Available online at www.sciencedirect.com
ScienceDirect
Computers & Operations Research 35 (2008) 373–391
computers & operations research
www.elsevier.com/locate/cor

# Dynamic supply chain design with inventory

Y. Hinojosa[a,*], J. Kalcsics[b], S. Nickel[c], J. Puerto[d], S. Velten[e]

[a] *Departamento de Economía Aplicada I, Universidad de Sevilla, Spain*
[b] *Universität des Saarlandes, Germany*
[c] *Universität des Saarlandes, Germany*
[d] *Departamento de Estadística e I.O., Universidad de Sevilla, Spain*
[e] *Universität des Saarlandes, Germany*

Available online 6 October 2006

Dedicated to the memory of Charles ReVelle

**Abstract**

In this paper, we deal with a facility location problem where we build new facilities or close down already existing facilities at two different distribution levels over a given time horizon. In addition, we allow to carry over stock in warehouses between consecutive periods. Our model intends to minimize the total costs, including transportation and inventory holding costs for products as well as fixed and operating costs for facilities.

After formulating the problem, we propose a Lagrangian approach which relaxes the constraints connecting the distribution levels. A procedure is developed to solve the resulting, independent subproblems and, based on this solution, to construct a feasible solution for the original problem.


*Keywords:* Dynamic multi-echelon facility-location; Inventory; Integer programming; Lagrangian dual; Heuristic

## 1. Introduction

Discrete location problems are an important group of problems within operational research. Especially in the context of strategic supply chain management, location problems experience more and more attention (see [1–3]).

A supply chain network comprises a number of facilities (e.g., manufacturing plants, distribution centers, warehouses, etc.) that perform a set of operations ranging from the acquisition of raw materials, the transformation of these materials into intermediate and finished products, to the distribution of the finished goods to the customers. Fig. 1 shows an example of a network with suppliers, plants, warehouses, and customers. Arrows indicate that products are shipped between two facilities in a particular time period. Note that operating facilities can change from one time period to another. In the following, we will concentrate on the distribution part (without procurement); therefore, we deal with a two-echelon network structure. The optimization of the complete logistics network is accomplished through efficient planning decisions. Strategic decisions, on the one hand, include facility location, among others, and have a long-lasting

* Corresponding author. Fax: +34 954 551 636.
*E-mail address:* yhinojos@us.es (Y. Hinojosa).

0305-0548/$ - see front matter 
doi:10.1016/j.cor.2006.03.017

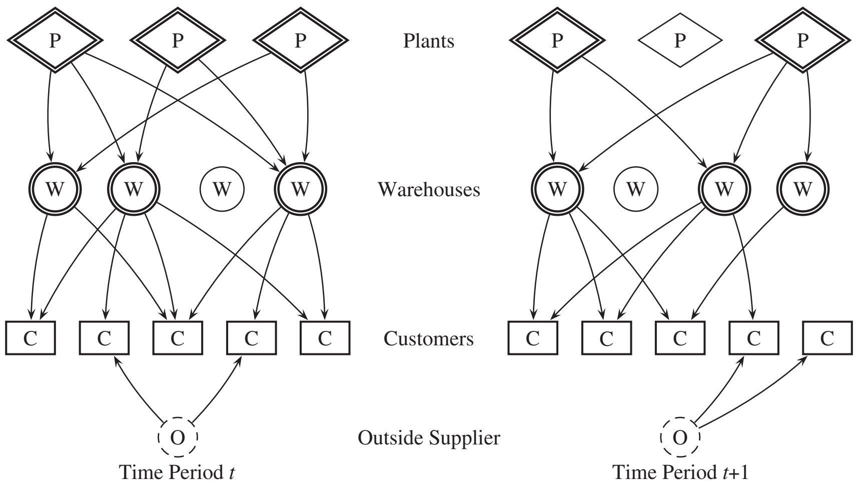


Fig. 1. Example of a logistics network.

effect on a company. On the other hand, the transportation pattern to be followed in each time period is considered as a tactical decision.

Most location models deal with the redesign of supply chain networks by deciding which existing facilities should be closed and where new facilities should be established. However, during this redesign process, one is often faced with the problem of how to transform an initial supply chain structure into a new one. For example, a company wishes to adapt the locations of its warehouses throughout Europe to meet changes in the customer behavior. This process has to be completed in five years, where the initial and the desired final state are known, and a time-dependent dynamic solution which transforms one into the other is sought. Hence, multi-period models have to be employed in order to address these questions about the redesign process. To cope with these types of problems, several approaches have already been proposed in the literature (see [4–7]). However, some typical features of supply chain networks, such as a multi-echelon structure or capacity and inventory aspects, have been considered only partially in the literature until now (see [5,6,8–13]). A recent modeling paper that covers most of the above-mentioned issues but that does not go into algorithmic considerations is Melo et al. [14].

In this paper, we investigate a dynamic two-echelon multi-commodity location model where potential new facilities can be opened and existing facilities can be closed. Assuming a high fixed cost for establishing and closing a facility, we do not allow a facility which has been closed once to be reopened and consequently, a facility opened during the planning horizon cannot be closed again. This model is an extension of the problem considered in Hinojosa et al. [15], where a multi-period distribution systems of perishable goods has been modeled and no outsourcing has been allowed to cover demand. Under these hypotheses, items do not carry over to consecutive time periods. However, when modeling distribution systems for non-perishable goods, inventories are an essential decision aspect. Moreover, it is unrealistic not to allow outsourcing, because in any distribution system it is nearly always possible to buy products from outside suppliers if capacities are insufficient. Note that this feature also avoids the infeasibility of some patterns of demand that may occur in the previous model. The integration of these two characteristics, inventories and outsourcing, makes the model more general, respectively more realistic, although the mathematical treatment is more difficult. To the best of our knowledge, this model has not been addressed in the literature yet.

The goal of this problem is to minimize the total cost of designing the supply chain network and of the distribution activities in order to fulfill the customer demands. The problem is modeled as a mixed-integer linear program. However, since approaches dealing directly with such formulations lead to extensive computation times, we propose the following alternative solution approach. First, we employ a Lagrangian Relaxation scheme incorporating a dual ascent method to obtain a lower bound on the optimal objective value (see [16–21] for applications of this method in different contexts). Afterward, based on the solution of the relaxed problem, we construct a heuristic solution, and hence an upper bound, for the problem. At last, this upper bound is improved using an interchange heuristic. Although we address a more general problem than in [15], we obtain solutions of similar quality.

The remaining paper is organized as follows. In Section 2, a mathematical formulation of the problem is presented. In Section 3, we introduce a Lagrangian Relaxation for this model. Section 4 contains the procedure to construct heuristic solutions based on the relaxed ones; moreover, the Interchange method is detailed. Computational results are presented in Section 5. The paper ends with some conclusions, an outlook to future research, and an Appendix where some technical results are included.

## 2. The model

We deal with a dynamic two-echelon multi-commodity capacitated facility location problem with inventory and outsourcing. The objective is to minimize the total cost for meeting demands of different products specified over different time periods at various customer locations. The version of the problem considered here makes the following assumptions. The planning horizon consists of a set $\mathcal{T}=\{1,\dots,T\}$ of different time periods indexed by $t\in\mathcal{T}$, where $|\mathcal{T}|=T$. For example, seasons or months are typical period lengths for this kind of problem. The sets of customers and commodities, together with the feasible locations for the facilities (plants and warehouses) are considered to be fixed and known beforehand. That is, they do not change over the planning horizon. These assumptions lead to the following index sets:

$\mathcal{LC}$: set of customer locations, indexed by $i\in\mathcal{LC}$,
$\mathcal{LW}$: set of warehouse locations, indexed by $j\in\mathcal{LW}$,
$\mathcal{LP}$: set of plant locations, indexed by $k\in\mathcal{LP}$,
$\mathcal{P}$: set of different product types (commodities), indexed by $p\in\mathcal{P}$.

Moreover, we assume that at the beginning of the first time period there exists a subset $\mathcal{LP}_{\rm c}(\mathcal{LW}_{\rm c})$ of the whole set of possible plant locations (warehouse locations) where facilities are already in operation. These facilities can be closed at the end of any time period $t\in\mathcal{T}$, but once closed they cannot be reopened again. We denote $\mathcal{LP}_{\rm o}(\mathcal{LW}_{\rm o})$ the set of possible plant locations (warehouse locations) where no operating facilities are yet established. At these locations, facilities can be opened at the beginning of each time period, but it is not allowed to close such facilities again. This hypothesis is quite reasonable as in real-life applications the opening or closing of facilities involves large investments. Another possibility would be to rent those facilities. However, this issue would lead to a different model that is not in the scope of this paper. (The reader interested in such a model is referred to [14].)

Additionally, we assume that a minimum number of plants and warehouses must be in operation at the beginning and at the end of the planning horizon, assuring a minimum coverage of the demand and some presence in the market. Therefore, we denote $NW^1$ and $NW^{\rm T}$ ($NP^1$ and $NP^{\rm T}$) the minimum number of warehouses (plants) which have to be in operation at the beginning of the first and at the end of the last time period. Note that this assumption does not represent a limitation of the model as $NW^1$, $NW^{\rm T}$, $NP^1$ and $NP^{\rm T}$ could be equal to zero.

As we consider the situation where plants as well as warehouses have limited capacity, each depending on the time period, we denote:

$WC_j^t$ capacity of warehouse $j$ in time period $t$,
$PC_k^t$ capacity of plant $k$ in time period $t$.
Furthermore, the customer demand is denoted:

$D_{ip}^t$ demand of product $p$ at customer $i$ during time period $t$.

As we want to minimize the total cost for meeting the customer demand, we have to define a cost structure that includes maintenance, opening, and closing costs for plants and warehouses as well as production, transportation, and inventory holding costs for the products. Maintenance costs include all costs related to operational costs as for instance, ageing of facilities, taxes, etc. For this purpose we use the following notation, where all costs refer to present values:

For $j\in\mathcal{LW}_{\rm o}$ and $t\in\mathcal{T}$:

$TCW_j^t$ total cost of warehouse $j$ being established at the beginning of time period $t$.

These costs include opening costs at the beginning of time period $t$ and maintenance costs from time period $t$ to time period $T$.

For $j\in\mathcal{LW}_{\rm c}$ and $t\in\mathcal{T}\setminus\{T\}$:

$TCW_j^t$ total cost of warehouse $j$ being removed at the end of time period $t$.

These costs include closing costs at the end of time period $t$ and maintenance costs from time period 1 to the end of time period $t$.

For $j \in \mathscr{LW}_c$:

$TCW_j^T$ total cost of warehouse $j$ open during the whole planning horizon.

$TCP_k^t$ is defined analogously for the set of possible plant locations $\mathscr{LP}$.

Furthermore, we have

$PTC_{jkp}^t$ production and transportation cost per unit of product $p$ from plant $k$ to warehouse $j$ in time period $t$.
$TC_{ijp}^t$ transportation cost per unit of product $p$ from warehouse $j$ to customer $i$ in time period $t$.
$IC_{jp}^t$ unit inventory holding cost of product $p$ at warehouse $j$ from time period $t$ to time period $t+1$.
$OSC_{ip}^t$ transportation cost per unit of product $p$ to customer $i$ served from an outside supplier.
According to this cost structure, we define the following decision variables:

*Binary variables (configuration decision)*

For $j \in \mathscr{LW}_o$ and $t \in \mathscr{T}$:

$$z_j^t = \begin{cases} 1 & \text{if warehouse } j \text{ is opened at the beginning of time period } t, \\ 0 & \text{otherwise.} \end{cases}$$

For $j \in \mathscr{LW}_c$ and $t \in \mathscr{T} \setminus \{T\}$:

$$z_j^t = \begin{cases} 1 & \text{if existing warehouse } j \text{ is closed at the end of time period } t, \\ 0 & \text{otherwise.} \end{cases}$$

For $j \in \mathscr{LW}_c$:

$$z_j^T = \begin{cases} 1 & \text{if existing warehouse } j \text{ is open during the whole planning horizon,} \\ 0 & \text{otherwise.} \end{cases}$$

$\zeta_k^t$ is defined analogously for the set of possible plant locations $\mathscr{LP}$.

*Continuous variables (tactical decision)*

$x_{ijp}^t$ fraction (with resp. to $D_{ip}^t$) of product $p$ delivered to customer $i$ from warehouse $j$ in time period $t$,
$y_{jkp}^t$ fraction (with resp. to $WC_j^t$) of product $p$ delivered to warehouse $j$ from plant $k$ in time period $t$,
$o_{ip}^t$ fraction (with resp. to $D_{ip}^t$) of product $p$ delivered to customer $i$ from an outside supplier in time period $t$,
$I_{jp}^t$ inventory holding of product $p$ at warehouse $j$ at the end of time period $t$.
At last, we introduce the following index sets to simplify the notation:

$$T_{jt} := \begin{cases} \{1,\dots,t\} & \text{if } j \in \mathscr{LW}_o, \\ \{t,\dots,T\} & \text{if } j \in \mathscr{LW}_c, \end{cases} \quad \text{and} \quad T_{kt} := \begin{cases} \{1,\dots,t\} & \text{if } k \in \mathscr{LP}_o, \\ \{t,\dots,T\} & \text{if } k \in \mathscr{LP}_c. \end{cases}$$

Next, we state the mathematical formulation for the dynamic two-echelon multi-commodity capacitated location problem with inventory and outsourcing ($D2ELI$):

$$\begin{aligned}(D2ELI) \quad \operatorname{Min} f(x,y,o,I,z,\zeta) \\ := \sum_{t\in\mathscr{T}} \sum_{i\in\mathscr{LC}} \sum_{j\in\mathscr{LW}} \sum_{p\in\mathscr{P}} TC_{ijp}^t x_{ijp}^t D_{ip}^t + \sum_{t\in\mathscr{T}} \sum_{j\in\mathscr{LW}} \sum_{k\in\mathscr{LP}} \sum_{p\in\mathscr{P}} PTC_{jkp}^t y_{jkp}^t WC_j^t \\ + \sum_{t\in\mathscr{T}} \sum_{i\in\mathscr{LC}} \sum_{p\in\mathscr{P}} OSC_{ip}^t o_{ip}^t D_{ip}^t + \sum_{t\in\mathscr{T}} \sum_{j\in\mathscr{LW}} \sum_{p\in\mathscr{P}} IC_{jp}^t I_{jp}^t \\ + \sum_{t\in\mathscr{T}} \sum_{j\in\mathscr{LW}} TCW_j^t z_j^t + \sum_{t\in\mathscr{T}} \sum_{k\in\mathscr{LP}} TCP_k^t \zeta_k^t\end{aligned}$$

$$\text{s.t.} \sum_{j\in\mathscr{LW}} x^t_{ijp} + o^t_{ip} \geqslant 1 \quad \forall i \in \mathscr{LC}, \forall p \in \mathscr{P}, \forall t \in \mathscr{T}, \tag{1}$$

$$\sum_{i\in\mathscr{LC}}\sum_{p\in\mathscr{P}} D^t_{ip}x^t_{ijp} + \sum_{p\in\mathscr{P}} I^t_{jp} \leqslant WC^t_j \sum_{r\in T_{jt}} z^r_j \quad \forall j \in \mathscr{LW}, \forall t \in \mathscr{T}, \tag{2}$$

$$\sum_{p\in\mathscr{P}} I^t_{jp} \leqslant WC^{t+1}_j \sum_{r\in T_{jt}} z^r_j \quad \forall j \in \mathscr{LW}, \forall t \in \mathscr{T}\backslash\{T\}, \tag{3}$$

$$\sum_{k\in\mathscr{LP}} WC^t_j y^t_{jkp} + I^{t-1}_{jp} = \sum_{i\in\mathscr{LC}} D^t_{ip}x^t_{ijp} + I^t_{jp} \quad \forall j \in \mathscr{LW}, \forall p \in \mathscr{P}, \forall t \in \mathscr{T}, \tag{4}$$

$$\sum_{j\in\mathscr{LW}}\sum_{p\in\mathscr{P}} WC^t_j y^t_{jkp} \leqslant PC^t_k \sum_{r\in T_{kt}} \zeta^r_k \quad \forall k \in \mathscr{LP}, \ \forall t \in \mathscr{T}, \tag{5}$$

$$\sum_{j\in\mathscr{LW}_{\mathrm{o}}} z^1_j + \sum_{j\in\mathscr{LW}_{\mathrm{c}}}\sum_{t\in\mathscr{T}} z^t_j \geqslant NW^1 \quad \sum_{j\in\mathscr{LW}_{\mathrm{o}}}\sum_{t\in\mathscr{T}} z^t_j + \sum_{j\in\mathscr{LW}_{\mathrm{c}}} z^t_j \geqslant NW^T, \tag{6}$$

$$\sum_{k\in\mathscr{LP}_{\mathrm{o}}} \zeta^1_k + \sum_{k\in\mathscr{LP}_{\mathrm{c}}}\sum_{t\in\mathscr{T}} \zeta^t_k \geqslant NP^1 \quad \sum_{k\in\mathscr{LP}_{\mathrm{o}}}\sum_{t\in\mathscr{T}} \zeta^t_k + \sum_{k\in\mathscr{LP}_{\mathrm{c}}} \zeta^t_k \geqslant NP^T, \tag{7}$$

$$\sum_{t\in\mathscr{T}} z^t_j = 1 \ \forall j \in \mathscr{LW}_{\mathrm{c}} \quad \sum_{t\in\mathscr{T}} z^t_j \leqslant 1 \ \forall j \in \mathscr{LW}_{\mathrm{o}}, \tag{8}$$

$$\sum_{t\in\mathscr{T}} \zeta^t_k = 1 \ \forall k \in \mathscr{LP}_{\mathrm{c}} \quad \sum_{t\in\mathscr{T}} \zeta^t_k \leqslant 1 \ \forall k \in \mathscr{LP}_{\mathrm{o}}, \tag{9}$$

$$I^0_{jp} = 0, \quad I^T_{jp} = 0, \quad I^t_{jp} \geqslant 0 \quad \forall j \in \mathscr{LW} \ \forall p \in \mathscr{P}, \ \forall t \in \mathscr{T}\backslash\{T\}, \tag{10}$$

$$0 \leqslant x^t_{ijp}, y^t_{jkp}, o^t_{ip} \leqslant 1 \quad \forall i \in \mathscr{LC} \ \forall j \in \mathscr{LW}, \ \forall k \in \mathscr{LP} \ \forall p \in \mathscr{P} \ \forall t \in \mathscr{T}, \tag{11}$$

$$z^t_j, \zeta^t_k \in \{0, 1\} \quad \forall j \in \mathscr{LW} \ \forall k \in \mathscr{LP} \ \forall t \in \mathscr{T}. \tag{12}$$

The objective function minimizes the total costs for meeting the customer demands. The first Constraints (1) assure that the demand for each commodity of each customer is satisfied in all time periods (either from the own network or from external suppliers). Constraints (2) represent the fact that products can be delivered to customers only from open warehouses. Furthermore, they ensure that the amount of products handled at a warehouse does not exceed its maximal capacity. In addition, it should not happen that the amount of products which are stored at a warehouse at the end of time period $t$ is greater than the capacity of that warehouse in the following time period $t+1$. This is ensured by the Constraints (3).

The following Constraints (4) are flow conservation constraints. They assure that the amount of product $p$ delivered to warehouse $j$ in time period $t$ plus the inventory of $p$ at $j$ from time period $t-1$ is equal to the amount of product $p$ delivered to customers in time period $t$ plus the inventory of product $p$ at $j$ at the end of time period $t$. The meaning of Constraints (5) for the plants is similar to that of (2) for the warehouses. The only difference is that we do not allow inventory holding at plants. With the next Constraints (6) and (7), we fulfill the required begin and end status for the number of open warehouses and plants. Constraints (8) (resp. (9)) reflect the special structure of $\mathscr{LW}_{\mathrm{o}}$ and $\mathscr{LW}_{\mathrm{c}}$ ($\mathscr{LP}_{\mathrm{o}}$ and $\mathscr{LP}_{\mathrm{c}}$). The last sets of Constraints (10)–(12) define the domains of the decision variables. Thereby, (10) additionally states that there is no inventory at the beginning and at the end of the planning horizon.

Observe that $(D2ELI)$ is a large mixed-integer programming problem including the uncapacitated facility location Problem (UFLP) as a special instance. Therefore, since the UFLP is $\mathscr{NP}$-hard (see [12]) one cannot expect to solve large instances of $(D2ELI)$ in polynomial time. For this reason, we will adopt a heuristic approach based on a Lagrangian Relaxation of the problem and an Interchange scheme. These concepts are developed in the following sections.

## 3. Decomposition of the problem: Lagrangian Relaxation

In this section, we consider a Lagrangian Relaxation of problem $(D2ELI)$ obtained by relaxing the constraints (1) and (4) into the objective function. This is done by associating non-negative multipliers $\mu^t_{ip} \geqslant 0$ to the constraints (1)

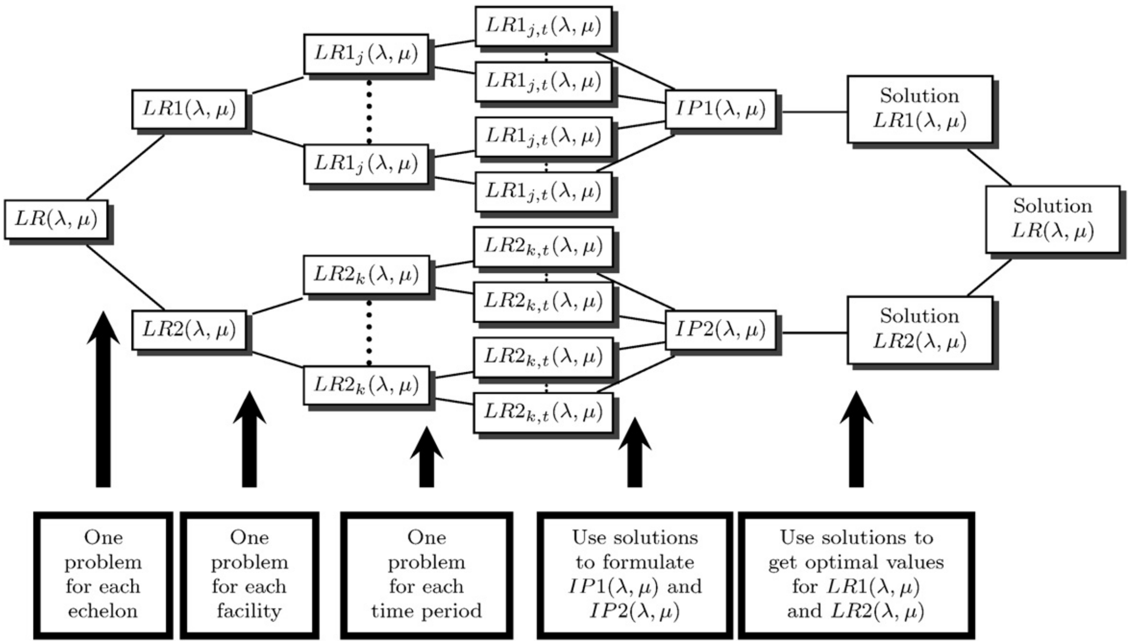


Fig. 2. Decomposition scheme for the relaxed problem.

and multipliers $\lambda_{jp}^t \in \mathbb{R}$ to the constraints (4). The relaxed problem, denoted by $LR(\lambda, \mu)$, is then given by

$$
\begin{aligned}
\operatorname{Min} & f_{(\lambda,\mu)}(x, y, o, I, z, \zeta) \\
:= & \sum_{t\in\mathscr{T}} \sum_{i\in\mathscr{LC}} \sum_{j\in\mathscr{LW}} \sum_{p\in\mathscr{P}} TC_{ijp}^t x_{ijp}^t D_{ip}^t + \sum_{t\in\mathscr{T}} \sum_{j\in\mathscr{LW}} \sum_{k\in\mathscr{LP}} \sum_{p\in\mathscr{P}} PTC_{jkp}^t y_{jkp}^t WC_j^t \\
& + \sum_{t\in\mathscr{T}} \sum_{i\in\mathscr{LC}} \sum_{p\in\mathscr{P}} OSC_{ip}^t o_{ip}^t D_{ip}^t + \sum_{t\in\mathscr{T}} \sum_{j\in\mathscr{LW}} \sum_{p\in\mathscr{P}} IC_{jp}^t I_{jp}^t \\
& + \sum_{t\in\mathscr{T}} \sum_{j\in\mathscr{LW}} TCW_j^t z_j^t + \sum_{t\in\mathscr{T}} \sum_{k\in\mathscr{LP}} TCP_k^t \zeta_k^t + \sum_{t\in\mathscr{T}} \sum_{i\in\mathscr{LC}} \sum_{p\in\mathscr{P}} \mu_{ip}^t \left(1 - \sum_{j\in\mathscr{LW}} x_{ijp}^t - o_{ip}^t\right) \\
& + \sum_{t\in\mathscr{T}} \sum_{j\in\mathscr{LW}} \sum_{p\in\mathscr{P}} \lambda_{jp}^t \left(\sum_{i\in\mathscr{LC}} D_{ip}^t x_{ijp}^t + I_{jp}^t - \sum_{k\in\mathscr{LP}} WC_j^t y_{jkp}^t - I_{jp}^{t-1}\right)
\end{aligned}
$$

s.t.(2), (3), (5), (6), (7), (8), (9), (10), (11), (12).

To solve $(D2ELI)$, which is still rather large, we use the following decomposition scheme. First, we decompose the problem by echelons. After that, the resulting subproblems are separated into one problem for each facility. Each of these problems can be further decomposed into one problem for each time period. A solution for $LR(\lambda, \mu)$ is then derived by combining the solutions of the subproblems. An overview of this procedure is given in Fig. 2.

According to the previous description, problem $LR(\lambda, \mu)$ can be separated into one subproblem per echelon, namely $LR1(\lambda, \mu)$ and $LR2(\lambda, \mu)$. Note that, due to $I_{jp}^0 = I_{jp}^T = 0$, we have

$$
\sum_{t\in\mathscr{T}} \sum_{j\in\mathscr{LW}} \sum_{p\in\mathscr{P}} (IC_{jp}^t + \lambda_{jp}^t) I_{jp}^t - \sum_{t\in\mathscr{T}} \sum_{j\in\mathscr{LW}} \sum_{p\in\mathscr{P}} \lambda_{jp}^t I_{jp}^{t-1} = \sum_{t=1}^{T-1} \sum_{j\in\mathscr{LW}} \sum_{p\in\mathscr{P}} (IC_{jp}^t + \lambda_{jp}^t - \lambda_{jp}^{t+1}) I_{jp}^t.
$$

Then, subproblems $LR1(\lambda, \mu)$ and $LR2(\lambda, \mu)$ are defined as follows:

$$LR1(\lambda,\mu) \quad \text{Min} \sum_{t\in\mathcal{T}} \sum_{i\in\mathcal{LC}} \sum_{j\in\mathcal{LW}} \sum_{p\in\mathcal{P}} (TC^t_{ijp} D^t_{ip} + \lambda^t_{jp} D^t_{ip} - \mu^t_{ip}) x^t_{ijp} + \sum_{t\in\mathcal{T}} \sum_{j\in\mathcal{LW}} TCW^t_j z^t_j$$
$$+ \sum_{t\in\mathcal{T}} \sum_{i\in\mathcal{LC}} \sum_{p\in\mathcal{P}} (OSC^t_{ip} D^t_{ip} - \mu^t_{ip}) o^t_{ip} + \sum_{t=1}^{T-1} \sum_{j\in\mathcal{LW}} \sum_{p\in\mathcal{P}} (IC^t_{jp} + \lambda^t_{jp} - \lambda^{t+1}_{jp}) I^t_{jp}$$
$$\text{s.t.(2), (3), (6), (8), (10)}, \quad 0 \leqslant x^t_{ijp}, \quad o^t_{ip} \leqslant 1, \quad z^t_j \in \{0,1\}$$

and

$$LR2(\lambda,\mu) \quad \text{Min} \sum_{t\in\mathcal{T}} \sum_{j\in\mathcal{LW}} \sum_{k\in\mathcal{LP}} \sum_{p\in\mathcal{P}} (PTC^t_{jkp} WC^t_j - \lambda^t_{jp} WC^t_j) y^t_{jkp} + \sum_{t\in\mathcal{T}} \sum_{k\in\mathcal{LP}} TCP^t_k \zeta^t_k$$
$$\text{s.t.(5), (7), (9)}, \quad 0 \leqslant y^t_{jkp} \leqslant 1, \quad \zeta^t_k \in \{0,1\}.$$

These problems can be solved independently and their solutions can be used to provide a solution for $LR(\lambda, \mu)$.

We will show in the next subsections how to solve the problems $LR1(\lambda, \mu)$ and $LR2(\lambda, \mu)$. Once these problems have been solved, the value of $LR(\lambda, \mu)$ is given by the following proposition, the proof of which is obvious.

**Proposition 1.** *Let $v(\cdot)$ be the optimal objective value of problem $(\cdot)$. Then*

$$v(LR(\lambda,\mu)) = v(LR1(\lambda,\mu)) + v(LR2(\lambda,\mu)) + \sum_{t\in\mathcal{T}} \sum_{i\in\mathcal{LC}} \sum_{p\in\mathcal{P}} \mu^t_{ip}.$$

### *3.1. Solution of $LR1(\lambda, \mu)$*

First of all, to solve $LR1(\lambda, \mu)$ we will leave the constraints (6) aside. Then $LR1(\lambda, \mu)$ can be separated into independent subproblems. However, once the solutions of these subproblems are obtained the constraints (6) might be violated. Therefore, we show at the end of this subsection how this solution can be transformed so that the constraints (6) are fulfilled in an optimal way.

Provided that (6) is removed, $LR1(\lambda, \mu)$ can be separated into $|\mathcal{LW}| + 1$ subproblems, one for each $j \in \mathcal{LW}$ plus one for the outsourcing aspect, as follows:

$$LR1_j(\lambda,\mu) \quad \text{Min} \sum_{t\in\mathcal{T}} \sum_{i\in\mathcal{LC}} \sum_{p\in\mathcal{P}} (TC^t_{ijp} D^t_{ip} + \lambda^t_{jp} D^t_{ip} - \mu^t_{ip}) x^t_{ijp} + \sum_{t\in\mathcal{T}} TCW^t_j z^t_j$$
$$+ \sum_{t=1}^{T-1} \sum_{p\in\mathcal{P}} (IC^t_{jp} + \lambda^t_{jp} - \lambda^{t+1}_{jp}) I^t_{jp}$$
$$\text{s.t.} \sum_{i\in\mathcal{LC}} \sum_{p\in\mathcal{P}} D^t_{ip} x^t_{ijp} + \sum_{p\in\mathcal{P}} I^t_{jp} \leqslant WC^t_j \sum_{r\in T_{jt}} z^r_j \quad \forall t \in \mathcal{T},$$
$$\sum_{p\in\mathcal{P}} I^t_{jp} \leqslant WC^{t+1}_j \sum_{r\in T_{jt}} z^r_t \quad \forall t \in \mathcal{T}\setminus\{T\},$$
$$\sum_{t\in\mathcal{T}} z^t_j = 1 \text{ if } j \in \mathcal{LW}_c \quad \text{or} \quad \sum_{t\in\mathcal{T}} z^t_j \leqslant 1 \text{ if } j \in \mathcal{LW}_o,$$
$$I^0_{jp} = 0, \quad I^T_{jp} = 0, \quad I^t_{jp} \geqslant 0 \quad \forall p \in \mathcal{P} \ \forall t \in \mathcal{T}\setminus\{T\}$$
$$0 \leqslant x^t_{ijp} \leqslant 1 \quad \forall i \in \mathcal{LC} \ \forall p \in \mathcal{P} \ \forall t \in \mathcal{T},$$
$$z^t_j \in \{0,1\} \quad \forall t \in \mathcal{T}.$$

Each of these problems is associated with a warehouse $j \in \mathcal{LW}$ and can be solved independently from one another. To cover the outsourcing part, we formulate the linear program

$$LR1_{\text{o}}(\mu) \quad \text{Min} \sum_{t \in \mathcal{T}} \sum_{i \in \mathcal{LC}} \sum_{p \in \mathcal{P}} (OSC^t_{ip} D^t_{ip} - \mu^t_{ip}) o^t_{ip}$$
$$\text{s.t.} 0 \leqslant o^t_{ip} \leqslant 1 \quad \forall i \in \mathcal{LC} \ \forall p \in \mathcal{P} \ \forall t \in \mathcal{T},$$

which can be solved by inspection:

$$o^t_{ip} = \begin{cases} 1 & \text{if } OSC^t_{ip} D^t_{ip} - \mu^t_{ip} < 0 \\ 0 & \text{otherwise} \end{cases} \quad \forall i \in \mathcal{LC} \ \forall p \in \mathcal{P} \ \forall t \in \mathcal{T}.$$

To solve $LR1_j(\lambda, \mu)$, we distinguish two cases depending on whether $j \in \mathcal{LW}_{\text{o}}$ or $j \in \mathcal{LW}_{\text{c}}$. First, we assume that $j \in \mathcal{LW}_{\text{o}}$. Then, we define for each $t_0 \in \mathcal{T}$ the following problem:

$$\begin{aligned} LR1_{jt_0}(\lambda, \mu) \quad \text{Min} & \sum_{t=t_0}^{T} \sum_{i \in \mathcal{LC}} \sum_{p \in \mathcal{P}} (TC^t_{ijp} D^t_{ip} + \lambda^t_{jp} D^t_{ip} - \mu^t_{ip}) x^t_{ijp} + TCW^{t_0}_j \\ & + \sum_{t=t_0}^{T-1} \sum_{p \in \mathcal{P}} (IC^t_{jp} + \lambda^t_{jp} - \lambda^{t+1}_{jp}) I^t_{jp} \\ \text{s.t.} & \sum_{i \in \mathcal{LC}} \sum_{p \in \mathcal{P}} D^t_{ip} x^t_{ijp} + \sum_{p \in \mathcal{P}} I^t_{jp} \leqslant WC^t_j \quad \forall t \geqslant t_0, \\ & \sum_{p \in \mathcal{P}} I^t_{jp} \leqslant WC^{t+1}_j \quad \forall t = t_0, \ldots, T-1, \\ & I^T_{jp} = 0, \quad I^t_{jp} \geqslant 0 \quad \forall p \in \mathcal{P} \ \forall t = t_0, \ldots, T-1, \\ & 0 \leqslant x^t_{ijp} \leqslant 1 \quad \forall i \in \mathcal{LC} \ \forall p \in \mathcal{P} \ \forall t \geqslant t_0. \end{aligned}$$

Now we assume that $j \in \mathcal{LW}_{\text{c}}$ and define for each $t_0 \in \mathcal{T}$ the following problem:

$$\begin{aligned} LR1_{jt_0}(\lambda, \mu) \quad \text{Min} & \sum_{t=1}^{t_0} \sum_{i \in \mathcal{LC}} \sum_{p \in \mathcal{P}} (TC^t_{ijp} D^t_{ip} + \lambda^t_{jp} D^t_{ip} - \mu^t_{ip}) x^t_{ijp} + TCW^{t_0}_j \\ & + \sum_{t=1}^{t_0-1} \sum_{p \in \mathcal{P}} (IC^t_{jp} + \lambda^t_{jp} - \lambda^{t+1}_{jp}) I^t_{jp} \\ \text{s.t.} & \sum_{i \in \mathcal{LC}} \sum_{p \in \mathcal{P}} D^t_{ip} x^t_{ijp} + \sum_{p \in \mathcal{P}} I^t_{jp} \leqslant WC^t_j \quad \forall t \leqslant t_0, \\ & \sum_{p \in \mathcal{P}} I^t_{jp} \leqslant WC^{t+1}_j \quad \forall t \leqslant t_0 - 1, \\ & I^{t_0}_{jp} = 0, \quad I^t_{jp} \geqslant 0 \quad \forall p \in \mathcal{P} \ \forall t \leqslant t_0 - 1, \\ & 0 \leqslant x^t_{ijp} \leqslant 1 \quad \forall i \in \mathcal{LC} \ \forall p \in \mathcal{P} \ \forall t \leqslant t_0. \end{aligned}$$

The solution of $LR1_j(\lambda, \mu)$ $(\forall j \in \mathcal{LW})$ is now given by the following result.

**Proposition 2.**

1. *If* $j \in \mathcal{LW}_{\text{o}}$, *then* $v(LR1_j(\lambda, \mu)) = \min\{\min_{1 \leqslant t_0 \leqslant T} v(LR1_{jt_0}(\lambda, \mu)), 0\}$.
2. *If* $j \in \mathcal{LW}_{\text{c}}$, *then* $v(LR1_j(\lambda, \mu)) = \min_{1 \leqslant t_0 \leqslant T} v(LR1_{jt_0}(\lambda, \mu))$.

**Proof.** First, let $j \in \mathcal{LW}_{\mathrm{o}}$. The constraint

$$\sum_{t \in \mathcal{T}} z_j^t \leqslant 1$$

of $LR1_j(\lambda, \mu)$ ensures that either there exists a $t_0 \in \mathcal{T}$ with $z_j^{t_0} = 1$ and $z_j^t = 0\ \forall t \neq t_0$ or $z_j^t = 0\ \forall t \in \mathcal{T}$. Moreover, it is easy to see that the optimal objective value of $LR1_j(\lambda, \mu)$ is equal to $v(LR1_{jt_0}(\lambda, \mu))$ if the binary variables are fixed as in the first case and equal to 0 if they are fixed as in the second case. Therefore, zero and $v(LR1_{jt_0}(\lambda, \mu))$ for $1 \leqslant t_0 \leqslant T$ are the only optimal values possible for $LR1_j(\lambda, \mu)$. Hence, their minimum has to be the optimal objective value of $LR1_j(\lambda, \mu)$.

Now let $j \in \mathcal{LW}_{\mathrm{c}}$. The constraint

$$\sum_{t \in \mathcal{T}} z_j^t = 1$$

of $LR1_j(\lambda, \mu)$ now ensures that there always exists a $t_0 \in \mathcal{T}$ with $z_j^{t_0} = 1$ and $z_j^t = 0\ \forall t \neq t_0$. In addition, it is again easy to see that $v(LR1_{jt_0}(\lambda, \mu))$ is the optimal objective value of $v(LR1_j(\lambda, \mu))$ if the binary variables are fixed in this way. Therefore, $v(LR1_{jt_0}(\lambda, \mu))$ for $1 \leqslant t_0 \leqslant T$ are the only optimal values possible for $LR1_j(\lambda, \mu)$ and so their minimum has to be the optimum. □

Due to Proposition 2, we only need to solve $T$ linear programs to get the solution of $LR1_j(\lambda, \mu)$ for $j \in \mathcal{LW}$. Additionally, we obtain for $j \in \mathcal{LW}_{\mathrm{o}}$ the information if and when warehouse $j$ is opened and for $j \in \mathcal{LW}_{\mathrm{c}}$ if and when existing warehouse $j$ is closed. Note that the $T$ linear programs $LR1_{jt}(\lambda, \mu), t = 1, \ldots, T$ can be solved by any LP solver, but due to their special structure it is more efficient to use the enumerative approach described in the Appendix.

As already mentioned at the beginning of this section, the solution obtained in the above process might not be feasible for $LR1(\lambda, \mu)$ because we left the constraints (6) aside. Hence, it might be necessary that some warehouses $j \in \mathcal{LW}$ have to be opened (if $j \in \mathcal{LW}_{\mathrm{o}}$) or closed (if $j \in \mathcal{LW}_{\mathrm{c}}$) in different time periods than those obtained in the above process, so that they are already in operation in the first or still in operation in the last time period. In [15], this solution process was not actually performed up to optimality. Thus, that paper only obtained suboptimal solutions for the relaxed problems. Here, we fix this approach. To do this in an optimal way, the optimal objective values of $LR1_{jt}(\lambda, \mu), \forall j, t$, are used to introduce an integer program which minimizes the additional cost for having constraints (6) fulfilled.

Let $\mathcal{LW}_{\mathrm{o}}^1 \subset \mathcal{LW}_{\mathrm{o}}$ ($\mathcal{LW}_{\mathrm{o}}^2 = \mathcal{LW}_{\mathrm{o}} \backslash \mathcal{LW}_{\mathrm{o}}^1$) be the set of indices of those warehouses which have been opened (not been opened) in the above process. Then, we define the following decision variables:

For $j \in \mathcal{LW}_{\mathrm{o}}^1$:

$$u_j := \begin{cases} 1 & \text{if warehouse } j \text{ is opened in the first time period,} \\ 0 & \text{if warehouse } j \text{ is open in its least cost time period.} \end{cases}$$

For $j \in \mathcal{LW}_{\mathrm{c}}$:

$$u_j := \begin{cases} 1 & \text{if warehouse } j \text{ is open during the whole planning horizon,} \\ 0 & \text{if warehouse } j \text{ is closed at the end of its least cost time period.} \end{cases}$$

For $j \in \mathcal{LW}_{\mathrm{o}}^2$:

$$u_j^a := \begin{cases} 1 & \text{if warehouse } j \text{ is opened in its least cost time period,} \\ 0 & \text{otherwise,} \end{cases}$$

$$u_j^b := \begin{cases} 1 & \text{if warehouse } j \text{ is opened in the first time period,} \\ 0 & \text{otherwise.} \end{cases}$$

These decisions represent the possibilities to increase the number of open warehouses in the first and in the last time period. Note that a warehouse $j \in \mathcal{LW}_{\mathrm{o}}^2$ can be opened in its least cost time period ($u_j^a = 1$) to have it open at the end of the planning horizon, or in the first time period ($u_j^a = 1$ and $u_j^b = 1$) to have it open throughout the whole planning horizon.

Furthermore, we associate additional penalty costs with those decisions.

For $j \in \mathscr{LW}^1_{\mathrm{o}}$:

$$\Delta_j := v(LR1_{j1}(\lambda,\mu)) - v(LR1_j(\lambda,\mu)).$$

For $j \in \mathscr{LW}_{\mathrm{c}}$:

$$\Delta_j := v(LR1_{jT}(\lambda,\mu)) - v(LR1_j(\lambda,\mu)).$$

For $j \in \mathscr{LW}^2_{\mathrm{o}}$:

$$\Delta^a_j := \min_{1 \leqslant t \leqslant T}\{v(LR1_{jt}(\lambda,\mu))\} \quad \text{and} \quad \Delta^b_j := v(LR1_{j1}(\lambda,\mu)).$$

Now, we formulate the following IP:

$$\begin{aligned} IP1(\lambda,\mu) \quad \text{Min} \; & \sum_{j\in\mathscr{LW}^1_{\mathrm{o}}} \Delta_j u_j + \sum_{j\in\mathscr{LW}^2_{\mathrm{o}}} (\Delta^a_j u^a_j + (\Delta^b_j - \Delta^a_j)u^b_j) + \sum_{j\in\mathscr{LW}_{\mathrm{c}}} \Delta_j u_j \\ \text{s.t.} \; & \sum_{j\in\mathscr{LW}^1_{\mathrm{o}}} u_j + \sum_{j\in\mathscr{LW}^2_{\mathrm{o}}} u^b_j \geqslant NW^1 - |\mathscr{LW}_{\mathrm{c}}|, \\ & \sum_{j\in\mathscr{LW}^2_{\mathrm{o}}} u^a_j + \sum_{j\in\mathscr{LW}_{\mathrm{c}}} u_j \geqslant NW^T - |\mathscr{LW}^1_{\mathrm{o}}|, \\ & u^a_j \geqslant u^b_j \quad \forall j \in \mathscr{LW}^2_{\mathrm{o}}, \\ & u_j \in \{0,1\} \quad \forall j \in \mathscr{LW}^1_{\mathrm{o}} \quad \forall j \in \mathscr{LW}_{\mathrm{c}}, \\ & u^a_j, u^b_j \in \{0,1\} \quad \forall j \in \mathscr{LW}^2_{\mathrm{o}}. \end{aligned}$$

The objective function of this IP minimizes all additional costs for fulfilling constraints (6). Note that the costs for opening a warehouse $j \in \mathscr{LW}^2_{\mathrm{o}}$ in the first time period are just $(\Delta^b_j - \Delta^a_j)$. The reason is that if a warehouse is open in the first time period, then it is also open in its best period.

The first constraint states that the number of operating warehouses $j \in \mathscr{LW}^1_{\mathrm{o}}$ and $j \in \mathscr{LW}^2_{\mathrm{o}}$ in the first time period has to be greater or equal to $NW^1$ minus the number of warehouses in $\mathscr{LW}_{\mathrm{c}}$ (they are open in the first time period anyway). The next constraint assures that the number of warehouses $j \in \mathscr{LW}^2_{\mathrm{o}}$ and $j \in \mathscr{LW}_{\mathrm{c}}$ which are in operation in the last time period is greater or equal to $NW^{\mathrm{T}}$ minus the number of warehouses $j \in \mathscr{LW}^1_{\mathrm{o}}$ (they are open the last time period anyway). Therefore, these two constraints ensure that the begin and end status are satisfied. The following group of constraints represents the fact that a warehouse $j \in \mathscr{LW}^2_{\mathrm{o}}$ which is already open in the first time period is also open in its best time period.

It is easy to observe that the solution of $IP1(\lambda,\mu)$ provides an optimal solution for $LR1(\lambda,\mu)$ by changing the binary variables obtained in the above process according to the optimal values of $u$. Moreover, we get the following result the proof of which is obvious.

**Proposition 3.**

$$v(LR1(\lambda,\mu)) = \sum_{j\in\mathscr{LW}} v(LR1_j(\lambda,\mu)) + v(IP1(\lambda,\mu)) + v(LR1_{\mathrm{o}}(\mu)).$$

Finally, to get the solution of $LR1(\lambda,\mu)$ we have to solve $IP1(\lambda,\mu)$. A closer look into the structure of this problem shows that its constraint matrix $A$ has the following form:

$$A = \begin{pmatrix} 1 & \cdots & 1 & 1 & \cdots & 1 & 0 & \cdots & 0 & 0 & \cdots & 0 \\ 0 & \cdots & 0 & 0 & \cdots & 0 & 1 & \cdots & 1 & 1 & \cdots & 1 \\ 0 & \cdots & 0 & -1 & \cdots & 0 & 0 & \cdots & 0 & 1 & \cdots & 0 \\ \vdots & \cdots & \vdots & 0 & \cdots & \vdots & \vdots & \cdots & \vdots & \vdots & \cdots & \vdots \\ \vdots & \cdots & \vdots & \vdots & \cdots & 0 & \vdots & \cdots & \vdots & \vdots & \cdots & 0 \\ 0 & \cdots & 0 & 0 & \cdots & -1 & 0 & \cdots & 0 & 0 & \cdots & 1 \end{pmatrix}.$$

As one can observe, this matrix is TU (see, e.g., [11, Theorem 8.9]). Hence, $IP1(\lambda, \mu)$ can be solved in polynomial time. Therefore, we get the solution of $LR1(\lambda, \mu)$ by solving $|\mathscr{LW}| \cdot T + 2$ linear programs.

### 3.2. Solution of $LR2(\lambda, \mu)$

To solve $LR2(\lambda, \mu)$, we use the same strategy as for $LR1(\lambda, \mu)$. Once the constraints (7) are removed, we separate $LR2(\lambda, \mu)$ into $|\mathscr{LP}|$ subproblems $LR2_k(\lambda, \mu)$ $(\forall k \in \mathscr{LP})$ which are defined as follows:

$$
\begin{aligned}
LR2_k(\lambda, \mu) \quad \text{Min} & \sum_{t\in\mathscr{T}} \sum_{j\in\mathscr{LW}} \sum_{p\in\mathscr{P}} (PTC^t_{jkp} WC^t_j - \lambda^t_{jp} WC^t_j) y^t_{jkp} + \sum_{t\in\mathscr{T}} TCP^t_k \zeta^t_k \\
\text{s.t.} & \sum_{j\in\mathscr{LW}} \sum_{p\in\mathscr{P}} WC^t_j y^t_{jkp} \leqslant PC^t_k \sum_{r\in T_{kt}} \zeta^t_k \quad \forall t \in \mathscr{T} \\
& \sum_{t\in\mathscr{T}} \zeta^t_k = 1 \ \text{ if } k \in \mathscr{LP}_c \quad \text{or} \quad \sum_{t\in\mathscr{T}} \zeta^t_k \leqslant 1 \ \text{ if } k \in \mathscr{LP}_o, \\
& 0 \leqslant y^t_{jkl} \leqslant 1 \quad \forall j \in \mathscr{LW}, \ \forall p \in \mathscr{P}, \ \forall t \in \mathscr{T}, \\
& \zeta^t_k \in \{0, 1\} \quad \forall t \in \mathscr{T}.
\end{aligned}
$$

Thus, we solve a problem for each plant. These problems are then decomposed into one problem for each time period, denoted by $LR2_{kt_0}(\lambda, \mu)(t_0 \in \mathscr{T})$. Observe that these problems are linear programs which can again be solved using any LP solver or by the enumerative approach described in the Appendix. Moreover, analogously to Propositions 2 and 3, the following result holds:

**Proposition 4.**

1. *If* $k \in \mathscr{LP}_o$, *then* $v(LR2_k(\lambda, \mu)) = \min\{\min_{1\leqslant t_0\leqslant T} v(LR2_{kt_0}(\lambda, \mu)), 0\}$.
2. *If* $k \in \mathscr{LP}_c$, *then* $v(LR2_k(\lambda, \mu)) = \min_{1\leqslant t_0\leqslant T} v(LR2_{kt_0}(\lambda, \mu))$.
3. $v(LR2(\lambda, \mu)) = \sum_{k\in\mathscr{LP}} v(LR2_k(\lambda, \mu)) + v(IP2(\lambda, \mu))$, *where* $IP2(\lambda, \mu)$ *is defined analogously to* $IP1(\lambda, \mu)$.

Once warehouses are substituted by plants, the proof is similar to the one of Propositions 2 and 3. Therefore, we omit it.

Note that due to this proposition, we again only need to solve $|\mathscr{LP}| \cdot T + 1$ linear programs to find a solution for $LR2(\lambda, \mu)$. Summarizing, the results of this section ensure that a solution for $LR(\lambda, \mu)$ can be found in polynomial time.

## 4. Heuristic solution algorithm

In the previous section, the solution of the Lagrangian Relaxation for $(D2ELI)$ has been described where $v(LR(\lambda, \mu))$ provides a lower bound on $v(D2ELI)$. This lower bound can be optimized using the well-known technique of Subgradient Optimization (see, e.g., [22]). Observe that for any given $\lambda \in \mathbb{R}$ and $\mu \geqslant 0$, a subgradient of $LR(\lambda, \mu)$ is given by

$$
\delta_{(\lambda,\mu)} = \begin{bmatrix} \delta_\lambda \\ \delta_\mu \end{bmatrix} = \begin{bmatrix} \sum_{i\in\mathscr{LC}} D^t_{ip} \tilde{x}^t_{ijp} + \tilde{I}^t_{jp} - \sum_{k\in\mathscr{LP}} WC^t_j \tilde{y}^t_{jkp} - \tilde{I}^{t-1}_{jp} \\ 1 - \sum_{j\in\mathscr{LW}} \tilde{x}^t_{ijp} - \tilde{o}^t_{ip} \end{bmatrix} \quad \forall i, j, p, t,
$$

with $(\tilde{x}, \tilde{y}, \tilde{o}, \tilde{I}, \tilde{z}, \tilde{\zeta})$ being an optimal solution for $LR(\lambda, \mu)$.

Furthermore, the optimal binary variables of $LR(\lambda, \mu)$ can be used to derive a feasible solution of $(D2ELI)$. This can be done by fixing the binary variables of $(D2ELI)$ to their optimal values in $LR(\lambda, \mu)$ and solving the remaining linear program. Therefore, we propose two heuristic procedures to improve this solution. The first one (PCAPACITY) tries to avoid the excessive use of outsourcing whenever outsourced supply is more expensive than supplying from inside the network. The second one is an Interchange heuristic which tries to further improve a given solution. These procedures are described next. After that, all these features (Lagrangian Relaxation, Subgradient Optimization and the two heuristic procedures) will be combined in an iterative algorithm.

*PCAPACITY* . Given a feasible solution for the binary variables of $(D2ELI)$, we perform the following steps:

*Step* 0 (Maximum amount of demand that could be covered using the own distribution network): Let $TP^t$ (respectively, $TW^t$) be the total possible plant (respectively, warehouse) capacity in time period $t$ ($t \in \mathscr{T}$). Moreover, let $IMAX^t$ ($0 \leqslant t < T$, with $IMAX^0 = 0$) be the maximal inventory that can be carried over from time period $t$ to $t+1$, and $D^t$ the total demand in time period $t$ ($t \in \mathscr{T}$). Then perform the following loop:

For $t=1, \dots, T$, check whether $TP^t + IMAX^{t-1} \geqslant D^t$ and $TW^t \geqslant D^t$. If not, set $D^t = \min\{TP^t + IMAX^{t-1}, TW^t\}$, $IMAX^t = 0$. Otherwise, set $IMAX^t = \min\{TW^t - D^t, TP^t + IMAX^{t-1} - D^t\}$.

*Step* 1 (warehouse capacity): Let $D^t$ be the total demand in time period $t$ (possibly adapted in Step 0) and $TWC^t$ the overall warehouse capacity in time period $t$ with respect to the given solution. In addition, let $GAP_w^t = D^t - TWC^t$. If $GAP_w^t \leqslant 0$ for all $t$, the warehouse capacity suffices in each time period and we proceed with Step 2. If not, we arrange all positive $GAP_w^t$'s in non-increasing order.

Let $t_0$ be the time period with the largest positive value for $GAP_w^t$. Then we define the following index for each warehouse $j \in \mathscr{LW}$ which is closed in time period $t_0$:

$$I(j,t_0) = \left[\min_{r \in T_{jt_0}} \{v(LR1_{jr}(\lambda,\mu))\} - \tilde{v}(LR1_j(\lambda,\mu))\right] \times \left[\max\left\{\frac{GAP_w^{t_0}}{WC_j^{t_0}}, 1\right\}\right]$$

with

$$\tilde{v}(LR1_j(\lambda,\mu)) = \begin{cases} v(LR1_{jt^*}(\lambda,\mu)) & \text{if } \exists\, t^* \text{ such that } z_j^{t^*} = 1,\\ 0 & \text{otherwise.}\end{cases}$$

This index can be interpreted as a measure of the additional costs for having warehouse $j$ open in time period $t_0$.

Now the warehouses which are closed in $t_0$ are opened one by one in non-decreasing order with respect to $I(\cdot, t_0)$, until the total customer demand can be fulfilled. "Opening" a warehouse $j$ in this context means that we change the values for $z_j^t$ for all $t$ in the following way:

$$z_j^r = \begin{cases} 1 & \text{if } r = \operatorname*{argmin}_{r \in T_{jt_0}}\{v(LR1_{jr}(\lambda,\mu))\},\\ 0 & \text{otherwise.}\end{cases}$$

After this process, it may happen that there is an excess of capacity in time period $t_0$ which is greater than or equal to the capacity of some of the opened warehouses. If this is the case, we close (reset the values for $z_j^t$) these warehouses again one by one in non-increasing order with respect to $I(\cdot, t_0)$, until all opened warehouses have (in time period $t_0$) a capacity which is greater than the excess.

After that, the values for $TWC^t$ and $GAP_w^t$ ($1 \leqslant t \leqslant T$) are adopted and the process is repeated until $GAP_w^t \leqslant 0\ \forall t$.

*Step* 2 (plant capacity): Let $TPC^t$ be the overall plant capacity in time period $t$ with respect to the given solution.

To determine the warehouse demand, we have to consider that inventory holding is allowed at warehouses from one time period to another. Therefore, we define the warehouse demand $DW^t$ for each time period $t$ ($1 \leqslant t \leqslant T$) as follows:

$$DW^1 = D^1, \quad DW^t = D^t - I^{t-1} \ \forall t > 1.$$

In this formula, $I^t$ is an upper bound on the total inventory holding that is possible at the end of time period $t$. To obtain this upper bound, we make use of the fact that the total inventory holding at the end of time period $t$ has to be:

1. less than or equal to the total plant capacity minus the total customer demand in time period $t$ plus the total inventory holding from time period $t-1$;
2. less than or equal to the difference between the total warehouse capacity and the total customer demand in time period $t$;
3. less than or equal to the total warehouse capacity in time period $t+1$.

The proof of these bounds follows directly from the feasibility region of $(D2ELI)$. Furthermore, $I^t$ cannot be negative even if one of the above bounds is less than zero. Therefore, we define $I^t$ as

$$I^t = \begin{cases} \tilde{I}^t & \text{for } \tilde{I}^t \geqslant 0,\\ 0 & \text{for } \tilde{I}^t < 0,\end{cases}$$

with

$$\tilde{I}^t = \min\{(TPC^t - D^t) + I^{t-1}, TWC^t - D^t, TWC^{t+1}\} \quad \forall t = 1, \ldots, T-1$$

and $I^0 = 0$.

To check whether there is enough plant capacity to fulfill the warehouse demand, we compute $I^{t-1}$, $DW^t$ and $TPC^t$ for $t = 1, \ldots, T$. The gap between warehouse demand and plant capacity is defined as $GAP_p^t = DW^t - TPC^t$ for all $t$. Then, we proceed as in Step 1; the only difference is that $v(LR2_k(\lambda, \mu))$ instead of $v(LR1_j(\lambda, \mu))$ is used to calculate the index $I(k, t_0)$.

*Step* 3 (transportation subproblem): We fix the binary variables according to Step 1 and 2 and solve the resulting linear program.

Next we give a formulation of the interchange procedure for improving solutions of $(D2ELI)$.

*PINTERCHANGE*. Given a solution for the binary variables of $(D2ELI)$, we execute the following steps:

*Step* 1: For all $j \in \mathscr{LW}$ do:

Change, one by one, the time period in which warehouse $j$ is opened (if $j \in \mathscr{LW}_o$) or closed (if $j \in \mathscr{LW}_c$) by changing the current value of the binary variables $(z_j^t)$. If $j \in \mathscr{LW}_o$ consider also the possibility to not open warehouse $j$ at all.

For each of these possibilities, check if the number of open warehouses in $t = 1$ (respectively, in $t = T$) is greater than or equal to $NW^1$ (respectively, $NW^{\mathrm{T}}$). If not, do not consider this possibility.

Otherwise, solve (*D2ELI*) with the binary variables fixed and perform *PCAPACITY* to see if the new solution is of lower cost than the current one. If this is the case, update the current solution.

*Step* 2: Proceed in the same way as in Step 1 for all $k \in \mathscr{LP}$.

*Step* 3: If there has been an improvement in Step 1 or 2, or both, go to Step 1. Otherwise STOP.

Finally we provide a complete formulation of the procedure for solving $(D2ELI)$.

**Algorithm**

**Initialization**

$Z_{\mathrm{LB}} = -\infty$ (current lower bound)

$Z_{\mathrm{UB}} = \infty$ (current upper bound)

$N1 = 0$ (number of Subgradient iterations)

$N2 = 0$ (number of Subgradient iterations without improvement)

$N3 = 0$ (number of PINTERCHANGE executions without improvement)

$\beta = \frac{1}{32}$

Use the optimal values of the dual variables of the linear relaxation of $(D2ELI)$ corresponding to the constraints 1 and 4 as initial solutions for the Lagrangian Multipliers $\lambda$ and $\mu$.

*Step* 0: Solve $LR(\lambda, \mu)$ with optimal solution $(\tilde{x}, \tilde{y}, \tilde{o}, \tilde{I}, \tilde{z}, \tilde{\xi})$. Set $Z_{\mathrm{LB}} = v(LR(\lambda, \mu))$. Solve $(D2ELI)$ with the binary variables fixed to $(\tilde{z}, \tilde{\xi})$. Execute PCAPACITY. Update $Z_{\mathrm{UB}}$ accordingly.

*Step* 1: Solve $LR(\lambda, \mu)$ with optimal solution $(\tilde{x}, \tilde{y}, \tilde{o}, \tilde{I}, \tilde{z}, \tilde{\xi})$. If $Z_{\mathrm{LB}} < v(LR(\lambda, \mu))$ set $N2{=}0$ and $Z_{\mathrm{LB}} = v(LR(\lambda, \mu))$, else set $N2 = N2 + 1$.

*Step* 2: If $N2 = |\mathscr{LW}| + |\mathscr{LP}|$ set $\beta = \frac{1}{2} \cdot \beta$, $N2 = 0$. Solve $(D2ELI)$ with the binary variables fixed to $(z,\ \xi)$ and execute PINTERCHANGE. If $Z_{\mathrm{UB}}$ could not be improved in PINTERCHANGE by more than 5% set $N3 = N3 + 1$, else set $N3 = 0$. If $N3 = 3$ then STOP.

*Step* 3: Determine the subgradient $\delta_{(\lambda,\mu)}$.

If $\|\delta_{(\lambda,\mu)}\|_2 = 0$ or $\beta \leqslant 0.001$ then STOP. Else compute

$$\pi = \frac{\beta \cdot (Z_{\mathrm{UB}} - Z_{\mathrm{LB}})}{\|\delta_{(\lambda,\mu)}\|_2},$$

and set $\lambda_{jp}^t = \lambda_{jp}^t + \pi \cdot (\delta_\lambda)_{jp}^t$ $(\forall j, p, t)$, $\mu_{ip}^t = \max\{0, \mu_{ip}^t + \pi \cdot (\delta_\mu)_{ip}^t\}$ $(\forall i, p, t)$.

*Step* 4: Set $N1 = N1 + 1$. If $N1 = 2000$ then STOP, else go to Step 1.

Table 1
Tested problem types

| | Customers | Commodities | Warehouses | | | Plants | | |
|---|---|---|---|---|---|---|---|---|
| | $\|\mathscr{LC}\|$ | $\|\mathscr{P}\|$ | $\|\mathscr{LW}_o\|$ | $\|\mathscr{LW}_c\|$ | $\|\mathscr{LW}\|$ | $\|\mathscr{LP}_o\|$ | $\|\mathscr{LP}_c\|$ | $\|\mathscr{LP}\|$ |
| P1 | 10 | 2 | 4 | 1 | 5 | 4 | 1 | 5 |
| P2 | 10 | 3 | 5 | 2 | 7 | 5 | 2 | 7 |
| P3 | 20 | 2 | 7 | 3 | 10 | 7 | 3 | 10 |
| P4 | 20 | 3 | 8 | 4 | 12 | 8 | 4 | 12 |
| P5 | 30 | 2 | 10 | 5 | 15 | 10 | 5 | 15 |
| P6 | 50 | 2 | 14 | 6 | 20 | 14 | 6 | 20 |
| P7 | 75 | 2 | 25 | 15 | 40 | 25 | 15 | 40 |
| P8 | 70 | 5 | 5 | 2 | 7 | 3 | 2 | 5 |
| P9 | 100 | 10 | 10 | 3 | 13 | 7 | 2 | 9 |
| P10 | 125 | 12 | 12 | 4 | 16 | 8 | 2 | 10 |

## 5. Computational study

The computational tests presented in this part have been designed to evaluate the performance of the heuristic procedure developed in the previous sections. On this account, the algorithm was implemented using *Visual C++* 6.0 where *ILOG Concert Technology* routines have been used for the implementation of the linear programs. Furthermore, *ILOG CPLEX* 8.1 has been used to solve these linear programs and to obtain exact solutions of the tested problems (by using *Branch* & *Bound* with default parameters of the solver). All computational studies have been performed on a PC with a *Pentium IV* processor with 2.4 GHz and 512 MB of RAM.

Ten problem types (P1–P10), which are described in Table 1, have been considered. For each of these types, the number of customers $|\mathscr{LC}|$, the number of commodities $|\mathscr{P}|$, and the number of warehouse and plant locations $|\mathscr{LW}|$ and $|\mathscr{LP}|$ are given.

Note that the structure of the first group of problem types (P1–P7) is similar to those in [15]. Therefore, the new results can be compared with those obtained in that paper for a much simpler model. Moreover, observe that for this group of problems, the relation between customer–commodity pairs and facility locations is roughly 2:1. However, this does not hold for the second group (P8–P10) where the number of customer–commodity pairs is considerably larger than the number of facility locations.

Furthermore, the number of time periods $T$ has been varied from 2 to 5 and from 7 to 8 so that altogether 60 problem type—time period pairs have been tested. For each of these pairs several problem instances have been generated. Thereby, the locations of all facilities have been uniformly distributed in the square $[1, 20] \times [1, 20]$ whereas the transportation costs have been determined proportionally to the Euclidean distance between the facilities. Additionally, these costs experience an increment between 5% and 10% in each time period, which reflects for example an inflation rate.

In contrast to that, all other cost factors have been generated using uniform distributions on different spaces. For the operating, opening, and closing costs of facilities, these spaces have been changed slightly for each time period, to represent the facts that new technologies become cheaper in the future and that costs for closing facilities are growing from time period to time period which is for example due to the obsolescence of production units. Furthermore, just like for the transportation costs, an increment between 5% and 10% has been added each time period. The customer demands have also been generated using uniform distributions whereas these distributions have been designed so that an increment could take place in each time period. At last, the capacities of the facilities have been generated (using again uniform distribution) so that a feasible solution (without outsourcing) for the resulting problems does always exist.

Using these definitions for data generation, several instances have been created for all problem type—time period pairs. For each of these instances a heuristic and an optimal solution have been obtained. The results of these computations can be found in Table 3. Hereby the values of Table 2 have been determined.

Observe that a time limit of two hours (i.e., 7200 s) has been used for solving the test-problems to optimality. If no optimal solution has been found within this limit, the best upper bound obtained so far has been used for the calculation of “E-Gap” and “Max. E-Gap”. Therefore, each time this happened for at least one example of a problem-type time period pair, the corresponding value in Table 3 was marked by ‘[a]’.

Table 2
Reported values

| Value | Description |
|---|---|
| H-Gap | Average percentage gap between the objective value of the heuristic solution and the lower bound resulting from the *Subgradient Optimization* |
| Max. H-Gap | Maximal (i.e. worst) value used to calculate H-Gap |
| E-Gap | Average percentage gap between the objective values of the heuristic and the optimal solution found by *CPLEX* |
| Max. E-Gap | Maximal (i.e. worst) value used to calculate E-Gap |
| CPU-E | Average CPU time needed to find the optimal solution |
| CPU-H | Average CPU time needed to find the heuristic solution |
| CPU-Dual | Average CPU time needed to find the initial Lagrangian Multipliers (initial lower bound) |
| *N* | Average number of iterations performed to find the heuristic solution |

Summarizing the results of Table 3 we start with the problem types P1–P7. We first observe that on the one hand the "H-Gap" for problem types P1–P3 ranges from 3.05% to 9.42% where the absolute maximum is 13.49%. Whereas, on the other hand, for problem types P4–P7, this value ranges from 1.43% to 4.11% with an absolute maximum of 6.81%. Moreover, one can observe that this gap decreases with increasing problem size, which is a quite promising fact. A very similar development takes place for an increasing number of time periods where larger time periods also lead to smaller gaps. Only for P7 this seems to be vice versa.

Considering the "E-Gap" for P1–P7 we obtain that these values range from 0.08% to 2.22% with an absolute maximum of 5.52%. However, an improvement for increasing problem sizes and number of time periods cannot be detected. Nonetheless, these results are very promising, too, since in nearly all cases this gap is only around 1% and very often even the optimum could be reached.

Comparing the running times of the optimal and the heuristic solution procedure we see that for the smallest problems *CPLEX* is faster than the proposed algorithm. But if the problem instances become larger, the heuristic proves to be much faster than the exact *Branch* & *Bound* algorithm.

For the second group of test problems (P8–P10), the values for the "H-Gap" range from 1.25% to 11.76% with an absolute maximum of 18.24%. These values are a slightly worse than those for the first group, but considering the values for the "E-Gap" we observe that the opposite is true. Because, if we exclude P9 and P10 with $T = 2$ we see that these values are always less than 1%. In fact, the heuristic solution was very often optimal. Only for the cases just mentioned, this value is larger; but this is because of two outliers for which the "E-Gap" values are 11.24% and 15.18%. Furthermore, the comparison of the running times leads to more or less the same results as for P1–P7.

Combining these results one can observe that the proposed method seems to be quite appropriate to solve $(D2ELI)$ heuristically. In this regard, especially the sharp upper bounds and the possibility to estimate the quality of a solution (using the lower bound provided by the Lagrangian Relaxation) if the exact values are not known, represent a noticeable advantage. Finally, a further direction of improvement is indicated by the relatively high computation time needed to obtain the initial Lagrangian Multipliers (CPU-Dual).

## 6. Conclusions

The design and configuration of a supply chain which involves, among other aspects, facility location decisions, is crucial for an efficient and cost-effective operation and management. We propose a formulation for a dynamic two-echelon multi-commodity capacitated plant location problem with inventory and outsourcing aspects which covers many issues of practical network configuration problems.

Therefore, the proposed model can be used to obtain better insight into the quantitative aspects of strategic planning within the supply chain context. Although one could argue that running times of algorithms are not too crucial for strategic planning, there are many more aspects than only quantitative ones which have to be considered. A typical planning cycle involves an ongoing iteration between applying the quantitative solution approach, validating the outcome, taking into account other managerial aspects, and then (eventually with corrected data) applying the quantitative approach again. In this application scenario, it is also clear that we do not need a model covering everything which might influence the final decision. Rather, we need a fast and robust method to check the quantitative implications of

Table 3
Average computational results

| | | P1 | P2 | P3 | P4 | P5 | P6 | P7 | P8 | P9 | P10 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| $T=2$ | H-Gap | 9.42 | 5.31 | 5.09 | 4.11 | 3.07 | 2.96 | 1.43 | 9.66 | 11.76 | 9.71 |
| | Max. H-Gap | 12.23 | 10.66 | 7.72 | 6.81 | 5.21 | 4.17 | 2.37 | 12.84 | 15.72 | 18.24 |
| | E-Gap | 1.44 | 0.56 | 1.32 | 1.05 | 0.65 | 1.30 | 0.66 | 0.33 | 4.23 | 3.61 |
| | Max. E-Gap | 5.52 | 2.76 | 3.71 | 3.09 | 1.46 | 2.78 | 1.51 | 1.66 | 11.24 | 15.18 |
| | CPU-E | 0.18 | 0.36 | 1.68 | 2.77 | 3.00 | 33.80 | 1154 | 4.56 | 77.00 | 162.6 |
| | CPU-H | 0.26 | 0.46 | 1.17 | 1.69 | 2.18 | 10.31 | 108.1 | 6.37 | 38.52 | 97.12 |
| | CPU-Dual | 0.075 | 0.055 | 0.083 | 0.23 | 0.29 | 1.28 | 16.91 | 0.87 | 19.18 | 46.16 |
| | $N$ | 28 | 32 | 74 | 50 | 64 | 169 | 547 | 163 | 44 | 52 |
| $T=3$ | H-Gap | 8.90 | 5.02 | 5.15 | 3.43 | 3.75 | 2.84 | 2.17 | 9.11 | 7.96 | 6.38 |
| | Max. H-Gap | 13.49 | 8.52 | 8.75 | 6.10 | 6.47 | 4.31 | 3.44 | 11.55 | 9.81 | 8.39 |
| | E-Gap | 2.96 | 0.97 | 2.22 | 1.23 | 1.27 | 1.00 | 1.34 | 0.32 | 0.092 | 0.53 |
| | Max. E-Gap | 5.22 | 2.38 | 4.91 | 4.02 | 4.06 | 2.46 | 2.44 | 0.77 | 0.46 | 1.21 |
| | CPU-E | 0.31 | 1.12 | 4.55 | 10.31 | 25.78 | 190.7 | 2493 | 12.54 | 284.8 | 566.7 |
| | CPU-H | 0.31 | 0.85 | 2.06 | 4.23 | 4.60 | 28.12 | 180.7 | 12.14 | 96.76 | 263.4 |
| | CPU-Dual | 0.033 | 0.087 | 0.17 | 0.76 | 0.86 | 3.86 | 170.1 | 2.64 | 48.21 | 119.1 |
| | $N$ | 23 | 42 | 69 | 85 | 67 | 262 | 306 | 142 | 50 | 120 |
| $T=4$ | H-Gap | 6.58 | 4.59 | 3.98 | 3.22 | 3.61 | 2.77 | 2.19 | 6.71 | 5.42 | 3.94 |
| | Max. H-Gap | 8.38 | 8.33 | 6.37 | 5.22 | 4.90 | 4.42 | 3.30 | 8.23 | 6.24 | 4.88 |
| | E-Gap | 0.71 | 0.73 | 1.11 | 1.04 | 1.60 | 1.09 | 1.07[a] | 0.31 | 0.12 | 0.099 |
| | Max. E-Gap | 3.21 | 4.24 | 3.47 | 2.47 | 2.81 | 2.26 | 2.13 | 0.90 | 0.58 | 0.49 |
| | CPU-E | 0.83 | 1.80 | 8.50 | 28.12 | 58.90 | 1523 | 7200[a] | 28.14 | 2331 | 3369 |
| | CPU-H | 0.39 | 1.93 | 3.13 | 7.51 | 13.49 | 48.72 | 587.1 | 16.65 | 161.2 | 926.2 |
| | CPU-Dual | 0.050 | 0.14 | 0.36 | 1.30 | 1.88 | 7.27 | 170.1 | 5.42 | 69.76 | 232.5 |
| | $N$ | 17 | 77 | 71 | 99 | 151 | 271 | 736 | 84 | 44 | 427 |
| $T=5$ | H-Gap | 5.88 | 4.90 | 3.88 | 2.87 | 2.99 | 2.84 | 2.35 | 4.99 | 3.96 | 3.29 |
| | Max. H-Gap | 11.98 | 7.62 | 6.22 | 5.00 | 4.59 | 4.11 | 3.16 | 7.89 | 5.19 | 4.46 |
| | E-Gap | 0.84 | 0.93 | 1.35 | 0.83 | 0.79 | 1.19[a] | 1.19[a] | 0.34 | 0.18[a] | 0.075[a] |
| | Max. E-Gap | 3.25 | 3.46 | 3.18 | 2.42 | 1.67 | 1.94 | 2.13 | 1.25 | 0.76 | 0.39 |
| | CPU-E | 0.93 | 5.24 | 29.68 | 189.4 | 395 | 5095[a] | 7200[a] | 31.77 | 3577[a] | 5033[a] |
| | CPU-H | 0.67 | 1.98 | 5.83 | 12.29 | 22.15 | 68.88 | 1048 | 21.52 | 307 | 1035 |
| | CPU-Dual | 0.067 | 0.22 | 0.61 | 2.32 | 3.34 | 16.80 | 385.5 | 9.47 | 115.2 | 352.2 |
| | $N$ | 28 | 50 | 96 | 105 | 164 | 316 | 657 | 33 | 44 | 52 |
| $T=7$ | H-Gap | 4.58 | 3.50 | 3.38 | 2.54 | 2.90 | 2.11 | 1.80 | 3.03 | 2.21 | 1.53 |
| | Max. H-Gap | 7.19 | 4.93 | 4.57 | 3.74 | 3.88 | 2.64 | 2.44 | 4.58 | 2.94 | 2.10 |
| | E-Gap | 0.08 | 0.60 | 0.84 | 0.80 | 1.14[a] | 0.79[a] | 0.49[a] | 0.009 | 0.23 | 0.01[a] |
| | Max. E-Gap | 0.64 | 2.39 | 1.96 | 1.54 | 1.95 | 1.23 | 1.00 | 0.043 | 0.51 | 0.07 |
| | CPU-E | 1.29 | 8.34 | 136.5 | 402.1 | 2322[a] | 5317[a] | 7200[a] | 92.68 | 3653 | 5948[a] |
| | CPU-H | 1.36 | 4.78 | 10.22 | 31.93 | 50.72 | 194.4 | 1297 | 55.37 | 795.9 | 3618 |
| | CPU-Dual | 0.13 | 0.44 | 1.46 | 5.28 | 7.68 | 43.67 | 989 | 11.78 | 334.8 | 2166 |
| | $N$ | 39 | 84 | 93 | 177 | 220 | 378 | 315 | 139 | 44 | 52 |
| $T=8$ | H-Gap | 4.47 | 3.05 | 3.07 | 2.19 | 2.81 | 2.03 | 1.94 | 2.08 | 1.59 | 1.25 |
| | Max. H-Gap | 6.42 | 4.85 | 4.30 | 3.61 | 3.97 | 3.96 | 3.33 | 3.11 | 1.94 | 2.10 |
| | E-Gap | 0.13 | 0.45 | 0.57 | 0.37 | 0.99[a] | 0.53[a] | 0.61[a] | 0.29 | 0.098[a] | 0.02[a] |
| | Max. E-Gap | 0.74 | 1.84 | 0.57 | 0.80 | 2.44 | 1.96 | 1.62 | 0.97 | 0.23 | 0.04 |
| | CPU-E | 1.20 | 9.82 | 278.1 | 1245 | 4304[a] | 6148[a] | 7200[a] | 51.13 | 3620[a] | 6768[a] |
| | CPU-H | 1.33 | 3.43 | 11.90 | 31.59 | 56.81 | 236 | 1871 | 43.86 | 1053 | 3938 |
| | CPU-Dual | 0.13 | 0.63 | 1.98 | 7.24 | 10.86 | 70.20 | 1455 | 17.88 | 504.9 | 2315 |
| | $N$ | 32 | 37 | 82 | 198 | 185 | 293 | 441 | 24 | 44 | 52 |

[a]Means that at least one example could not be solved to optimality within the time limit.

certain assumptions within a short time, allowing the decision makers to rule out quickly non-desirable solutions and identify good ones.

Because of the complexity of the model, we present a heuristic solution method which is based on a Lagrangian Relaxation of the problem. This relaxation yields solutions which are feasible for the original formulation, but, due to a lack in available capacity, possibly use outsourcing to satisfy customer demands. In a second step, starting with these solutions, we first construct solutions which do not rely on outsourcing but use internal capacity instead. Afterward, we try to improve them using a simple local search procedure. The computational results for this approach, as reported in Table 3, are very encouraging. The gaps between the optimal and the heuristic solutions indicate that our method is acceptable to solve the dynamic two-echelon multi-commodity capacitated plant location problem with inventory and outsourcing aspects. Different experiments, as for instance stopping *CPLEX* after some reasonable time, might lead to other interesting comparisons.

We note that the proposed method can easily be extended to cope with a multi-echelon (more than two echelons) network structure. In this case the flow constraints between each consecutive level of facilities are relaxed which results in a similar decomposition as for the two-echelon case. Then, we can again apply the construction heuristic in order to obtain feasible solutions. Another extension that can easily be incorporated into the model, is the possibility of backlogged demand. In this regard, it is sufficient to define a new set of variables, namely $b^t_{ip}$, as the fraction (with resp. to $D^t_{ip}$) of product $p$ ordered by customer $i$ that is backlogged in time period $t$ and adapt the objective function (adding a term that amounts to the new cost) and constraints (1) ($\sum_{j\in\mathscr{LW}} x^t_{ijp}+o^t_{ip}+b^t_{ip}-b^{t-1}_{ip}\geqslant 1\ \forall i\in\mathscr{LC}, \forall p\in\mathscr{P}, \forall t\in\mathscr{T}$), accordingly. Some further extensions are currently under research.

To obtain tighter upper bounds meta heuristics, like variable neighborhood search, can be employed. A different approach to tackle the relaxed problem in order to obtain lower bounds is the use of Benders decomposition (see [23]).

**Acknowledgments**

The authors thank Spanish Ministry of Education and Science (Grant numbers: HA2003:0121 and MTM2004:0909) and the German Academic Exchange Service (DAAD) (Grant number: D/03/40310) for partial support.

**Appendix**

In the following, we will show how to solve $LR1_{jt_0}(\lambda,\mu)$ ($\forall j\in\mathscr{LW},\ t\in\mathscr{T}$) by inspection. Therefore, consider the following two types of linear programs (for fixed $j\in\mathscr{LW}$ and $t\in\mathscr{T}$): ($LR1A_{jt}(\lambda,\mu)$)

$$
\begin{aligned}
\operatorname{Min} f^1_{Ajt(\lambda,\mu)}(x,I) := & \sum_{i\in\mathscr{LC}}\sum_{p\in\mathscr{P}}(TC^t_{ijp}D^t_{ip}+\lambda^t_{jp}D^t_{ip}-\mu^t_{ip})x^t_{ijp}+\sum_{p\in\mathscr{P}}(IC^t_{jp}+\lambda^t_{jp}-\lambda^{t+1}_{jp})I^t_{jp}\\
\text{s.t.} & \sum_{i\in\mathscr{LC}}\sum_{p\in\mathscr{P}} D^t_{ip}x^t_{ijp}+\sum_{p\in\mathscr{P}} I^t_{jp}\leqslant WC^t_j,\\
& \sum_{p\in\mathscr{P}} I^t_{jp}\leqslant WC^{t+1}_j,\\
& I^t_{jp}\geqslant 0 \quad \forall p\in\mathscr{P},\\
& 0\leqslant x^t_{ijp}\leqslant 1 \quad \forall i\in\mathscr{LC}, \forall p\in\mathscr{P},
\end{aligned}
$$

and
($LR1B_{jt}(\lambda,\mu)$)

$$
\begin{aligned}
\operatorname{Min} f^1_{Bjt(\lambda,\mu)}(x,I) := & \sum_{i\in\mathscr{LC}}\sum_{p\in\mathscr{P}}(TC^t_{ijp}D^t_{ip}+\lambda^t_{jp}D^t_{ip}-\mu^t_{ip})x^t_{ijp}\\
\text{s.t.} & \sum_{i\in\mathscr{LC}}\sum_{p\in\mathscr{P}} D^t_{ip}x^t_{ijp}\leqslant WC^t_j,\\
& 0\leqslant x^t_{ijp}\leqslant 1 \quad \forall i\in\mathscr{LC}, \forall p\in\mathscr{P}.
\end{aligned}
$$

Now, we distinguish the following two cases:

1. Let $j \in \mathscr{LW}_{\rm o}$ and $t_0 \in \mathscr{T}$. Then $LR1_{jt_0}(\lambda,\mu)$ can obviously be separated into $T-t_0$ problems of type $(LR1A_{jt}(\lambda,\mu))$ $(t=t_0,\ldots,T-1)$ and one problem of type $(LR1B_{jt}(\lambda,\mu))$ $(t=T)$. Thus

$$v(LR1_{jt_0}(\lambda,\mu))=\sum_{t=t_0}^{T-1} v(LR1A_{jt}(\lambda,\mu))+v(LR1B_{jT}(\lambda,\mu))+TCW_j^{t_0}.$$

2. Let $j \in \mathscr{LW}_{\rm c}$ and $t_0 \in \mathscr{T}$. Then $LR1_{jt_0}(\lambda,\mu)$ can obviously be separated into $t_0-1$ problems of type $(LR1A_{jt}(\lambda,\mu))$ $(t=1,\ldots,t_0-1)$ and one problem of type $(LR1B_{jt}(\lambda,\mu))$ $(t=t_0)$. Thus

$$v(LR1_{jt_0}(\lambda,\mu))=\sum_{t=1}^{t_0-1} v(LR1A_{jt}(\lambda,\mu))+v(LR1B_{jt_0}(\lambda,\mu))+TCW_j^{t_0}.$$

Therefore, we only need to solve $LP1A_{jt}(\lambda,\mu)$ and $LP1B_{jt}(\lambda,\mu)$ $(\forall j \in \mathscr{LW},\ t \in \mathscr{T})$ which can be done by inspection as described below.

First, denote $C_{ijp}^t=TC_{ijp}^t D_{ip}^t+\lambda_{jp}^t D_{ip}^t-\mu_{ip}^t$ and $P_{jp}^t=IC_{jp}^t+\lambda_{jp}^t-\lambda_{jp}^{t+1}$. Observe that for fixed $j \in \mathscr{LW}$ and $t \in \mathscr{T}$ the following holds:

$$C_{ijp}^t>0 \;\Rightarrow\; x_{ijp}^t=0 \quad \text{and} \quad P_{jp}^t>0 \;\Rightarrow\; I_{jp}^t=0.$$

Therefore, we assume in the following that there exists at least one $C_{ijp}^t \leqslant 0$ (for some $i \in \mathscr{LC}$ and $p \in \mathscr{P}$). Moreover, we assume that there exists at least one $P_{jp}^t \leqslant 0$ (for some $p \in \mathscr{P}$).

Solution of $LR1A_{jt}(\lambda,\mu)$ for fixed $j$ and $t$

Let $p'=\arg\min_{p\in\mathscr{P}} P_{jp}^t$. Then arrange in non-decreasing sequence $C_{ijp}^t/D_{ip}^t$ $(\forall i \in \mathscr{LC},\ p \in \mathscr{P})$ together with $P_{jp'}^t$ and let $r^p$ be the index of $P_{jp'}^t$ in this sequence. Moreover, let $C_r/D_r$ $(r \neq r^p)$ be the $r$th element of the sequence and $x_r$ its associated variable $x_{ijp}^t$. Finally, let $r^-$ be the index of the last non-positive element in the sequence. Note that $r^p \leqslant r^-$ since, by assumption, there exists at least one $P_{jp}^t \leqslant 0$ for some $p \in \mathscr{P}$. Now $LR1A_{jt}(\lambda,\mu)$ can be solved by the following procedure:

Procedure 1
Initialization
Set $x_r=0\ \forall r \neq r^p$; $I_{jp}^t=0\ \forall p \in \mathscr{P}$; $WC_0=WC_j^t$; $r=1$.
Iteration
While $r \leqslant r^-$ do:
If $(r \neq r^p)$,
If $(D_r \leqslant WC_{r-1})$,
$x_r=1$; $WC_r=WC_{r-1}-D_r$; $r=r+1$.
Else
$x_r=\dfrac{WC_{r-1}}{D_r}$; STOP.
Else $(r=r^p)$,
If $(WC_{r-1}>WC_j^{t+1})$,
$I_{jp'}^t=WC_j^{t+1}$; $W_r=W_{r-1}-WC_j^{t+1}$; $r=r+1$.
Else,
$I_{jp'}^t=WC_{r-1}$; STOP.

Solution of $LR1B_{jt}(\lambda,\mu)$ for a fixed $j$ and $t$

Arrange in non-decreasing sequence $C_{ijp}^t/D_{ip}^t$ $(\forall i \in \mathscr{LC},\ p \in \mathscr{P})$. Then, let $C_r/D_r$ be the $r$th element in this sequence and $x_r$ its associated variable $x_{ijp}^t$. Finally, let $r^-$ be the index of the last non-positive element in the sequence.

Now $LR1B_{jt}(\lambda, \mu)$ can be solved by the following procedure:

Procedure 2
Initialization
Set $x_r = 0\ \forall r$; $WC_0 = WC_j^t$; $r = 1$.
Iteration
While $r \leqslant r^-$ do:
If $(D_r \leqslant WC_{r-1})$,
$x_r = 1; \quad WC_r = WC_{r-1} - D_r; r = r + 1.$
Else,
$x_r = \dfrac{WC_{r-1}}{D_r}; \quad$ STOP.

Note that these two procedures solve the proposed problems since they can both be reduced to continuous Knapsack problems. Furthermore observe that a similar approach can be made for the solution of $LR2_{kt_0}(\lambda, \mu)$ so that these problems can be solved by inspection, as well.

## References


[1] Kalcsics J, Melo T, Nickel S, Schmid-Lutz V. Facility location decisions in supply chain management. Operations Research Proceedings 1999; 467–472.
[2] Bender T, Hennes H, Kalcsics J, Melo T, Nickel S. Location software and interface with GIS and supply chain management. In: Drezner Z, Hamacher HW, editors. Facility location. Applications and theory. Berlin, Heidelberg: Springer; 2002. p. 233–74.
[3] Bramel J, Simchi-Levi D. The logic of logistics. Theory, algorithms, and applications for logistics management. Springer series in operations research. New York: Springer; 1997.
[4] Aikens CH. Facility location models for distribution planning. European Journal of Operational Research 1985;22:263–79.
[5] Crainic TG, Delorme L. Dual-ascent procedures for multicommodity location-allocation problems with balancing requirements. Transportation Science 1993;27(2):90–101.
[6] Crainic TG, Delorme L, Dejax P. A branch and bound method for multicommodity location with balancing requirements. European Journal of Operational Research 1993;65:368–82.
[7] Chardaire P, Sutter A, Costa MC. Solving the dynamic facility location problem. Networks 1996;28:117–24.
[8] Marín A. Análisis y resolución de problemas de localización discreta en dos etapas mediante técnicas basadas en la descomposición lagrangiana, PhD thesis, Departamento de Matemática Aplicada y Estadística, Universidad de Murcia, 1996.
[9] Pirkul H, Jayaraman V. Production transportation and distribution planning in a multi-commodity tri-echelon system. Transportation Science 1996;30(4):291–302.
[10] Drezner Z, editor, Facility location. Berlin: Springer; 1995.
[11] Hu TC. Integer programming and network flows. Reading, MA: Addison-Wesley; 1970.
[12] Krarup J, Pruzan PM. The simple plant location problem: survey and synthesis. European Journal of Operational Research 1983;12:36–81.
[13] Holmberg K, Jornsten K. Dual search procedures for the exact formulation of the simple plant location problem with spatial interaction. Location Science 1996;4:83–100.
[14] Melo MT, Nickel S, Saldanha da Gama F. Dynamic multi-commodity capacitated facility location: a mathematical modeling framework for strategic supply chain planning. Computers and Operations Research 2006;33:181–208.
[15] Hinojosa Y, Puerto J, Fernández FR. A multi-period two-echelon multi-commodity capacitated plant location problem. European Journal of Operational Research 2000;123:271–91.
[16] Barceló J, Fernández E, Jornsten K. Computational results from a new lagrangean relaxation algorithm for the capacitated plant location problem. European Journal of Operational Research 1991;53:38–45.
[17] Beasley JE. Lagrangian heuristics for location problems. EJOR 1993;65:383–99.
[18] Daskin M. Network and discrete location. Models, algorithms and applications, Wiley-Interscience series in discrete mathematics and optimization. New York, NY: Wiley, 1995.
[19] Erlenkotter D. A dual-based procedure for uncapacitated facility location. Operational Research 1978;26(6):992–1009.
[20] Fisher ML. The Lagrangian relaxation method for solving integer programming problems. Management Science 1981;27(1):1–18.
[21] Guignard M, Opaswongkarn K. Lagrangean dual ascent algorithms for computing bounds in capacitated plant location problems. EJOR 1990;46:73–83.
[22] Held M, Wolfe P, Crowder H. Validation of subgradient optimization. Mathematical programming 1974;6:62–88.
[23] Benders JF. Partitioning procedures for solving mixed-variables programming problems. Numerische Mathematik 1962;4:238–52.