \max\{1, t - \rho + 1\}, \dots, t$): sets of customers belonging to J^1 with unfilled demands from period t'
Output: (Complete/partial) allocation of demands of customers from sets $U^{t'}$, $t' = \max\{1, t - \rho + 1\}, \dots, t$

```

1 Create list  $D$  of demands of all customers from  $U^{t'}$  ( $t' = \max\{1, t - \rho + 1\}, \dots, t$ ), sorted
  by non-increasing values of  $(p_j^{t'} + 1) d_j^{t'}$ 
2 repeat
3   Extract a demand from the top of  $D$ , say  $d_j^{t'}$ 
4   // Check if there is enough residual capacity
5   if  $d_j^{t'} \leq \sum_{i \in \widehat{I}_t^e \cup \widehat{I}_t^n} RQ_i^t$  then
6      $rd_j^{t'} := d_j^{t'}$  // (residual) demand of customer  $j$ 
7     Create list  $\widehat{I}_t$  of facilities  $\widehat{I}_t^e \cup \widehat{I}_t^n$  that have positive residual capacity; sort these
      facilities by non-decreasing cost  $o_{ik}^t + c_{ij}^t$  (with  $k = 1$  if  $i \in \widehat{I}_t^e$  and  $k = k_i^*$  if  $i \in \widehat{I}_t^n$ )
8     repeat
9       Extract a facility from the top of  $\widehat{I}_t$ , say  $i$ 
10      if  $rd_j^{t'} \leq RQ_i^t$  then
11        // Assign residual demand of customer  $j$  to facility  $i$ 
12         $s_{ij}^{t'} := rd_j^{t'}$ 
13         $RQ_i^t := RQ_i^t - rd_j^{t'}$  // update residual capacity
14         $w_{ik}^t := w_{ik}^t + rd_j^{t'}$ 
15         $rd_j^{t'} := 0$ 
16      else
17        // Assign part of residual demand of customer  $j$  to facility  $i$ 
18         $s_{ij}^{t'} := RQ_i^t$ 
19         $rd_j^{t'} := rd_j^{t'} - RQ_i^t$  // update residual demand of customer  $j$ 
20         $w_{ik}^t := w_{ik}^t + RQ_i^t$ 
21         $RQ_i^t := 0$ 
22      end
23    until  $rd_j^{t'} = 0$ 
24     $U^{t'} := U^{t'} \setminus \{j\}$ 
25     $v_j^{t'} := 1$ 
26  end
27 until  $D = \emptyset$ 
28 return  $v, s, w$ 

```

Therefore, $z_{i1}^\ell = 0$ for every $\ell \in T_L$ and $i \in I^e$. This choice is motivated by the fact that from a fixed cost perspective, we avoid paying closing costs by operating these facilities. If the total capacity of these facilities is adequate to cover the demand estimate \bar{D} for a pre-specified value of \bar{m} then no additional potential facilities are required and thus, $z_{ik_i}^\ell = 0$ for every $\ell \in T_L$, $i \in I^n$ and $k_i \in K_i$. Hence, in this case, our choice is defined by $\hat{I}^e = I^e$ and $\hat{I}^n = \emptyset$.

Often, the overall capacity of the existing facilities is insufficient to meet \bar{D} and as a result, additional facilities need to be chosen from the potential set I^n . The selection process is controlled by the total cost incurred by each potential facility to handle one unit of product per period, \overline{TC}_{ik} . This cost is estimated by the sum of four components: the mean processing cost, the mean distribution cost, the mean operating cost, and the fixed cost for establishing the facility. The latter depends on the capacity level that is chosen from the set of available discrete sizes. Hence,

$$\overline{TC}_{ik} = \frac{\sum_{t \in T} o_{ik}^t}{|T|} + \frac{\sum_{t \in T} \sum_{j \in J} c_{ij}^t}{|T| \cdot |J|} + \frac{\sum_{t \in T} M_{ik}^t}{Q_{ik} \cdot |T|} + \frac{FO_{ik}^1}{Q_{ik} \cdot |T|} \quad i \in I^n, k \in K_i \quad (15)$$

Since at most one facility can be established in a potential location with a given capacity level, we identify $k_i^* = \arg \min_{k \in K_i} \{\overline{TC}_{ik}\}$ for every $i \in I^n$, and discard all other capacity options. Due to economies of scale, the capacity size k_i^* is usually associated with the largest available capacity level.

In case $\sum_{i \in I^e} Q_{i1} < \bar{D}$, we build the list F^n of all facilities $i \in I^n$ sorted by non-decreasing estimated total costs \overline{TC}_{ik} , and choose the minimum number of potential facilities whose total capacity is at least equal to $\bar{D} - \sum_{i \in I^e} Q_{i1}$. The selected locations are gathered in set \hat{I}^n and the values of the associated location variables are fixed as follows for $i \in \hat{I}^n$: $z_{ik_i^*}^1 = 1$, $z_{ik}^1 = 0$ for $k \in K_i \setminus \{k_i^*\}$, and $z_{ik}^\ell = 0$ for $\ell \in T_L \setminus \{1\}$ and $k \in K_i$. Furthermore, for every $i \notin \hat{I}^n$, $k \in K_i$ and $\ell \in T_L$, we consider $z_{ik}^\ell = 0$.

Heuristic 1 uses the above strategy for facility selection under different demand estimates. Initially, the mean demand per period is estimated by taking $\bar{m} = 0$ in (14). After fixing $\hat{I}^e = I^e$ and having identified the set \hat{I}^n , we proceed with the allocation of customer demands, period by period, according to Procedure AllocateDemands (cf. Section 3.1). If the total available capacity is not enough to cover all demand requirements then a feasible solution cannot be constructed. In this case, we adjust our estimation of the mean demand to be filled per period by incrementing the value of parameter \bar{m} by 1, and use again the list F^n to enlarge our facility selection.

Heuristic 1: Construction of a feasible solution

Input : All instance data
Output: Feasible solution \mathcal{S}

- 1 // All existing facilities are maintained
- 2 $\hat{I}_t^e := I^e$ for all $t \in T$, $z_{i1}^\ell := 0$ for all $\ell \in T_L$, $i \in I^e$
- 3 Create list F^n of potential facilities sorted by non-decreasing estimated costs, $\overline{TC}_{ik_i^*}$
- 4 $\overline{m} := -1$
- 5 **repeat**
- 6 $\overline{m} := \overline{m} + 1$
- 7 $\hat{I}^n := \emptyset$ // no potential facilities are initially selected
- 8 $z_{ik_i}^\ell := 0$ for all $\ell \in T_L$, $i \in I^n$, $k_i \in K_i$
- 9 Estimate mean demand per period, \overline{D}
- 10 **if** $\sum_{i \in \hat{I}^e} Q_{i1} < \overline{D}$ **then**
- 11 Extract from F^n the minimum number of facilities whose total capacity plus $\sum_{i \in \hat{I}^e} Q_{i1}$ is
 capable to cover \overline{D}
- 12 Assign these potential facilities to \hat{I}^n and set $z_{ik_i^*}^1 := 1$ for $i \in \hat{I}^n$ and k_i^* denoting the
 selected capacity level
- 13 **end**
- 14 $\hat{I}_t^n := I^n$ for all $t \in T$
- 15 AllocateDemands($\hat{I}_1^e, \dots, \hat{I}_{|T|}^e, \hat{I}_1^n, \dots, \hat{I}_{|T|}^n$)
- 16 **until** *feasible solution is obtained*
- 17 **if** $\overline{m} = 0$ **then**
- 18 **repeat**
- 19 $\overline{m} := \overline{m} - 1$
- 20 $\hat{I}^n := \emptyset$ // no potential facilities are initially selected
- 21 $z_{ik_i}^\ell := 0$ for all $\ell \in T_L$, $i \in I^n$, $k_i \in K_i$
- 22 Estimate mean demand per period, \overline{D}
- 23 **if** $\sum_{i \in \hat{I}^e} Q_{i1} < \overline{D}$ **then**
- 24 Extract from F^n the minimum number of facilities whose total capacity plus
 $\sum_{i \in \hat{I}^e} Q_{i1}$ is capable to cover \overline{D}
- 25 Assign these potential facilities to \hat{I}^n and set $z_{ik_i^*}^1 := 1$ for $i \in \hat{I}^n$ and k_i^* denoting
 the selected capacity level
- 26 **end**
- 27 $\hat{I}_t^n := I^n$ for all $t \in T$
- 28 AllocateDemands($\hat{I}_1^e, \dots, \hat{I}_{|T|}^e, \hat{I}_1^n, \dots, \hat{I}_{|T|}^n$)
- 29 **until** *infeasible solution is obtained*
- 30 **end**
- 31 **return** best feasible solution $\mathcal{S} = (z, v, r, s, w)$

In the worst case, we only stop when all potential facilities have been chosen, i.e. when $\hat{I}^n = I^n$.

In case a feasible solution is directly identified with $\overline{m} = 0$, we reduce our demand estimate by setting $\overline{m} = -1$ in (14), and choose a sufficient number of potential locations (lines 19–24 in Heuristic 1). With a lower demand estimate we expect that fewer facilities will be selected, and as a result, the total cost will decrease. In all our numerical experiments, feasible solutions could be identified for values of parameter \overline{m} ranging from -3 to $+3$.

Heuristic 2 uses a similar construction mechanism as Heuristic 1, but it does not automatically select all existing facilities. In fact, the selection process takes into account the estimated cost to handle one unit of product per period for *all* locations. Therefore, we extend the estimate (15) to existing facilities as follows:

$$\overline{TC}_{i1} = \frac{\sum_{t \in T} o_{i1}^t}{|T|} + \frac{\sum_{t \in T} \sum_{j \in J} c_{ij}^t}{|T| \cdot |J|} + \frac{\sum_{t \in T} M_{i1}^t}{Q_{i1} \cdot |T|} \quad i \in I^e \quad (16)$$

Initially, we set $\widehat{I}^e = \emptyset$, $\widehat{I}^n = \emptyset$ and $z_{ik}^\ell = 0$ for all $\ell \in T_L$, $i \in I$, and $k \in K_i$. We then create the list F of facilities sorted by non-increasing values of (15) and (16) (line 1 in Heuristic 2). For a given value of \overline{m} , the minimum number of facilities with total capacity capable to satisfy the demand estimate (14) are chosen. Compared to Heuristic 1, this scheme may result in a different choice of the facilities to be operated throughout the planning horizon.

The third heuristic for constructing a feasible solution partially adopts the outcome of Heuristic 1. Specifically, all existing facilities are retained ($\widehat{I}^e = I^e$) but we only make use of the total number α of new facilities that are chosen by Heuristic 1, i.e. $\alpha = |\widehat{I}^n|$. Let $\mathcal{V}(\mathcal{S})$ be the objective function value of solution \mathcal{S} identified by Heuristic 1. To avoid the computational effort incurred by examining all possible combinations of α new facilities, we restrict our search by considering a subset of I^n , namely the $\alpha + m$ ($m \geq 1$) cheapest facilities according to their estimated costs (15). We start with $m = 1$ and examine all possible selections with α facilities. Naturally, the choice made in Heuristic 1 does not have to be inspected again and can thus be discarded. If a feasible solution can be identified with an objective function value lower than $\mathcal{V}(\mathcal{S})$, we stop. Otherwise, we take the $\alpha + 2$ ($m = 2$) cheapest new facilities, and evaluate all possible ways of selecting α facilities from this set. Again, we stop if we can construct a solution that is better than the one found by Heuristic 1. If that is not the case, we repeat this procedure with $m = 3$. For a given value of m , there are in total $\binom{\alpha+m}{\alpha}$ different selections, each with exactly α new facilities. However, in practice, we do not need to examine them all because some of them have already been evaluated upon considering $m - 1$. Hence, for a fixed

Heuristic 2: Construction of a feasible solution

Input : All instance data

Output: Feasible solution \mathcal{S}

```
1 Create list  $F$  of all facilities sorted by non-decreasing estimated costs
2  $\bar{m} := -1$ 
3 repeat
4    $\bar{m} := \bar{m} + 1$ 
5   Estimate mean demand per period,  $\bar{D}$ 
6   Extract from  $F$  the minimum number of facilities with total capacity capable to cover
    $\bar{D}$ ; specify  $\hat{I}^e, \hat{I}^n$  and the values of variables  $z$  accordingly
7    $\hat{I}_1^e := I^e$  and  $\hat{I}_t^e := \hat{I}^e$  for every  $t = 2, \dots, |T|$ 
8    $\hat{I}_t^n := \hat{I}^n$  for all  $t \in T$ 
9   AllocateDemands( $\hat{I}_1^e, \dots, \hat{I}_{|T|}^e, \hat{I}_1^n, \dots, \hat{I}_{|T|}^n$ )
10 until feasible solution is obtained
11 if  $\bar{m} = 0$  then
12   repeat
13      $\bar{m} := \bar{m} - 1$ 
14     Estimate mean demand per period,  $\bar{D}$ 
15     Extract from  $F$  the minimum number of facilities with total capacity capable to
     cover  $\bar{D}$ ; specify  $\hat{I}^e, \hat{I}^n$  and the values of variables  $z$  accordingly
16      $\hat{I}_1^e := I^e$  and  $\hat{I}_t^e := \hat{I}^e$  for every  $t = 2, \dots, |T|$ 
17      $\hat{I}_t^n := \hat{I}^n$  for all  $t \in T$ 
18     AllocateDemands( $\hat{I}_1^e, \dots, \hat{I}_{|T|}^e, \hat{I}_1^n, \dots, \hat{I}_{|T|}^n$ )
19   until infeasible solution is obtained
20 end
21 return best feasible solution  $\mathcal{S} = (z, v, r, s, w)$ 
```

value of m , the search for a better feasible solution involves inspecting $\binom{\alpha+m-1}{\alpha-1}$ possible facility selections. In the worst case, this procedure has to be executed for every $m = 1, \dots, |I^n| - \alpha$. This strategy, which we call Heuristic 3, always returns a feasible solution, even if it is the same as the one identified by Heuristic 1.

To illustrate how the set \hat{I}^n is specified in the second phase of Heuristic 3, let us assume that there are 5 potential locations for establishing new facilities and that the estimated cost (15) for location 1 is the smallest, whereas the estimated cost for location 5 is the largest. In other words, ranking the facilities by increasing estimated costs yields $F^n = (1, 2, 3, 4, 5)$. Moreover, let us suppose that according to Heuristic 1, two new facilities should be opened in period $\ell = 1$, in addition to keeping all existing facilities. Hence, $\hat{I}^e = I^e$, $\hat{I}^n = \{1, 2\}$ and $\alpha = 2$. Starting with $m = 1$, we consider the three cheapest facilities (1, 2, 3) and examine the pairs

$\{1, 3\}$ and $\{2, 3\}$. If none of them yields a solution with total cost lower than $\mathcal{V}(\mathcal{S})$ then we take the four cheapest facilities $(1, 2, 3, 4)$ by setting $m = 2$. In this case, from the $\binom{4}{2} = 6$ possible combinations with two facilities, we only need to investigate three of them, namely $\{1, 4\}$, $\{2, 4\}$ and $\{3, 4\}$, since the remaining options have already been inspected. Again, if a better feasible solution cannot be identified, we take $m = 3$. From the $\binom{5}{2} = 10$ options with two facilities, only four have to be examined, namely $\{1, 5\}$, $\{2, 5\}$, $\{3, 5\}$, and $\{4, 5\}$.

Heuristic 3: Construction of a feasible solution

Input : All instance data
Output: Feasible solution \mathcal{S}

- 1 Obtain feasible solution \mathcal{S} with Heuristic 1 and let $\mathcal{V}(\mathcal{S})$ be its objective value
- 2 $\hat{I}_1^e := I^e$ and $\hat{I}_t^e := \hat{I}^e$ for every $t = 2, \dots, |T|$ // keep all existing facilities
- 3 $\alpha := |\hat{I}^n|$ // number of new facilities to be selected
- 4 $\hat{I}^n := \emptyset$, $bestSol := false$
- 5 $m := 1$
- 6 **while** $m \leq |I^n| - \alpha$ **and** $bestSol = false$ **do**
- 7 Create list F^n with the $\alpha + m$ new facilities with lowest estimated costs, $\overline{TC}_{ik_i^*}$
- 8 **for** all combinations of α facilities from list F^n **do**
- 9 **if** combination was not inspected before **then**
- 10 Assign the selected α facilities to \hat{I}^n
- 11 $\hat{I}_t^n := \hat{I}^n$ for all $t \in T$
- 12 AllocateDemands($\hat{I}_1^e, \dots, \hat{I}_{|T|}^e, \hat{I}_1^n, \dots, \hat{I}_{|T|}^n$)
- 13 **if** feasible solution \mathcal{S}' is identified **and** $\mathcal{V}(\mathcal{S}') < \mathcal{V}(\mathcal{S})$ **then**
- 14 $\mathcal{S} := \mathcal{S}'$, $\mathcal{V}(\mathcal{S}) := \mathcal{V}(\mathcal{S}')$
- 15 $bestSol := true$
- 16 **end**
- 17 **end**
- 18 **end**
- 19 $m := m + 1$
- 20 **end**
- 21 **return** feasible solution $\mathcal{S} = (z, v, r, s, w)$

Finally, Heuristic 4 allows for the construction of a feasible solution in a similar way as Heuristic 3, but it partially uses the outcome of Heuristic 2. Accordingly, the (subset of) existing facilities selected by Heuristic 2 are operated throughout the planning horizon and the process of choosing $\alpha = |\hat{I}^n|$ new facilities to be opened at time period 1 is executed as in lines 5–20 (Heuristic 3).

4 Improvement heuristics

Given a feasible solution \mathcal{S} , different local transformations can be applied in an attempt to identify a better solution. In this section, we introduce two types of facility moves aiming, respectively, at postponing the time period in which a new facility is opened, and identifying the time period to close an existing facility. We have also developed a third strategy that replaces the capacity initially installed in new facilities by smaller capacity levels. However, in our numerical study, this type of move turned out to be ineffective (see Section 4.3).

A move is profitable if it maintains the feasibility of the solution and results in a decrease of the total cost. The evaluation of a move requires reassigning customers to the modified set of operating facilities by applying Procedure `AllocateDemands` (cf. Section 3.1). In order to efficiently detect improving moves, our approach does not enumerate and evaluate the large number of neighboring solutions. Instead, a greedy scheme is used that partially explores a neighborhood. Algorithm 2 outlines the general structure of the local search heuristic. At each iteration, the neighborhood of the incumbent solution is explored with a particular type of facility move. Whenever an improved solution is identified, it replaces the current solution and the search is restarted. Algorithm 2 can be executed only once with a particular choice of move, or several times according to a pre-specified sequence of facility moves.

Algorithm 2: General improvement scheme

Input : All instance data, feasible solution \mathcal{S}

Output: Feasible solution, \mathcal{S}^*

- 1 $\mathcal{S}^* := \mathcal{S}$, $\mathcal{V}(\mathcal{S}^*) := \mathcal{V}(\mathcal{S})$
 - 2 Choose type of facility move to use
 - 3 Apply local search to \mathcal{S}^*
 - 4 **if** *profitable feasible solution \mathcal{S}' is identified* **then** $\mathcal{S}^* := \mathcal{S}'$ and $\mathcal{V}(\mathcal{S}^*) := \mathcal{V}(\mathcal{S}')$
 - 5 **return** best feasible solution, $\mathcal{S}^* = (\mathbf{z}, \mathbf{v}, \mathbf{r}, \mathbf{s}, \mathbf{w})$
-

Again, we denote by \widehat{I}_t^e , resp. \widehat{I}_t^n , the set of existing, resp. new, facilities operating in time period t ($t \in T$) according to the initial solution \mathcal{S} . If \mathcal{S} has been obtained with one of the constructive heuristics then $\widehat{I}_1^e := I^e$, $\widehat{I}_t^e := \widehat{I}^e$ for every $t = 2, \dots, |T|$, and $\widehat{I}_t^n := \widehat{I}^n$ for all $t \in T$. However, if \mathcal{S} is the best cost-decreasing solution yielded by the exploration of a facility move then sets \widehat{I}_t^e and \widehat{I}_t^n are initialized accordingly. Next, we describe how local search is performed (line 3) according to a particular choice of facility move.

4.1 Postponing moves

The first improvement strategy, denoted LI1, attempts to open new facilities as late as possible. All pre-selected existing facilities $i \in \widehat{I}_t^e$ ($t \in T$) continue to be available. Any postponement of the opening of a new facility uniquely defines a new solution. Depending on the total number of strategic periods and the total number of open new facilities, the size of the neighborhood can be excessively large if we consider all possibilities of shifting the period in which each new facility is opened. In order to overcome this difficulty, we only search a restricted facility neighborhood. To this end, we sequentially visit the list of strategic periods, and at each time period $\ell \in T_L$ we examine the possibility of constructing a feasible solution without having to operate one or several of the new facilities in that period. The shift of the opening period of a new facility yields savings in the fixed facility operating costs for all tactical periods until the next strategic period, i.e. for periods $t = \ell, \dots, \phi(\ell)$. These savings may be offset by additional facility processing costs, distribution costs and tardiness penalty costs due to the reallocation of some customer demands. Hence, a move may not be necessarily profitable.

Since postponing the time period in which a facility is opened reduces the total capacity available in the network, we sort the facilities in $\widehat{I}^n = \bigcup_{t \in T} \widehat{I}_t^n$ by non-decreasing order of their capacities. Let \underline{I}^n denote this list. At a given strategic period ℓ , we consider the new facilities in list \underline{I}^n in turn as indicated by the ordering. Therefore, we assume that there is a higher chance to maintain feasibility when the opening of a relatively small facility is postponed rather than the setup of a large facility. If the decision to shift the opening of a particular facility, say $i \in \underline{I}^n$ with capacity level k_i^* , one strategic period ahead, i.e. from period ℓ to period $\ell + 1$, results in an illegal move then we know that this facility must be operated until the end of the planning horizon. Hence, facility i is excluded from further consideration in future strategic periods and is removed from list \underline{I}^n . Facility i is also eliminated from \underline{I}^n when the move is legal but does not yield an improved solution. In contrast, if a feasible and profitable solution can be constructed without operating facility i in periods $\ell, \dots, \phi(\ell)$, this change is implemented and the new solution becomes the incumbent solution. This process is repeated until we reach the last strategic period ℓ_{max} . The pseudo-code `Local improvement 1` details this procedure. At the end of the algorithm, the best solution identified is returned.

Local improvement 1: PostponeFacilityOpenings - Defer opening new facilities

Input : All instance data, feasible solution \mathcal{S}

Output: Feasible solution \mathcal{S}^*

```
1  $\mathcal{S}^* := \mathcal{S}$  and  $\mathcal{V}(\mathcal{S}^*) := \mathcal{V}(\mathcal{S})$ 
2 Initialize sets  $\widehat{I}_t^e$  and  $\widehat{I}_t^n$ , for all  $t \in T$ , according to the facilities selected in  $\mathcal{S}$ 
3 Create list  $\underline{I}^n$  of new facilities  $\widehat{I}^n = \bigcup_{t \in T} \widehat{I}_t^n$  sorted by non-decreasing capacities
4  $\ell := 1$ 
5 repeat
6    $F := \underline{I}^n$ 
7   repeat
8     Extract a facility from the top of  $F$ , say  $i$ 
9     // Check if opening of facility  $i$  can be postponed to next strategic
      period
10     $\widehat{I}_t^n := \widehat{I}_t^n \setminus \{i\}$  for all  $t = \ell, \dots, \phi(\ell)$ 
11    AllocateDemands( $\widehat{I}_1^e, \dots, \widehat{I}_{|T|}^e, \widehat{I}_1^n, \dots, \widehat{I}_{|T|}^n$ )
12    if feasible solution  $\mathcal{S}'$  is constructed and  $\mathcal{V}(\mathcal{S}') < \mathcal{V}(\mathcal{S}^*)$  then
13       $z_{ik_i}^\ell := 0$  // facility  $i$  is not opened in strategic period  $\ell$ 
14       $\mathcal{S}^* := \mathcal{S}'$  and  $\mathcal{V}(\mathcal{S}^*) := \mathcal{V}(\mathcal{S}')$ 
15    else
16      // Opening of facility  $i$  is not postponed to next strategic period
17       $\widehat{I}_t^n := \widehat{I}_t^n \cup \{i\}$  for all  $t = \ell, \dots, \phi(\ell)$ 
18       $\underline{I}^n := \underline{I}^n \setminus \{i\}$  // further postponement is not possible
19    end
20  until  $F = \emptyset$ 
21   $\ell := \ell + 1$ 
22 until last strategic period  $\ell_{max}$ 
23 return best feasible solution  $\mathcal{S}^* = (z, v, r, s, w)$ 
```

4.2 Closing moves

The aim of the second local improvement mechanism, denoted LI2, is to decide if and when selected existing facilities \widehat{I}^e should be removed. The choices previously made with respect to new facilities and their capacity levels remain unaltered. As for the postponing move, we only explore a restricted neighborhood generated by closing moves.

When a facility is closed at the end of a given strategic period then fixed facility operating costs are saved until the end of the planning horizon. These may counterbalance the fixed closing cost that is charged and possibly also the higher cost incurred by reassigning the demands previously allocated to this facility. Even though the largest saving can be expected to be obtained when an existing facility is closed at the end of the first period, this action also poses

a greater risk of not producing a feasible solution. Therefore, closing moves are explored by starting from the last strategic period ℓ_{max} and stopping at the first period. Similar to the first local improvement scheme, we sort the facilities in $\widehat{I}^e = \bigcup_{t \in T} \widehat{I}_t^e$ by non-decreasing capacities. At each strategic period ℓ , we use this ordering to examine the impact of closing each one of these facilities at the end of ℓ . If a feasible solution is detected, it becomes the incumbent solution provided it is also profitable. In the event that the move is illegal, the facility is removed from the sorted list. This exclusion is motivated by the fact that trying to close the facility earlier will also not yield a feasible solution, and thus computational effort can be saved. A legal but unprofitable move leads to the same action.

Local improvement 2: CloseExistingFacilities - Schedule facility closings

Input : All instance data, feasible solution \mathcal{S}

Output: Feasible solution \mathcal{S}^*

```

1  $\mathcal{S}^* := \mathcal{S}$  and  $\mathcal{V}(\mathcal{S}^*) := \mathcal{V}(\mathcal{S})$ 
2 Initialize sets  $\widehat{I}_t^e$  and  $\widehat{I}_t^n$ , for all  $t \in T$ , according to the facilities selected in  $\mathcal{S}$ 
3 Create list  $\underline{I}^e$  of existing facilities  $\widehat{I}^e = \bigcup_{t \in T} \widehat{I}_t^e$  sorted by non-decreasing capacities
4  $\ell := \ell_{max}$ 
5 repeat
6    $F := \underline{I}^e$ 
7   repeat
8     Extract a facility from the top of  $F$ , say  $i$ 
9     // Check if facility  $i$  can be closed at the end of strategic period  $\ell$ 
10     $\widehat{I}_t^e := \widehat{I}_t^e \setminus \{i\}$  for all  $\ell < t \leq \phi(\ell)$ 
11    AllocateDemands( $\widehat{I}_1^e, \dots, \widehat{I}_{|T|}^e, \widehat{I}_1^n, \dots, \widehat{I}_{|T|}^n$ )
12    if feasible solution  $\mathcal{S}'$  is constructed and  $\mathcal{V}(\mathcal{S}') < \mathcal{V}(\mathcal{S}^*)$  then
13       $z_{i1}^\ell := 1$  // facility  $i$  is closed at the end of period  $\ell$ 
14       $\mathcal{S}^* := \mathcal{S}'$  and  $\mathcal{V}(\mathcal{S}^*) := \mathcal{V}(\mathcal{S}')$ 
15    else
16      // Facility  $i$  cannot be closed at the end of strategic period  $\ell$ 
17       $\widehat{I}_t^e := \widehat{I}_t^e \cup \{i\}$  for all  $\ell < t \leq \phi(\ell)$ 
18       $\underline{I}^e := \underline{I}^e \setminus \{i\}$  // earlier closing is also not possible
19    end
20  until  $F = \emptyset$ 
21   $\ell := \ell - 1$ 
22 until first strategic period
23 return best feasible solution  $\mathcal{S}^* = (z, v, r, s, w)$ 

```

If a closing move is successful in the first strategic period then the associated existing facility is not active throughout the planning horizon except in the first period. The Procedure

Local Improvement 2 describes how closing moves are performed. The most profitable solution is returned at the end of this process.

4.3 Other moves

We have also developed a third local improvement mechanism, denoted LI3, that explores the possibility to exchange the capacity level of a new facility for a smaller capacity size. Due to economies of scale, cost savings are expected to be achieved in opening the facility, operating it, and processing deliveries to the customers assigned to it. As with the postponing and closing moves, we perform a restricted neighborhood search for capacity exchange moves. We start by sorting the pre-selected new facilities \hat{I}^n by non-decreasing order of the capacity levels Q_{ik} that were installed in the new locations. These facilities are examined in turn as indicated by the ordering. For a given facility i , we replace its current capacity level with the next smaller size in all periods the facility is available. If successful postponing moves were performed earlier then the facility may be opened in some strategic period ℓ such that $\ell > 1$. A profitable exchange move results in replacing the current solution with the improved neighbor. In contrast, if the move yields an infeasible solution, we restore feasibility by discarding the capacity level exchange. Moreover, we also exclude the facility from further consideration since lower capacity sizes will also not maintain the feasibility of the solution. After having inspected all new facilities sequentially and updated the set \hat{I}^n , this process is restarted in an attempt to further decrease the capacity of one or several new facilities. The pseudo-code Local improvement 3 details this local search procedure.

In our computational experiments, capacity exchange moves (LI3) never generated better solutions. Therefore, we report in the next section only the results obtained with postponing moves (LI1) and closing moves (LI2). We attribute the ineffectiveness of procedure LI3 to the magnitude of the economies of scale that are present in our instances in the fixed cost of opening a new facility with a given capacity level. This characteristic is also evidenced by the solutions provided by a state-of-the-art MILP solver, with new facilities being often operated with the largest capacity level available. Since the constructive heuristics always select the largest capacity size in a new location, the outcome of applying LI3 was expected. However, we believe that this procedure may facilitate the improvement of a solution to a problem instance having smaller differences between the costs per unit of capacity installed in a potential facility location $(FO_{ik}^\ell/Q_{ik}, i \in I^n, k \in K_i)$.

Local improvement 3: ExchangeCapacity - Decrease capacity of new facilities

Input : All instance data, feasible solution \mathcal{S}

Output: Feasible solution \mathcal{S}^*

```
1  $\mathcal{S}^* := \mathcal{S}$  and  $\mathcal{V}(\mathcal{S}^*) := \mathcal{V}(\mathcal{S})$ 
2 Initialize sets  $\widehat{I}_t^e$  and  $\widehat{I}_t^n$ , for all  $t \in T$ , according to the facilities selected in  $\mathcal{S}$ 
3 Create list  $\underline{I}^n$  of new facilities  $\widehat{I}^n = \bigcup_{t \in T} \widehat{I}_t^n$  sorted by non-decreasing capacities
4 repeat
5    $F := \underline{I}^n$ 
6   repeat
7     Extract a facility from the top of  $F$ , say  $i$ ; let  $Q_{ik}$  be its current capacity level and
8      $\ell$  be the strategic time period in which the facility is opened
9     if  $k > 1$  then
10      Replace capacity  $Q_{ik}$  with  $Q_{i,k-1}$  in all periods that facility  $i$  is available
11      AllocateDemands( $\widehat{I}_1^e, \dots, \widehat{I}_{|T|}^e, \widehat{I}_1^n, \dots, \widehat{I}_{|T|}^n$ )
12      if feasible solution  $\mathcal{S}'$  is constructed and  $\mathcal{V}(\mathcal{S}') < \mathcal{V}(\mathcal{S}^*)$  then
13         $z_{ik}^\ell := 0$  for all  $\ell \in T_L$ 
14         $z_{i,k-1}^\ell := 1$  for all  $\ell \in T_L$  such that  $i \in \widehat{I}_\ell^n$ 
15         $\mathcal{S}^* := \mathcal{S}'$  and  $\mathcal{V}(\mathcal{S}^*) := \mathcal{V}(\mathcal{S}')$  // better solution
16        if  $k - 1 = 1$  then  $\underline{I}^n := \underline{I}^n \setminus \{i\}$  // further capacity decrease not
17        possible
18      else
19        // Capacity level cannot be exchanged for a smaller size
20         $\underline{I}^n := \underline{I}^n \setminus \{i\}$ 
21      end
22    end
23  until  $F = \emptyset$ 
24 until  $\underline{I}^n = \emptyset$ 
25 return best feasible solution  $\mathcal{S}^* = (z, v, r, s, w)$ 
```

5 Computational study

In this section, we assess the computational efficiency of the proposed heuristics and the quality of the solutions obtained for a set of randomly generated test instances. Our heuristic solutions are compared against the solutions identified with IBM ILOG CPLEX 12.3 within a time limit of 36000 CPU seconds. The heuristics were implemented in C++ and all heuristic experiments were performed on a laptop computer with a 1.33 GHz Intel Atom[®] processor Z3740 with 2 GB RAM and running Windows 8.1 (32-bit operating system).

5.1 Test instances

We consider the two sets of test instances that were randomly generated by Correia and Melo (2016). These sets mainly differ in the total number of customers, namely one set has 100 customers and the other 150 customers. We have also enlarged this testbed with a third set of instances, each having 200 customers. In total, 162 instances were randomly created by combining the parameters shown in Table 2, thereby obtaining 54 instances for each value of $|J|$. The planning horizon is divided into 36 time periods. Two different temporal opportunities for facility and capacity acquisition decisions are considered, specifically $T_L = \{1, 13, 25\}$ and $T_L = \{1, 7, 13, 19, 25, 31\}$. In the first time period, the demand of each customer is selected at random from the interval $[20, 100]$ according to a continuous uniform distribution. In each subsequent period, demands exhibit fluctuations ranging from -5% to $+5\%$ compared to the previous period. Accordingly, in each time period t , the rate of change is chosen randomly from the interval $[0.95, 1.05]$ for customer j and multiplied by the demand of this customer for period $t - 1$ ($t \geq 2$). In each potential location, three different capacity levels are available, representing small, medium and large sizes. The medium (small) capacity option corresponds to 70% (49%) of the large size and the associated fixed costs exhibit economies of scale. For further details on the generation of the test instances, we refer to Correia and Melo (2016), which also includes a sophisticated scheme for obtaining the cost parameters.

Parameter	Value
$ J $	100, 150, 200
$ J^0 $	$\lceil \beta^J J \rceil$ with $\beta^J \in \{0.25, 0.5, 0.75\}$
$ I $	$0.1 J $
$ I^n $	$0.8 I $ ($ I^e = 0.2 I $)
$ K_i $	3 ($i \in I^n$)
$ T $	36
$ T_L $	3, 6
ρ	1, 2, 3

Table 2: Parameter values

5.2 Analysis of results

In Table 3, the type of solutions obtained with CPLEX and the constructive heuristics is summarized for different values of the maximum allowed delay for demand fulfilment (column 1).

Columns 2–4 refer to CPLEX and indicate the number of instances solved to optimality ($\#$ opt sol), the number of instances not solved to proven optimality within the specified time limit ($\#$ non-opt sol), and the number of instances for which CPLEX failed to identify a single feasible solution ($\#$ unsolved inst). For each constructive heuristic, the following information is presented: the number of instances for which the heuristic created a feasible solution ($\#$ feas sol, columns 5, 7, 9 and 11) and the number of solutions with higher quality than the best solutions reported by CPLEX ($\#$ impr sol over CPLEX, columns 6, 8, 10 and 12). The last row (All) in Table 3 provides the above information over all test instances.

Increasing the maximum allowed delivery time results in more challenging problems and this is reflected in the growing number of instances that cannot be solved to optimality by CPLEX within the computing time of 10 h (in total, 84% or 136 instances). Furthermore, CPLEX is unable to find feasible solutions to 5.6% of the instances. These features are associated with the large size of the instances and, as mentioned earlier, the additional complexity posed by parameter ρ . On average, the MIP formulation has 194,606 binary variables, 173,979 continuous variables, and 18,220 constraints. Additionally, as the value of ρ increases, more opportunities arise for satisfying the demands of customer segment J^1 and these have to be investigated. While the satisfaction of a customer order can be delayed at most one time period when $\rho = 1$, taking $\rho = 3$ leads to four alternative time periods for meeting the demands of these customers (either on time or with a delay of one, two or three periods). This also explains why a significantly larger number of instances could be solved to optimality with $\rho = 1$ against $\rho \in \{2, 3\}$. In contrast to CPLEX, all heuristics construct a feasible solution to every instance (100% feasible solutions versus 94.4% with CPLEX). It is also noteworthy that the heuristics even return a better solution than CPLEX for a few instances.

ρ	CPLEX			Heuristic 1		Heuristic 2		Heuristic 3		Heuristic 4	
	$\#$ opt sol	$\#$ non-opt sol	$\#$ un-solved inst	$\#$ feas sol	$\#$ impr sol over CPLEX	$\#$ feas sol	$\#$ impr sol over CPLEX	$\#$ feas sol	$\#$ impr sol over CPLEX	$\#$ feas sol	$\#$ impr sol over CPLEX
1	13	36	5	54	3	54	4	54	3	54	4
2	3	50	1	54	0	54	0	54	0	54	1
3	1	50	3	54	0	54	0	54	0	54	0
All	17	136	9	162	3	162	4	162	3	162	5

Table 3: Solutions obtained with CPLEX and the constructive heuristics.

Table 4 reports the quality of the solutions identified by CPLEX and the constructive heuristics. For each value of ρ considered, this table indicates the minimum (min), average (avg) and maximum (max) integrality gaps achieved by CPLEX at termination (MIP gap (%), column 3).

In addition, the minimum, average and maximum CPU times (in seconds) are also presented for CPLEX (CPU (s), column 4). Columns 5–8 report for each constructive heuristic the minimum, average and maximum GAP-UB in percent, which is the gap between the objective value of the heuristic solution (z^H) and the objective value of the best solution identified by CPLEX (z) within the time limit, i.e. $(z^H - z)/z \times 100\%$. A negative value indicates a saving achieved by the heuristic under examination over CPLEX. The last row in Table 4 displays the average values over all test instances. The CPU times of the heuristics are omitted from this table because our code and the Correia and Melo (2016) code were run on two different computers. Therefore, direct comparisons between CPLEX and the heuristics are not possible. However, the computing time does not seem to be the critical issue for our approaches. All runs with Heuristics 1 and 2 required less than 30 CPU seconds. Runs with Heuristics 3 and 4 demanded more computing time due to the procedure developed to select new facilities, with an average of 1.2 and 3.5 CPU minutes, respectively. We note that the processor of the computer used for the experiments with the heuristics is considerably less powerful than the computer used by Correia and Melo (2016) to run the commercial code CPLEX.

ρ		CPLEX		Heuristic 1	Heuristic 2	Heuristic 3	Heuristic 4
		MIP gap (%)	CPU (s)	Gap-UB (%)	Gap-UB (%)	Gap-UB (%)	Gap-UB (%)
1	min	0.00	183.50	-4.27	-3.32	-4.40	-3.90
	avg	1.39	30890.78	4.33	3.61	3.89	3.07
	max	6.15	36000.00	11.79	11.01	11.60	10.56
2	min	0.00	956.07	0.67	0.01	0.33	-0.17
	avg	1.64	34190.46	4.43	3.25	4.03	2.84
	max	4.36	36000.00	11.60	10.71	11.41	10.27
3	min	0.00	25266.11	0.79	0.43	0.52	0.31
	avg	1.85	35789.53	4.59	3.63	4.24	3.25
	max	4.60	36000.00	9.32	10.40	8.65	9.96
All	avg	1.64	33666.72	4.45	3.49	4.06	3.05

Table 4: Performance of the constructive heuristics.

Even though proven optimal solutions are only available for 10.5% (17/162) of the instances, the integrality gaps reported by CPLEX are, on average, rather small (less than 2%) and the maximum gaps are lower than 5% with the exception of a single instance with 150 customers and $\rho = 1$ for which an integrality gap of 6.15% is achieved. Since the maximum computing time is attained by CPLEX for the majority of the instances, it is not surprising that good

solutions are found at the expense of high average CPU time. The quality of the solutions produced by the four constructive heuristics is not significantly worse than that of the CPLEX solutions. Depending on the heuristic and the value of ρ , on average, a feasible solution is 2.84% to 4.59% more costly than the best solution identified by CPLEX. Moreover, each heuristic is robust with respect to the different values for the maximum allowed delay in demand fulfilment. Three solutions are constructed by Heuristic 1 that outperform CPLEX, with a reduction of the corresponding objective value of 0.39%, 3.32% and 4.27%. With Heuristic 2, the improvement over CPLEX occurs in four instances and amounts to 0.39%, 1.17%, 1.74%, and 3.32%, respectively. The magnitude of the improved solutions obtained with Heuristic 3 is similar to that of Heuristic 1. Heuristic 4 produces four solutions that are 0.44%, 1.64%, 2.14%, and 3.90% better than those found by CPLEX.

Among the four constructive heuristics, Heuristic 1 seems to be the weakest. This is not surprising since this procedure offers the least flexibility for selecting the facilities to operate over the planning horizon. In particular, all the existing facilities are retained as opposed to the choices made by Heuristic 2. Heuristic 3 also exhibits this characteristic but invests more time in searching for an improved subset of the new facilities to establish. As a result, the quality of the solutions obtained is, on average, superior to Heuristic 1 but still inferior to Heuristic 2. Heuristic 4 retains the subset of existing facilities identified by Heuristic 2 and makes use of the same flexible approach for selecting new facilities as Heuristic 3. The combination of these strategies brings additional benefits and so Heuristic 4 performs consistently better than the other three constructive heuristics.

Table 5 reports the results obtained for the proposed local improvements applied to all constructive heuristics. The table shows the minimum, average and maximum percent deviations of the heuristic solutions from the best CPLEX solutions (Gap-UB (%)) for executing one local improvement strategy (LI1 or LI2) or two local improvement strategies (LI1 followed by LI2 or LI2 followed by LI1). All possible sequences of local improvements are tested on the initial solutions produced by the four constructive heuristics, yielding in total 16 different ways to obtain feasible solutions to the problem at hand. For each value of parameter ρ and each heuristic used in the first phase, the average gap is highlighted in boldface for the best sequence of local improvements. The last row in Table 5 summarizes the average gaps over all instances. The computing times are omitted from this table because LI1 and LI2 are extremely fast (< 60 CPU seconds).

As can be observed in Table 5, all local improvement schemes succeed in making transfor-

ρ	Gap-UB (%)																
	Heuristic 1				Heuristic 2				Heuristic 3				Heuristic 4				
	LI1	LI2	LI1 + LI2	LI2 + LI1	LI1	LI2	LI1 + LI2	LI2 + LI1	LI1	LI2	LI1 + LI2	LI2 + LI1	LI1	LI2	LI1 + LI2	LI2 + LI1	
1	min	-4.27	-4.27	-4.27	-4.27	-3.32	-3.32	-3.32	-3.32	-4.40	-4.40	-4.40	-4.40	-3.90	-3.90	-3.90	-3.90
	avg	3.03	1.40	0.82	0.49	2.29	2.75	1.72	1.64	2.67	1.31	0.71	0.41	1.85	2.53	1.52	1.48
	max	11.79	7.39	6.08	3.51	10.57	11.01	10.57	10.57	11.60	7.39	5.89	4.04	10.39	10.56	10.39	10.39
2	min	-1.23	-2.06	-2.06	-2.06	-1.23	0.01	-1.23	-1.23	-0.88	-1.51	-1.51	-1.51	-1.13	-0.17	-1.13	-1.13
	avg	3.36	1.79	1.41	1.08	2.22	2.53	1.84	1.67	3.06	1.81	1.39	1.11	1.90	2.39	1.61	1.56
	max	11.60	7.10	7.03	4.51	7.38	10.71	5.80	5.80	11.41	7.10	6.79	4.22	7.12	10.27	6.01	6.01
3	min	0.47	-1.56	-1.56	-1.56	-0.25	-0.87	-0.25	-0.87	0.33	-1.56	-1.56	-1.56	-0.25	0.31	-0.73	-0.73
	avg	3.57	2.41	2.00	1.76	2.69	2.99	2.26	2.17	3.38	2.46	1.95	1.80	2.44	2.90	2.16	2.13
	max	8.70	8.11	8.11	8.11	7.79	10.40	7.79	7.79	8.65	8.12	8.12	8.12	7.79	9.96	7.79	7.79
All	avg	3.32	1.87	1.42	1.12	2.40	2.75	1.94	1.83	3.04	1.87	1.36	1.12	2.07	2.61	1.76	1.73

Table 5: Performance of the local improvements.

mations that yield cost reductions, even though with varying degree of enhancement. When a single procedure is employed, LI1 performs better with Heuristics 2 and 4, whereas LI2 provides better solutions with Heuristics 1 and 3. Recall that the latter constructive heuristics have in common the operation of all existing facilities from period 1 until period $|T|$. Depending on the test instance, this corresponds to operating 2, 3 or 4 facilities (3 facilities on average). Hence, deciding if and when these facilities should be closed by means of LI2 yields additional cost savings. In contrast, most of the solutions constructed by Heuristics 2 and 4 already include a subset of the existing facilities to be in place over the whole planning horizon. In fact, an average of 1.7 facilities belonging to I^e are chosen and in about 10% of the instances no existing facility is even selected. Since the capacity provided by the chosen existing facilities is often insufficient to cover all demands, additional capacity needs to be made available through establishing new facilities. This results in opening, on average, 4.7 facilities (for comparison, an average of 4.2 facilities are opened with Heuristics 1 and 3). The decision about when these facilities should be set up is ruled by LI1 and this strategy yields a stronger improvement due to the larger number of new facilities involved. Additional enhancements are achieved when both types of local moves are executed. For all initial solutions and all values of ρ considered, the sequence LI2 + LI1 always returns the best solutions. Closing moves (LI2) aim at reducing the amount of capacity available in the system towards the end of the planning horizon, while postponing moves (LI1) aim at decreasing the capacity available in the first strategic periods. By readjusting the time periods during which the selected existing and new facilities are operating, excess capacity tends to be eliminated. We have observed that when LI2 is run before LI1, slightly fewer existing facilities are kept, as opposed to the sequence LI1 + LI2.

Although Heuristic 4 provides, on average, the best solutions during the construction phase, it is noteworthy that the local improvements are more effective when applied to the lower quality solutions obtained with Heuristics 1 and 3. The latter solutions leave more room for improvement which is exploited by LI1 and LI2. For the instances with $\rho = 1$, Heuristic 3 together with the sequence LI2 + LI1 achieves the best solutions with an average deviation of 0.41% to CPLEX solutions. Moreover, 18.5% of the solutions obtained (10/54) strictly outperform CPLEX. For the larger delivery delays $\rho = 2$ and $\rho = 3$, Heuristic 1 followed by LI2 + LI1 yields the best solutions with an average deviation of 1.08% and 1.76% to CPLEX, respectively. Improved solutions over CPLEX are returned in 5 ($\rho = 2$) and 4 instances ($\rho = 3$).

5.3 Additional insights

Since the constructive heuristics and the local improvement strategies are undemanding in their computational requirements, they can all be applied to a problem instance and the best solution can be retained at the end. Table 6 reports the performance of the best solutions obtained for every instance and after each phase. Columns 2–4 give the number of instances for which the best heuristic solutions outperform the best solutions found by CPLEX within the time limit of 10 CPU hours. Columns 6–8 present the quality of the heuristic solutions in terms of the minimum, average and maximum deviations to CPLEX (GAP-UB (%)). The last row (All) summarizes the results over all test instances. The effectiveness of the local improvements is evidenced by the generation of 22 solutions with higher quality than the CPLEX solutions, against the 5 better solutions obtained during the construction phase. Since CPLEX failed to identify feasible solutions to 9 instances (cf. Table 3) and the heuristics always find a feasible solution, in total 31 solutions outperform CPLEX (19.1%). While the construction phase yields an average deviation of 2.49% to CPLEX, the improvement phase results in a significant decrease of the average gap to 0.72%. For $\rho = 1$ and $\rho = 2$, the solutions deviate at most 2.19% and 3.88%, respectively. For $\rho = 3$, in all instances, except one, the best solutions are at most 3.49% more expensive than CPLEX. In the outlier instance, a gap of 7.79% is achieved. For the 17 instances for which a guaranteed optimal solution is available (cf. Table 3), the quality of the best heuristic solutions can be assessed. In 11 of these instances, at least one heuristic provided a solution that is within 1% of the optimal solution. In 5 other instances, the quality of the best heuristic solution obtained varies between 1% and 1.5%. Finally, there is one single instance for which the best heuristic solution deviates 2.19% from the optimal solution.

ρ	# impr sol over CPLEX				Gap-UB (%)		
	Constructive Phase	Improvement Phase			Constructive Phase	Improvement Phase	
		with 1 LI	with 2 LI			with 1 LI	with 2 LI
1	9	15	15	min	-4.40	-4.40	-4.40
				avg	2.38	0.16	0.12
				max	10.39	2.19	2.19
2	2	8	9	min	-0.17	-2.06	-2.06
				avg	2.34	0.74	0.71
				max	7.10	3.88	3.88
3	3	7	7	min	0.31	-1.56	-1.56
				avg	2.75	1.31	1.30
				max	7.79	7.79	7.79
All	14	30	31	avg	2.49	0.75	0.72

Table 6: Performance of the best heuristic solutions.

In order to apprehend the characteristics of the best solutions obtained at the end of the improvement phase, we have compared them to the best solutions provided by CPLEX (153 instances) with respect to different cost categories. The minimum, average and maximum percent deviations are presented in Table 7. The deviation is calculated as $(c^H - c')/c' \times 100\%$ for each cost category, where c^H and c' denote the cost obtained by the heuristic and by CPLEX, respectively. A negative value indicates that a lower cost is achieved by the heuristic solution over CPLEX for the category under examination. The heuristics show a tendency to operate more existing facilities for a longer period of time, which results in savings in the closing costs but in increased costs for maintaining these facilities. A significant variability within each of these two cost categories is observed. The average investment spending on opening new facilities does not show large differences against the setup costs in the CPLEX solutions. While the average costs of processing customer orders at the operating facilities are slightly lower than the corresponding costs provided by CPLEX, the costs of distributing the product to the customers are more expensive, even though the differences are not substantial.

ρ	Deviation to CPLEX (%)							
	New facilities		Existing facilities		Processing cost	Distribution cost	Tardiness penalty cost	
	Opening cost	Maint. cost	Closing cost	Maint. cost				
1	min	-24.7	-22.3	-100.0	-41.3	-7.4	-0.5	-100.0
	avg	-0.7	-2.0	-17.7	10.8	-0.9	1.7	771.6
	max	16.8	7.7	0.0	447.7	2.8	4.9	11420.9
2	min	-26.2	-22.3	-100.0	-86.5	-5.5	-0.6	-100.0
	avg	-0.7	-1.1	-10.1	8.5	-0.7	1.6	1313.8
	max	11.8	7.2	38.8	424.7	2.7	4.8	21532.3
3	min	-22.3	-22.3	-100.0	-97.3	-6.5	-0.3	-100.0
	avg	0.8	1.6	-10.9	7.9	-0.8	1.5	2235.1
	max	31.5	31.5	37.6	447.7	2.6	5.4	36686.7
All	avg	-0.2	-0.5	-12.8	9.0	-0.8	1.6	1442.1

Table 7: Cost deviations of the best heuristic solutions to CPLEX solutions.

The most striking discrepancy concerns the tardiness penalty cost, with the heuristics incurring more delays in customer satisfaction in 68% of the instances. Late shipments do not occur in the best heuristic solutions identified to 14/153 instances compared to the CPLEX solutions. This feature produces the best possible deviation (-100%). Recall that the decisions on demand allocation are made by the Procedure AllocateDemands, which is used by all heuristics.

Therefore, it seems that there is some potential for enhancing the mechanism that determines the time period in which a customer order should be met. This decision is certainly affected by the facilities available in a certain period and their capacities. Interestingly, the average share of the tardiness penalty cost over the total cost is rather small, namely 0.3% ($\rho = 1$), 0.6% ($\rho = 2$) and 0.9% ($\rho = 3$). In the CPLEX solutions, this share represents 0.2% of the total objective value for every value of ρ considered. The savings achieved in some cost categories are weighed against the additional spending on other categories, and therefore, the quality of the solutions generated by the best heuristics turns out to be very good.

Further insights on the characteristics of the best heuristic solutions are provided in Table 8 with respect to the utilization level of the capacity of operating facilities. Similar information is also displayed for the best CPLEX solutions. Interestingly, both groups of solutions exhibit similar average capacity utilization rates. The only difference occurs in the minimum values for the new facilities, with the heuristics producing more slack capacity in a few instances. Table 8 also presents the mean number of new facilities that are opened and the mean number of initially existing facilities that are closed at the end of some strategic period. As observed earlier, the best heuristic solutions are characterized by fewer facility closings and openings compared to the CPLEX solutions.

Sol method	ρ	Capacity usage (%)						# facilities (avg)	
		new facilities			existing facilities			new	closed
		min	avg	max	min	avg	max		
CPLEX	1	87.9	94.9	99.9	73.2	89.8	100.0	4.4	1.5
	2	88.9	96.1	100.0	70.5	90.0	100.0	4.5	1.6
	3	87.7	96.1	100.0	67.4	90.3	100.0	4.4	1.6
Best heur	1	69.8	96.5	100.0	72.4	92.2	100.0	4.3	1.1
	2	69.8	96.3	100.0	69.5	89.9	100.0	4.3	1.1
	3	69.1	96.3	100.0	60.6	88.8	100.0	4.4	1.1

Table 8: Capacity utilization rates and average number of new/closed facilities.

In summary, our numerical results indicate that the proposed constructive heuristics along with the local improvements provide good quality solutions in significantly shorter CPU times than CPLEX. In addition, for 19% of the test instances, the heuristics even achieve higher solution quality.

6 Conclusions

In this article, we have considered a recently introduced multi-period facility location problem with features that go beyond the classical perspective. In particular, customers are differentiated by their service requirements in terms of delivery times. While one demand class includes customers whose orders must be met on time, the second class of customers tolerates delayed deliveries as long as lateness does not exceed a given threshold. In addition to location and allocation decisions, capacity sizing decisions are also addressed, with their associated fixed investment costs and variable processing costs being subject to economies of scale.

We have proposed the first two-phase heuristic procedures for this new problem. During the constructive phase, special focus is given to the identification of a suitable set of initially existing facilities and new locations that are operated over the whole planning horizon. For the facility choices made, customer demands are served according to a heuristic scheme. In the improvement phase, adjustments in the opening and closing schedule of the pre-selected facilities are made in an attempt to obtain solutions with a lower total cost. The computational results show that the mechanisms developed for the constructive and improvement heuristics are effective both in terms of computational efficiency and solution quality. Compared to a state-of-the-art MILP solver, all constructive procedures identify a feasible solution to every instance in significantly shorter computing times. Furthermore, at the end of the first phase, the average heuristic objective value is 2.49% more costly than the objective value of the best solution provided by the solver. This rate decreases to 0.72% after using the local improvement schemes. This quality enhancement is achieved with low computational effort. At the end of the two phases, solutions with higher quality than those found by the solver are available for almost one-fifth of the instances. These findings suggest that the proposed heuristics could be integrated into a decision support framework, enabling a 'what-if' analysis to be performed within reasonable computing time. Such an analysis is important to help decision makers gain deeper insight into the impact of delays in demand fulfillment and the trade-offs achieved from location and capacity sizing decisions. Additionally, the heuristic procedures provide a useful alternative to a commercial solver, especially for large-sized instances.

A future line of research would be the extension of the heuristic algorithms to deal with single-assignment conditions for serving customers. Moreover, since some input parameters for long-term planning are inherently uncertain (e.g. demand), focus could also be given to extending the heuristics to explicitly account for the uncertainty associated with future conditions.

Acknowledgements

This work was partially supported by FCT - Fundação para a Ciência e a Tecnologia (Portuguese Foundation for Science and Technology) through the project UID/MAT/00297/2019 (CMA/FCT/UNL). This support is gratefully acknowledged.

References

- A. Ahmadi-Javid, P. Seyedi P, and S.S. Syam. A survey of healthcare facility location. *Computers & Operations Research*, 79:223–263, 2017.
- M. Albareda-Sambola, E. Fernández, and S. Nickel. Multiperiod location-routing with decoupled time scales. *European Journal of Operational Research*, 217:248–258, 2012.
- E.M. Alvarez, M.C. van der Heijden, and W.H.M. Zijm. Service differentiation in spare parts supply through dedicated stocks. *Annals of Operations Research*, 231:283–303, 2015.
- A. Amiri. Designing a distribution network in a supply chain system: Formulation and efficient solution procedure. *European Journal of Operational Research*, 171:567–576, 2006.
- H. Amrani, A. Martel, N. Zufferey, and P. Makeeva. A variable neighborhood search heuristic for the design of multicommodity production-distribution networks with alternative facility configurations. *OR Spectrum*, 33:989–1007, 2011.
- A.B. Arabani and R.Z. Farahani. Facility location dynamics: An overview of classifications and applications. *Computers & Industrial Engineering*, 62:408–420, 2012.
- M.A. Arostegui Jr., S.N. Kadipasaoglu, and B.M. Khumawala. An empirical comparison of Tabu Search, Simulated Annealing and Genetic Algorithms for facilities location problems. *International Journal of Production Economics*, 103:742–754, 2006.
- H. Badri, M. Bashiri, and T.H. Hejazi. Integrated strategic and tactical planning in a supply chain network design with a heuristic solution method. *Computers & Operations Research*, 40:1143–1154, 2013.
- M. Bashiri, H. Badri, and J. Talebi. A new approach to tactical and strategic planning in production-distribution networks. *Applied Mathematical Modelling*, 36:1703–1717, 2012.

- M.K. Boujelben, C. Gicquel, and M. Minoux. A MILP model and heuristic approach for facility location under multiple operational constraints. *Computers & Industrial Engineering*, 98: 446–461, 2016.
- J. Castro, S. Nasini, and F. Saldanha-da-Gama. A cutting-plane approach for large-scale capacitated multi-period facility location using a specialized interior-point method. *Mathematical Programming*, 163:411–444, 2017.
- I. Correia and M.E. Captivo. Bounds for the single source modular capacitated plant location problem. *Computers & Operations Research*, 33:2991–3003, 2006.
- I. Correia and T. Melo. Multi-period capacitated facility location under delayed demand satisfaction. *European Journal of Operational Research*, 255:729–746, 2016.
- I. Correia, T. Melo, and F. Saldanha-da-Gama. Comparing classical performance measures for a multi-period, two-echelon supply chain network design problem with sizing decisions. *Computers & Industrial Engineering*, 64:366–380, 2013.
- J. Dias, M.E. Captivo, and J. Clímaco. A memetic algorithm for multi-objective dynamic location problems. *Journal of Global Optimization*, 42:221–253, 2008.
- S. Duran, T. Liu, D. Simchi-Levi, and J.L. Swann. Policies utilizing tactical inventory for service-differentiated customers. *Operations Research Letters*, 36:259–264, 2008.
- S. Elhedhli and F. Gzara. Integrated design of supply chain networks with three echelons, multiple commodities and technology selection. *IIE Transactions*, 40:31–44, 2008.
- P. Escalona, F. Ordóñez, and V. Marianov. Joint location-inventory problem with differentiated service levels using critical level policy. *Transportation Research Part E*, 83:141–157, 2015.
- M. Fattahi, M. Mahootchi, K. Govindan, and S.M.M. Husseini. Dynamic supply chain network design with capacity planning and multi-period pricing. *Transportation Research Part E*, 81: 169–202, 2015.
- B. Fortz. Location problems in telecommunications. In G. Laporte, S. Nickel, and F. Saldanha da Gama, editors, *Location Science*, chapter 20, pages 537–554. Springer, Heidelberg, 2015.

- H.-C. Hung, E.P. Chew, L.H. Lee, and S. Liu. Dynamic inventory rationing for systems with multiple demand classes and general demand processes. *International Journal of Production Economics*, 139:351–358, 2012.
- C.A. Irawan and D. Jones. Formulation and solution of a two-stage capacitated facility location problem with multilevel capacities. *Annals of Operations Research*, 272:41–67, 2019.
- S.K. Jacobsen. Multiperiod capacitated location models. In P.B. Mirchandani and R.L. Francis, editors, *Discrete Location Theory*, chapter 4. Wiley, New York, 1990.
- A. Klose and A. Drexl. Facility location models for distribution system design. *European Journal of Operational Research*, 162:4–29, 2005.
- Z. Li, Q. Lu, and M. Talebian. Online versus bricks-and-mortar retailing: a comparison of price, assortment and delivery time. *International Journal of Production Research*, 53:3823–3835, 2015.
- C.L. Martins, M.T. Melo, and M.V. Pato. Redesigning a food bank supply chain network in a triple bottom line context. *International Journal of Production Economics*, 2018. doi: 10.1016/j.ijpe.2018.11.011. *In press*.
- M. Marufuzzaman, R. Gedik, and M.S. Roni. A Benders based rolling horizon algorithm for a dynamic facility location problem. *Computers & Industrial Engineering*, 98:462–469, 2016.
- M.T. Melo, S. Nickel, and F. Saldanha-da-Gama. Facility location and supply chain management: a review. *European Journal of Operational Research*, 196:401–412, 2009.
- M.T. Melo, S. Nickel, and F. Saldanha-da-Gama. A tabu search heuristic for redesigning a multi-echelon supply chain network over a planning period. *International Journal of Production Economics*, 136:218–230, 2012.
- S. Nickel and F. Saldanha da Gama. Multi-period facility location. In G. Laporte, S. Nickel, and F. Saldanha da Gama, editors, *Location Science*, chapter 11, pages 289–310. Springer, Heidelberg, 2015.
- S.H. Owen and M.S. Daskin. Strategic facility location: A review. *European Journal of Operational Research*, 111:423–447, 1998.

- C. Prodhon and C. Prins. A survey of recent research on location-routing problems. *European Journal of Operational Research*, 238:1–17, 2014.
- A. Shulman. An algorithm for solving dynamic capacitated plant location problems with discrete expansion sizes. *Operations Research*, 39:423–436, 1991.
- J.E. Torres-Soto and H. Üster. Dynamic-demand capacitated facility location problems with and without relocation. *International Journal of Production Research*, 49:3979–4005, 2011.
- Y. Wang, M.A. Cohen, and Y.-S. Zheng. Differentiating customer service on the basis of delivery lead-times. *IIE Transactions*, 34:979–489, 2002.
- Z. Wu and J. Wu. Price discount and capacity planning under demand postponement with opaque selling. *Decision Support Systems*, 76:24–34, 2015.

Die PDF-Dateien der folgenden Berichte sind verfügbar unter:

The PDF files of the following reports are available under:

<http://www.htwsaar.de/wiwi>

1 I. Correia, T. Melo, F. Saldanha da Gama

Comparing classical performance measures for a multi-period, two-echelon supply chain network design problem with sizing decisions

Keywords: supply chain network design, facility location, capacity acquisition, profit maximization, cost minimization

(43 pages, 2012)

2 T. Melo

A note on challenges and opportunities for Operations Research in hospital logistics

Keywords: Hospital logistics, Operations Research, application areas

(13 pages, 2012)

3 S. Hütter, A. Steinhaus

Forschung an Fachhochschulen – Treiber für Innovation im Mittelstand: Ergebnisse der Qbing-Trendumfrage 2013

Keywords: Innovation, Umfrage, Trendbarometer, Logistik-Konzepte, Logistik-Technologien, Mittelstand, KMU

(5 pages, 2012)

4 A. Steinhaus, S. Hütter

Leitfaden zur Implementierung von RFID in kleinen und mittelständischen Unternehmen

Keywords: RFID, KMU, schlanke Prozesse, Prozessoptimierung, Produktion, Forschungsgruppe Qbing

(49 pages, 2013)

5 S.A. Alumur, B.Y. Kara, M.T. Melo

Location and Logistics

Keywords: forward logistics network design, reverse logistics network design, models, applications

(26 pages, 2013)

6 S. Hütter, A. Steinhaus

Forschung an Fachhochschulen – Treiber für Innovation im Mittelstand: Ergebnisse der Qbing-Trendumfrage 2013

Keywords: Innovation, Umfrage, Trendbarometer, Logistik-Konzepte, Logistik-Technologien, Mittelstand, KMU

(6 pages, 2014)

7 M.J. Cortinhal, M.J. Lopes, M.T. Melo

Redesigning a three-echelon logistics network over multiple time periods with transportation mode selection and outsourcing opportunities

Keywords: logistics network design/re-design, multiple periods, transportation mode selection, product outsourcing, mixed-integer linear programming

(49 pages, 2014)

8 T. Bousonville, C. Ebert, J. Rath

A comparison of reward systems for truck drivers based on telematics data and driving behavior assessments

Keywords: Telematics, driving behavior, incentives, award systems

(9 pages, 2015)

9 I. Correia, T. Melo

Multi-period capacitated facility location under delayed demand satisfaction

Keywords: Location, multi-period, capacity choice, delivery lateness, MILP models

(35 pages, 2015)

10 C.L. Martins, M.T. Melo, M.V. Pato

Redesigning a food bank supply chain network, Part I: Background and mathematical formulation

Keywords: Supply chain, sustainability, tri-objective problem, MILP model

(30 pages, 2016)

11 I. Correia, T. Melo

A computational comparison of formulations for a multi-period facility location problem with modular capacity adjustments and flexible demand fulfillment

Keywords: Facility location, multi-period, capacity expansion and contraction, delivery lateness, mixed-integer linear models

(42 pages, 2016)

12 A. Bernhardt, T. Melo, T. Bousonville, H. Kopfer

Scheduling of driver activities with multiple soft time windows considering European regulations on rest periods and breaks

Keywords: road transportation, driver scheduling, rest periods, breaks, driving hours, Regulation (EC) No 561/2006, mixed integer linear programming models

(41 pages, 2019)

(137 pages, 2016)

13 C.L. Martins, M.T. Melo,
M.V. Pato

Redesigning a food bank supply chain network, Part II: Computational study

Keywords: Food rescue and delivery, sustainability, supply chain network design, tri-objective problem, social impact, economic and environmental performance

(57 pages, 2017)

14 A. Bernhardt, T. Melo,
T. Bousonville, H. Kopfer

Truck driver scheduling with combined planning of rest periods, breaks and vehicle refueling

Keywords: road transportation, refueling, fuel cost, driver scheduling, rest periods, breaks, driving hours, Regulation (EC) No 561/2006, mixed integer linear programming

(90 pages, 2017)

15 M.J. Cortinhal, M.J. Lopes,
M.T. Melo

Impact of partial product outsourcing, transportation mode selection, and single-assignment requirements on the design of a multi-stage supply chain network

Keywords: supply chain network design, facility location, supplier selection, in-house production, product outsourcing, transportation mode selection, single-assignment, mixed-integer linear programming

(51 pages, 2018)

16 C. Sauvey, T. Melo, I. Correia

Two-phase heuristics for a multi-period capacitated facility location problem with service-differentiated customers

Keywords: facility location, multi-period, delivery lateness, constructive heuristics, local improvements

Hochschule für Technik und Wirtschaft des Saarlandes

Die Hochschule für Technik und Wirtschaft des Saarlandes (htw saar) wurde im Jahre 1971 als saarländische Fachhochschule gegründet. Insgesamt studieren rund 6000 Studentinnen und Studenten in 46 verschiedenen Studiengängen an der htw saar, aufgeteilt auf vier Fakultäten.

In den vergangenen zwanzig Jahren hat die Logistik immens an Bedeutung gewonnen. Die htw saar hat dieser Entwicklung frühzeitig Rechnung getragen und einschlägige Studienprogramme sowie signifikante Forschungs- und Technologietransferaktivitäten entwickelt. Die Veröffentlichung der Schriftenreihe Logistik soll die Ergebnisse aus Forschung und Projektpraxis der Öffentlichkeit zugänglich machen.

Weitere Informationen finden Sie unter <http://logistik.htwsaar.de>



Institut für Supply Chain und Operations Management

Das Institut für Supply Chain und Operations Management (ISCOM) der htw saar ist auf die Anwendung quantitativer Methoden in der Logistik und deren Implementierung in IT-Systemen spezialisiert. Neben öffentlich geförderten Forschungsprojekten zu innovativen Themen arbeitet ISCOM eng mit Projektpartnern aus der Wirtschaft zusammen, wodurch der Wissens- und Technologietransfer in die Praxis gewährleistet wird. Zu den Arbeitsgebieten zählen unter anderem Distributions- und Transportplanung, Supply Chain Design, Bestandsmanagement in Supply Chains, Materialflussanalyse und -gestaltung sowie Revenue Management.

Weitere Informationen finden Sie unter <http://iscom.htwsaar.de>



Forschungsgruppe Qbing

Qbing ist eine Forschungsgruppe an der Hochschule für Technik und Wirtschaft des Saarlandes, die spezialisiert ist auf interdisziplinäre Projekte in den Bereichen Produktion, Logistik und Technologie. Ein Team aus derzeit acht Ingenieuren und Logistikexperten arbeitet unter der wissenschaftlichen Leitung von Prof. Dr. Steffen Hütter sowohl in öffentlich geförderten Projekten als auch zusammen mit Industriepartnern an aktuellen Fragestellungen zur Optimierung von logistischen Prozessabläufen in Handel und Industrie unter Einbeziehung modernster Sensortechnologie und Telemetrie. Qbing hat auch und gerade auf dem Gebiet der angewandten Forschung Erfahrung in der Zusammenarbeit mit kleinen und mittelständischen Unternehmen.

Weitere Informationen finden Sie unter <http://www.qbing.de>

ISSN 2193-7761