Top (2014) 22:80–108
DOI 10.1007/s11750-011-0235-3

ORIGINAL PAPER

An efficient heuristic approach for a multi-period logistics network redesign problem

M.T. Melo · S. Nickel · F. Saldanha-da-Gama

Received: 19 July 2010 / Accepted: 10 September 2011 / Published online: 6 October 2011

Abstract In this paper, a multi-period logistics network redesign problem arising in the context of strategic supply chain planning is studied. Several aspects of practical relevance are captured, namely, multiple echelons with different types of facilities, product flows between facilities in the same echelon, direct shipments to customers, and facility relocation. A two-phase heuristic approach is proposed to obtain high-quality feasible solutions to the problem, which is initially modeled as a large-scale mixed-integer linear program. In the first phase of the heuristic, a linear programming rounding strategy is applied to find initial values for the binary location variables. The second phase of the heuristic uses local search to correct the initial variable choices when a feasible solution is not identified, or to improve the initial feasible solution when its quality does not meet given criteria. The results of a computational study are reported for randomly generated instances comprising a variety of logistics networks.

Keywords Logistics network redesign · Heuristic · Linear programming · Rounding · Local search

Mathematics Subject Classification (2000) 90B06 · 90B80 · 90C06 · 90C11 · 90C59

M.T. Melo
Business School, Saarland University of Applied Sciences, 66123 Saarbrücken, Germany
e-mail: teresa.melo@htw-saarland.de

S. Nickel
Fraunhofer Institute for Industrial Mathematics and Institute for Operations Research, Karlsruhe Institute of Technology, 76128 Karlsruhe, Germany
e-mail: stefan.nickel@kit.edu

F. Saldanha-da-Gama (✉)
Department of Statistics and Operations Research, and Operations Research Center, University of Lisbon, 1749-016 Lisbon, Portugal
e-mail: fsgama@fc.ul.pt

1 Introduction

Over the past decades, real-world production and distribution networks have become increasingly complex logistics systems comprising multiple facilities linked by transportation channels. Strategic logistics network design is concerned with long-term decisions regarding the configuration of the supply chain network. Typically, it involves selecting sites for the location of new facilities, deciding on their number and size, and choosing distribution channels as well as transportation modes to meet customer demands. Clearly, these decisions have a major impact on the long-term profitability and competitive advantage of a company. Determinant elements include customer service levels, flexibility to deal with potential pitfalls (e.g., equipment breakdown) and shipment reliability. According to Harrison (2004), up to 80% of the total cost of a product is driven by decisions made during the design phase of the logistics network.

Network design decisions are mostly triggered by changing market conditions rather than by the need to create a new supply chain from scratch (see Simchi-Levi et al. 2005). Therefore, in practice a company considers changing the structure of its distribution network from time to time. Due to the globalization of the economy and advances in information technology, redesign processes have become more frequent and their efficiency more important. This has been experienced, for example, by many European companies as a result of the economic transition that started in Eastern Europe during the last decade and the successive enlargement of the European Union. The impact of these changes has been noticed, for example, on markets, freight rates, transport infrastructures, and road networks. Expansion opportunities to new markets have appeared, thereby giving rise to the need to redesign existing supply chains. Usually, expansion plans take the form of opening new facilities in new geographical areas either because of the lack of room for capacity increase at the present locations or to be closer to new markets. In other cases, fierce competition has forced companies to relocate their facilities to areas with more favorable economic conditions (e.g., lower labor costs). Finally, mergers, acquisitions, and strategic alliances often motivate network design studies for supply chain consolidation. Hammami et al. (2008) provide a detailed discussion of the factors leading to supply chain reconfiguration, in particular in a delocalization context.

The contribution of this paper is to propose an efficient heuristic approach to solve a comprehensive network redesign problem. Given a supply chain network with the general structure depicted in Fig. 1, a multi-echelon, capacitated facility relocation problem is considered. It is assumed that a number of customer zones have known demands for multiple commodities over a multi-period horizon. In addition, several potential sites are available for establishing new facilities. The operation of the latter is triggered by moving capacity from existing facilities to new sites over the planning horizon. This enables modeling real-world situations in which the operating activity of a new facility gradually increases until it reaches a desired level. At the same time, the activity level of an existing facility progressively decreases until the facility is eventually removed from service. Capacity transfers lead thus to facility relocation and are financed by a limited budget, which also pays for establishing new facilities and closing existing facilities.

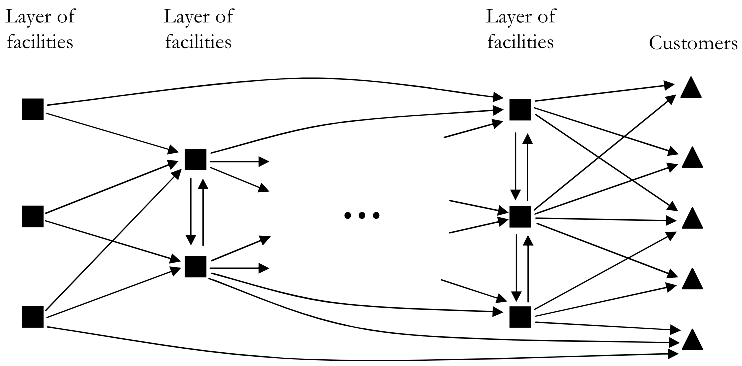


Fig. 1 A generic supply chain network

The main strategic decisions to be taken are outlined as follows: (i) Which existing facilities should have their capacities partially or totally transferred and in which periods should relocation take place? (ii) Which potential facility sites should be selected to receive the transferred capacities and when should they start operating? (iii) How should commodities flow through the network and, in particular, from which facilities should customer demands be satisfied in each period? (iv) Which facilities should hold stock? How large should stock levels be in each period? (v) How should the available budget be invested? How much money should be retained in each period to gain interest and be used in future investments?

The objective is to redesign the distribution network so as to minimize the sum of variable and fixed costs associated with the above location and supply chain decisions.

Melo et al. (2006) presented a comprehensive modeling framework for this problem and showed that medium-sized instances can be solved to optimality with commercial, off-the-shelf (COTS) optimization software within a reasonable time limit. Nevertheless, it is clear that this approach fails when supply chain redesign problems of realistic size need to be solved. Furthermore, most companies need an optimization-based decision support tool able to tackle the complexity and the dynamic nature of their supply chains. At the same time, such tools should allow rapid prototyping and the evaluation of alternative network configurations. In other words, companies need analytical tools with re-optimization capabilities for performing "what-if" analyses in a reasonable amount of computing time. This calls for the development of heuristic methods with a good trade-off between solution quality and computational effort.

The main contribution of this paper is to propose a new heuristic approach that explores the structure of one of the mixed-integer linear programming (MILP) formulations in Melo et al. (2006) to obtain high-quality feasible solutions. The new approach consists of two distinct phases. In the first phase, a linear programming rounding procedure is applied to obtain an initial solution to the problem, which may be infeasible. The second phase is a repair and improvement procedure for correcting infeasibilities (in case they exist) and to improve the initial solution.

The general principle behind a rounding approach is rather simple. However, often, the application of a basic principle to a concrete problem is far from being straightforward, as it is the case with the problem addressed in this paper. Our contribution aims at showing how a rounding procedure can be successfully applied to a problem comprising the complex features we consider in this paper. Additionally, a rounding procedure is not always likely to be successful. Since it is important to understand when such an approach can be effectively applied, we pay special attention to this issue. A third significant aspect that we consider is the usefulness of allowing the search path to include infeasible solutions and the ability to repair them. One important feature of our methodology is precisely the repair mechanism whose efficiency in removing infeasibilities is supported by the computational experiments performed. Finally, the heuristic procedure we propose is modular in the sense that the construction phase or the repair/improvement phase can be replaced by some other appropriate procedure.

The remainder of the paper is organized as follows. Section 2 provides a critical review of multi-period network design models in a supply chain management (SCM) context. For the sake of completeness, Sect. 3 presents one of the MILP formulations introduced in Melo et al. (2006) that renders the general setting for the heuristic development. Section 4 is dedicated to the new solution methodology which combines a linear programming rounding based strategy with local search. The results of our computational study are reported in Sect. 5. Finally, in Sect. 6 conclusions and directions for further research are given.

2 Literature review

In the last few years, the interaction between facility location and supply chain design has received increasing attention as shown by the extensive survey by Melo et al. (2009a). Driven by the need to model real-world problems, researchers have attempted to go beyond the classic facility location setting by considering key features to SCM such as supplier selection, production planning, inventory management, distribution, routing and other logistics activities (see Daskin et al. 2005). Moreover, globalization trends have also strongly impacted the development of new facility location models as described by Goetschalckx et al. (2002) and Meixell and Gargeya (2005).

Facility location decisions are inherently strategic and long-term in nature due to the large capital outlays that are involved. Consequently, the timing of facility locations, expansions and relocations over an extended time horizon is of major importance to decision-makers. In contrast to the static case, significantly fewer papers have been published on dynamic (i.e., multi-period) facility location problems. Within this problem class, the focus has been mostly given to rather simple networks comprising a single echelon of facilities and a single product, thus disregarding the supply chain context (see the recent review by Melo et al. 2009a and references therein). However, many real-world supply chain networks exhibit a multi-layer structure with at least two facility echelons, in addition to the customer layer. Moreover, the flow of multiple commodities through the network is often conveyed by an elaborated distribution system, linking facilities belonging to the same layer as well as to different layers (see Fig. 1).

A few contributions have appeared in the last decade that capture relevant aspects of supply chain network design in a multi-period context. Hinojosa et al. (2000, 2008) address a two-echelon multi-commodity network redesign problem. An initial network configuration is considered that gradually changes over a multi-period horizon through opening new facilities and closing existing ones. A lower bound on the number of facilities operating in each location layer is imposed both in the first and the last period of the planning horizon. The initial model developed in Hinojosa et al. (2000) was later extended in Hinojosa et al. (2008) through the integration of strategic inventory decisions. Canel et al. (2001) partly capture this supply chain feature in a two-echelon, capacitated, multi-commodity facility location model by allowing customers to be directly delivered from manufacturing plants as well as from intermediate level facilities. Location decisions are confined to the latter facilities which may be opened and closed more than once during the planning horizon. This feature is more suited to new facilities that are rented instead of being built, since in that case lower fixed setup costs are incurred. Ambrosino and Scutellà (2005) broaden the scope of the previous models (Canel et al. 2001; Hinojosa et al. 2000, 2008) through a three-echelon model that integrates strategic and operational decisions. Location and inventory decisions concern intermediate echelons comprising central and regional depots, while the operational aspect involves the design of vehicle routes to service customer demands.

In the context of reverse logistics, Srivastava (2008) addresses the problem of locating new collection and rework centers for product recovery over a given time horizon. Ko and Evans (2007) also study the problem of expanding an existing multi-commodity network through the location of new warehouses and repair centers. The former facilities receive end products from manufacturing plants and distribute them to end users, while the latter facilities distribute products returned by customers to the plants. In contrast to Srivastava (2008), location and supply chain decisions are integrated in a single model. Moreover, a nonlinear cost minimization objective for network reconfiguration is considered. This is a feature that has received little attention in the literature since it adds further complexity. The setup of new facilities is phased during the time horizon by allowing their operating activity to gradually increase through capacity expansion, which is a feature that is seldom considered by classic location models (see also the critical review by Julka et al. 2007). Thanh et al. (2008a) incorporate this and other relevant supply chain features in a multi-period model. In addition to the usual facility location and transportation decisions, the authors include decisions regarding supplier selection, multi-stage production planning, inventory management, and capacity operating levels. Furthermore, in a three-echelon network, materials flow downstream not only between adjacent layers but also across facilities in different layers.

The models developed by Vila et al. (2006) and Wilhelm et al. (2005) have a broader scope than the one in Thanh et al. (2008a) due to the integration of multiple features relevant to international supply chains. In particular, they include transfer prices and various financial rates (currency exchange, import duties, and income taxes) in different countries. A global after-tax profit maximization objective function is considered in contrast to the frequently used cost minimization objective. Multi-stage production, inventory and distribution decisions may change over time while location decisions are implemented at the beginning of the planning horizon. In Vila et al. (2006), the capacity of a facility may be expanded and later removed to deal with demand fluctuations. The model is applied to a real-case from the Canadian softwood lumber industry. The application context of the mathematical model developed by Troncoso and Garrido (2005) is also the forest industry. The objective is to select the optimal location and size of a new saw-mill in Chile. Although a single commodity is modeled, it may flow between any pair of facilities in the network, including between facilities within the same layer. This feature is also present in Vila et al. (2006). In addition to location and transportation decisions, capacity decisions are addressed by allowing the initially installed production capacity to increase during the planning horizon.

The literature reviewed so far has several features in common, namely a network topology with at least two facility echelons and transportation channels that go beyond links between adjacent layers. Furthermore, location and supply chain decisions are often integrated in a multi-period, multi-commodity model. In particular, location planning is not confined to fixing the time periods for opening and closing facilities. Capacity expansion decisions are also modeled. The framework developed by Melo et al. (2006) takes all these features into account and even extends the scope of the existing models by explicitly considering facility relocation through gradual capacity transfers from existing locations to new sites over time. Following the seminal article by Ballou (1968), this aspect remained overlooked until recently. Melachrinoudis and Min (2000) consider a simple network structure with a single facility layer and a single commodity. A limited budget is available for facility relocation which is a feature rarely captured by network design models (see Melo et al. 2009a). Hammami et al. (2008) identify the key features that impact the redesign of a supply chain in a delocalization context and criticize the lack of mathematical models that incorporate all relevant decisions. The large scope and complexity of the problem along with difficulties in data collection account for this gap. Finally, Wolf and Merz (2007) developed a simple evolutionary algorithm to solve one of the problems addressed by Melo et al. (2006) heuristically.

3 Mathematical formulation

In this section, we first introduce the notation that will be used throughout the paper. As the new heuristic solution method that will be presented in Sect. 4 relies upon one of the MILP formulations developed by Melo et al. (2006), we will briefly describe it. Details regarding the motivating assumptions and the underlying supply chain context can be found in Melo et al. (2006).

The network topology shown in Fig. 1 is the starting point for our logistics network redesign model. It comprises different types of operating facilities (any number of facility layers may be considered as well as any system of transportation channels). In addition, a finite set of candidate sites for locating new facilities has been identified. Over the planning horizon, facility relocation takes place by gradually moving capacity from existing facilities to the selected new sites.

Table 1 Index sets

Symbol	Description
L	Set of all facilities
S^c	Set of *existing* facilities that can be closed
S^o	Set of potential sites for establishing *new* facilities
S	Set of *selectable* facilities; $S = S^c \cup S^o$, $S \subset L$
$L \setminus S$	Set of *non-selectable* facilities
P	Set of product families
T	Set of periods; $\|T\| = n$

Table 2 Costs

Symbol	Description
$PC^t_{i,p}$	Variable cost of producing or purchasing (from an external supplier) one unit of product $p \in P$ by facility $i \in L$ in period $t \in T$
$TC^t_{i,j,p}$	Variable cost of shipping one unit of product $p \in P$ from facility $i \in L$ to facility $j \in L$ $(i \neq j)$ in period $t \in T$
$IC^t_{i,p}$	Variable inventory carrying cost per unit on hand of product $p \in P$ in facility $i \in L$ at the end of period $t \in T$
$MC^t_{i,j}$	Variable cost of moving one unit of capacity at the beginning of period $t \in T \setminus \{1\}$ from the existing facility $i \in S^c$ to a new facility established at site $j \in S^o$
OC^t_i	Fixed cost of operating facility $i \in L$ in period $t \in T$
FC^t_i	Fixed setup cost charged in period $t \in T \setminus \{n\}$ when a new facility established at site $i \in S^o$ starts its operation at the beginning of period $t+1$
SC^t_i	Fixed cost charged in period $t \in T \setminus \{1\}$ for closing the existing facility $i \in S^c$ at the end of period $t-1$

Table 1 describes the index sets. *Non-selectable* facilities refer to facilities that are not subject to capacity relocation. Such facilities may include plants and warehouses that must operate throughout the planning horizon. Moreover, customer locations always belong to this class.

Table 2 summarizes all costs. Since the establishment of a new facility is often a time-consuming process, it is assumed that it takes place in the period immediately preceding the start-up of operations. On the other hand, when an existing facility ceases operating, the corresponding fixed closing costs are charged in the following period. Relocation costs due to capacity shifts depend on the amount moved from an existing facility to a new site, and account, for example, for workforce and equipment transfers. Capacities moved to new sites cannot be withdrawn in later periods.

Table 3 introduces additional input parameters. The capacity of each existing facility is assumed to be nonincreasing over the planning horizon. Similarly, potential new facilities have nondecreasing capacities throughout the time horizon.

Table 4 describes the decision variables. Existing facilities may have an initial positive inventory level which in that case fixes the values of the inventory variables $y^0_{i,p}$ for every $i \in L \setminus S^o$ and $p \in P$. Clearly, potential sites do not hold initial stock so that $y^0_{j,p} = 0$ for every $j \in S^o$ and $p \in P$. The statuses of the facilities over the time

Table 3 Other input parameters

Symbol	Description
$\overline{K}_i^t$	Maximum capacity of facility $i \in L$ in period $t \in T$
$\underline{K}_i^t$	Minimum required throughput at facility $i \in S$ in period $t \in T$
$\mu_{i,p}$	Amount of capacity required by one unit of product $p \in P$ at facility $i \in L$
$D_{i,p}^t$	Demand of customer/facility $i \in L$ for product $p \in P$ in period $t \in T$
B^t	Available budget in period $t \in T$
α^t	Unit return factor on capital not invested in period $t \in T \setminus \{n\}$
ϵ	Sufficiently small positive number

Table 4 Decision variables

Symbol	Description
$b_{i,p}^t$	Amount of product $p \in P$ produced/purchased by facility $i \in L$ in period $t \in T$
$x_{i,j,p}^t$	Amount of product $p \in P$ shipped from facility $i \in L$ to facility $j \in L$ $(i \neq j)$ in period $t \in T$
$y_{i,p}^t$	Amount of product $p \in P$ held in stock in facility $i \in L$ at the end of period $t \in T \cup \{0\}$; $y_{i,p}^0$ denotes the initial inventory level
$z_{i,j}^t$	Amount of capacity shifted at the beginning of period $t \in T$ from the existing facility $i \in S^c$ to a newly established facility at site $j \in S^o$
ξ^t	Amount of capital not invested in period $t \in T$
η_i^t	$= 1$ if the selectable facility $i \in S$ changes its status in period $t \in T$; 0 otherwise

horizon are ruled by the binary variables η_i^t. If an existing facility $i \in S^c$ ceases to operate at the end of period t then $\eta_i^t = 1$. Similarly, if a new facility starts to operate in site $j \in S^o$ at the beginning of period t then $\eta_j^t = 1$. Observe that a new facility can never operate in the first period since that would incur a setup cost prior to the beginning of the planning horizon. Analogously, an existing facility cannot be closed at the end of the last period since the fixed closing cost would be charged in a period beyond the time horizon. Hence, $z_{i,j}^1 = 0$ for every $i \in S^c$ and $j \in S^o$. Moreover, $\eta_i^1 = 0$ for every $i \in S^o$ and $\eta_j^n = 0$ for every $j \in S^c$.

Melo et al. (2006) proposed two alternative MILP formulations for the above problem. The model presented next is the basis for the development of the heuristic procedure to be described in Sect. 4. Note that model (P) captures several features identified by Hammami et al. (2008) that are associated with realistic relocation scenarios:

$$
\begin{aligned}
(P) \quad \text{MIN} & \sum_{t\in T}\sum_{i\in L}\sum_{p\in P} PC_{i,p}^t\, b_{i,p}^t + \sum_{t\in T}\sum_{i\in L}\sum_{j\in L\setminus\{i\}}\sum_{p\in P} TC_{i,j,p}^t\, x_{i,j,p}^t \\
& + \sum_{t\in T}\sum_{i\in L}\sum_{p\in P} IC_{i,p}^t\, y_{i,p}^t + \sum_{t\in T}\sum_{i\in S^c} OC_i^t \left(1 - \sum_{\tau=1}^{t-1}\eta_i^\tau\right) \\
& + \sum_{t\in T}\sum_{i\in S^o} OC_i^t \sum_{\tau=1}^{t}\eta_i^\tau + \sum_{t\in T}\sum_{i\in L\setminus S} OC_i^t
\end{aligned} \tag{1}
$$

s.t.

$$b_{i,p}^t + \sum_{j \in L \setminus \{i\}} x_{j,i,p}^t + y_{i,p}^{t-1} = D_{i,p}^t + \sum_{j \in L \setminus \{i\}} x_{i,j,p}^t + y_{i,p}^t,$$
$$i \in L,\ p \in P,\ t \in T, \tag{2}$$

$$\overline{K}_i^1 - \sum_{\tau=1}^{t} \sum_{j \in S^o} z_{i,j}^\tau \le \overline{K}_i^t \left(1 - \sum_{\tau=1}^{t-1} \eta_i^\tau\right), \quad i \in S^c, t \in T, \tag{3}$$

$$\sum_{\tau=1}^{t} \sum_{i \in S^c} z_{i,j}^\tau \le \overline{K}_j^t \sum_{\tau=1}^{t} \eta_j^\tau, \quad j \in S^o,\ t \in T, \tag{4}$$

$$\sum_{\tau=1}^{t} \sum_{j \in S^o} z_{i,j}^\tau + \epsilon \left(1 - \sum_{\tau=1}^{t-1} \eta_i^\tau\right) \le \overline{K}_i^1, \quad i \in S^c, t \in T, \tag{5}$$

$$\sum_{p \in P} \mu_{i,p} \left(b_{i,p}^t + \sum_{j \in L \setminus \{i\}} x_{j,i,p}^t + y_{i,p}^{t-1}\right) \le \overline{K}_i^1 - \sum_{\tau=1}^{t} \sum_{j \in S^o} z_{i,j}^\tau,$$
$$i \in S^c, t \in T, \tag{6}$$

$$\sum_{p \in P} \mu_{i,p} \left(b_{i,p}^t + \sum_{j \in L \setminus \{i\}} x_{j,i,p}^t + y_{i,p}^{t-1}\right) \le \sum_{\tau=1}^{t} \sum_{j \in S^c} z_{j,i}^\tau,$$
$$i \in S^o, t \in T, \tag{7}$$

$$\sum_{p \in P} \mu_{i,p} \left(b_{i,p}^t + \sum_{j \in L \setminus \{i\}} x_{j,i,p}^t + y_{i,p}^{t-1}\right) \le \overline{K}_i^t, \quad i \in L \setminus S,\ t \in T, \tag{8}$$

$$\sum_{p \in P} \mu_{i,p} \left(b_{i,p}^t + \sum_{j \in L \setminus \{i\}} x_{j,i,p}^t + y_{i,p}^{t-1}\right) \ge \underline{K}_i^t \left(1 - \sum_{\tau=1}^{t-1} \eta_i^\tau\right),$$
$$i \in S^c, t \in T, \tag{9}$$

$$\sum_{p \in P} \mu_{i,p} \left(b_{i,p}^t + \sum_{j \in L \setminus \{i\}} x_{j,i,p}^t + y_{i,p}^{t-1}\right) \ge \underline{K}_i^t \sum_{\tau=1}^{t} \eta_i^\tau,$$
$$j \in S^o, t \in T, \tag{10}$$

$$\sum_{t \in T} \eta_i^t \le 1, \quad i \in S, \tag{11}$$

$$\sum_{i \in S^o} FC_i^1 \left(\sum_{\tau=1}^{2} \eta_i^\tau\right) + \xi^1 = B^1, \tag{12}$$

$$\sum_{i \in S^c} \sum_{j \in S^o} MC_{i,j}^t z_{i,j}^t + \sum_{i \in S^c} SC_i^t \eta_i^{t-1} + \sum_{j \in S^o} FC_j^t \eta_j^{t+1} + \xi^t$$
$$= B^t + \alpha^{t-1} \xi^{t-1}, \quad t \in T \setminus \{1, n\}, \tag{13}$$

$$\sum_{i\in S^c}\sum_{j\in S^o} MC_{i,j}^n\, z_{i,j}^n + \sum_{i\in S^c} SC_i^n \eta_i^{n-1} + \xi^n = B^n + \alpha^{n-1}\xi^{n-1}, \tag{14}$$

$$b_{i,p}^t \geq 0, \qquad y_{i,p}^t \geq 0, \qquad x_{i,j,p}^t \geq 0, \qquad \xi^t \geq 0,$$
$$i \in L,\ j \in L \setminus \{i\}, p \in P, t \in T, \tag{15}$$

$$z_{i,j}^t \geq 0, \quad i \in S^c, j \in S^o, t \in T, \tag{16}$$

$$\eta_i^t \in \{0, 1\}, \quad i \in S, t \in T. \tag{17}$$

The objective function (1) minimizes total supply chain costs which comprise variable supply, transportation, and inventory holding costs as well as fixed facility operating costs. Constraints (2) are the usual flow conservation conditions and also ensure the satisfaction of customer demands. Inequalities (3) guarantee that only operating existing facilities can have their capacities transferred to new facilities. Constraints (4) state that a new facility can only start receiving capacity after its setup, while constraints (5) ensure that an existing facility is closed after complete removal of its capacity. In fact, note that when the first sum equals $\overline{K}_i^1$ (i.e., the initial capacity of the facility has been completely transferred) then the second term on the left-hand side will become zero, which means that the facility is removed. Capacity constraints are imposed by inequalities (6)–(8). Constraints (9)–(10) guarantee that a selectable facility operates with at least a given throughput. Constraints (11) allow the status of each selectable facility to change at most once over the time horizon. This means that a facility that is removed cannot be re-opened; and once open, a new facility cannot be closed. Conditions (12)–(14) guarantee that the available budget is invested in capacity transfers, the setup of new facilities and the removal of existing facilities upon complete relocation. The amount of capital not used in a given period earns interest and can later be invested. Finally, constraints (15)–(17) represent non-negativity and binary conditions.

4 Heuristic approach

The MILP model (P) contains two types of inherently different decisions: on the one hand, the yes/no-decisions to change the operating status of facilities (variables η_i^t) and, on the other hand, a large number of strategic and tactical supply chain decisions modeled by non-negative continuous variables. Once the binary choice for facility operation has been made, the resulting problem is linear and thus much simpler to solve. Hence, the design of solution procedures that decouple the binary decisions variables from the continuous variables is a natural approach to overcome the computational hurdle resulting from model (P) being NP-hard, and therefore, being in general extremely difficult to solve to optimality (particularly for real-size instances). Wolf and Merz (2007) use variable decoupling in their evolutionary algorithm for solving (P). To reduce the search space, several mechanisms are proposed for filtering out infeasible solutions. Unfortunately, no information is conveyed with respect to the running time and the quality of the solution obtained for a small number of test instances.

Approximation algorithms based on linear programming (LP) have been used extensively to obtain near-optimal solutions for many classes of discrete optimization problems (see, e.g., Gonzalez 2007 and Vazirani 2001 for some applications). A basic technique is to solve the linear relaxation of the integer program and then convert the fractional solution into an integer solution, trying to ensure that in the process the objective value does not deteriorate much. A variety of facility location problems have been solved efficiently by LP-rounding techniques (see, e.g., Chudak and Shmoys 2004 and Shmoys 2004). In contrast, this algorithmic approach has been scarcely applied to facility location problems in an SCM context due to the real challenges presented by this class of difficult problems. Recently, Thanh et al. (2008b) proposed an LP-rounding heuristic for a large-scale multi-period network design problem. Unfortunately, large running times are reported while solving medium-sized instances.

Our motivation for designing an LP-rounding heuristic stems not only from this being a natural approach to explore the structure of our problem, but also from the tight lower bound provided by the linear relaxation of model (P). For medium-sized instances, Melo et al. (2006) observed that on average the LP bound is within 2% of the corresponding optimal solution.

Our solution approach consists of a fast construction phase where four rounding strategies are applied to iteratively replace the fractional location variables in the LP relaxation by binary values. If during this process infeasibility arises, the incumbent solution will be repaired in the second phase. Otherwise, local search is used in an attempt to improve the quality of the feasible solution delivered by the construction phase.

4.1 Construction phase

The aim of the construction phase is to identify an initial feasible solution. Table 5 introduces the required notation.

The steps performed during this phase are summarized in Algorithm 1. The procedure starts by solving the linear relaxation of the original problem (Step 0). It is widely known that simply rounding all fractional facility status variables to their nearest integer values frequently causes constraint violation. Hence, we devised four variable fixing strategies (VFS1–VFS4) to gradually assign binary values to the location variables η_i^t, $i \in S$, $t \in T$. The procedure ends when all facility status variables are binary (Step 4).

The algorithm relies on a careful selection of variables to be made binary. Priority is given to rounding fractional values to zero as this decision usually has little impact on the network configuration. The variable fixing strategies VFS1 and VFS3 comprise selection mechanisms for rounding to zero. In contrast, fixing a facility status variable at one triggers a sequence of changes in the network configuration that in the worst case may violate several constraints. If it is decided to round some variable η_i^t with $i \in S^c$ to one then the corresponding existing facility i will cease to operate at the end of period t. This decision can only be performed if capacity and budget conditions allow moving the entire capacity of facility i to new sites until period t. On the other hand, the selection of some variable η_i^t with $i \in S^o$ to be fixed at one leads to opening a new facility in site i at the beginning of period t. This action is mainly limited by the available budget in period $t-1$ to cover the setup of the new facility. The variable fixing strategies VFS2 and VFS4 were devised to perform rounding to one.

Table 5 Notation used in Algorithm 1

Symbol	Description
H_0	Set of pairs (i, t) with $i \in S$ and $t \in T$ such that $\eta_i^t = 0$
H_1	Set of pairs (i, t) with $i \in S$ and $t \in T$ such that $\eta_i^t = 1$
H	Union of sets H_0 and H_1; the elements of set H correspond to facility status variables η_i^t that take binary values
$\overline{H}$	Set of pairs (i, t) with $i \in S$ and $t \in T$ such that $(i, t) \notin H$; the elements of set $\overline{H}$ refer to facility status variables η_i^t that take fractional values
LP_H	Linear relaxation of problem (P) with the facility status variables associated with set H taking the values 0 or 1 depending on whether they refer to H_0 or H_1
Sol_{LP}	0–1 flag indicating whether LP_H is feasible (value 1) or infeasible (value 0)
P_H	Problem (P) when *all* facility status variables η_i^t have given fixed binary values
Δ_H	Gap between the upper bound $v(P_H)$ provided by the solution of P_H and the lower bound $v(LP)$ given by the linear relaxation of (P); $\Delta_H = (v(P_H) - v(LP))/v(LP)$
$\underline{\eta}$	Threshold for variable fixing at zero
$\overline{\eta}$	Threshold for variable fixing at one
$\widetilde{\eta}$	Threshold for variable fixing at zero or one
$i_{\max}$	Maximum number of fractional facility status variables that are rounded to zero

Steps 1–3, which form the first part of Algorithm 1, focus on iteratively rounding fractional variables using pre-specified lower and upper thresholds (see the description of VFS1 and VFS2 below). Each time one or several variables $\{\eta_i^t\}_{(i,t)\in\overline{H}}$ become integer, a new LP relaxation with the remaining location variables being allowed to take values in the interval [0, 1], is solved. If variable fixing leads to an infeasible LP problem then we proceed to the second part of the algorithm, where all variables that are still fractional are made binary according to two additional strategies (VFS3 and VFS4). In particular, the parameter $i_{\max}$ limits the number of variables to be rounded to zero before exploring the possibility of fixing a location variable at one.

An alternative outcome of the first part of Algorithm 1 is a feasible fractional solution for which further rounding is not possible as the non-binary facility status variables take values within the lower and upper thresholds. In this case, we continue with the second part (Steps 5–8) by first selecting a fractional variable to be rounded to zero. The impact of this choice is evaluated by solving the remaining linear program (Step 7). If the LP relaxation is feasible then we try to round one more variable to one (Step 8). When this measure proves to be successful the variable fixing process is restarted by returning to the first part of the algorithm.

In the following sections, we describe in detail the strategies that were implemented.

4.1.1 Initialization

In view of the assumptions made in Sect. 3 with respect to the periods in which the fixed setup cost of a new facility and the fixed closing cost of an existing facility may

Algorithm 1: Construction phase

Step 0:	Initialize sets H_0 and H_1, solve LP_H, set $\mathsf{Sol}_{LP}=1$ and $k=0$
Step 1:	Apply VFS1
	If unsuccessful *then* go to Step 3
Step 2:	Solve LP_H
	If LP_H infeasible *then* set $\mathsf{Sol}_{LP}=0$ and go to Step 4
Step 3:	Apply VFS2
	If VFS1 or VFS2 successful *then* return to Step 1
....	....
Step 4:	*If* $\overline{H}=\emptyset$ *then* calculate Δ_H and STOP
Step 5:	*If* $k \le i_{\max}$ *then*
	Set $k=k+1$ and apply VFS3
	If successful *then*
	If $\mathsf{Sol}_{LP}=1$ *then* go to Step 7 *else* return to Step 4
Step 6:	Set $k=0$ and apply VFS4
	If $\mathsf{Sol}_{LP}=0$ *then* return to Step 4
Step 7:	Solve LP_H
	If LP_H infeasible *then* set $\mathsf{Sol}_{LP}=0$ and return to Step 4
Step 8:	Apply VFS2
	If successful *then* return to Step 1
	else
	Apply VFS1
	If successful *then* return to Step 2 *else* return to Step 4

be incurred, it is natural to solve the linear relaxation to problem (P) with an already fixed set of variables. Hence, $H=H_0\cup H_1$ with $H_0=\{(i,1): i\in S^o\}\cup\{(i,n): i\in S^c\}$ and $H_1=\emptyset$.

4.1.2 Variable fixing strategies

Each of the following procedures aims at rounding one or several fractional variables $\{\eta_i^t\}_{(i,t)\in\overline{H}}$ to one or zero.

VFS1 All facility status variables with fractional values not exceeding a user-defined lower threshold $\underline{\eta}$ are assumed to remain unchanged and therefore have their values rounded to zero:

$$\text{Set } \eta_i^t=0 \quad \text{for every } (i,t)\in\overline{H} \quad \text{such that} \quad \eta_i^t\le\underline{\eta}.$$

If rounding occurs then sets H_0 and H are updated accordingly.

VFS2 All facility status variables with fractional values above a given upper threshold $\overline{\eta}$ are potential candidates to be rounded to one, that is, $\eta_i^t\ge\overline{\eta}$ with $(i,t)\in\overline{H}$.

Since the resulting linear relaxation is very sensitive to variable fixing at one, at most *one* of these variables will be selected. Among the candidate variables for which feasibility of the corresponding LP relaxation is retained, the one yielding the highest objective value is chosen. If such a variable can be found, let us denote it by η_j^τ. It

Table 6 Notation used in Algorithms 2a and 2b

Symbol	Description
h_ℓ	ℓth element of set H_1 (in case $H_1 \neq \emptyset$)
$i(h_\ell)$	First component of h_ℓ representing a facility
$t(h_\ell)$	Second component of h_ℓ representing a time period
$\underline{\Delta}$	Gap threshold
$\ell_{\max}$	Maximum number of facility status variables to be investigated
$j_{\max}$	Maximum number of facility status variables that have their binary values changed from 1 to 0
$k_{\max}$	Maximum number of visited solutions starting from a given solution

follows that the pair (j, τ) is transferred from set $\overline{H}$ to set H_1. Furthermore, due to constraints (11), set H_0 is extended with pairs (j, t) for every $t \in T \setminus \{\tau\}$. Finally, set $\overline{H}$ includes those pairs (i, t) for which η_i^t take fractional values in the feasible solution to the retained LP relaxation.

VFS3 The aim of this strategy is to round to zero a fractional variable corresponding to an *existing* facility. To this end, among the pairs $(i, t) \in \overline{H}$ with $i \in S^c$, the facility status variable η_i^t with lowest fractional value is identified and fixed at zero. If such variable can be found then sets H_0 and H are updated accordingly.

VFS4 This strategy aims at setting a fractional variable corresponding to a *potential new* facility to a binary value. Let η_j^t denote the status variable with current largest fractional value such that $(j, t) \in \overline{H}$ and $j \in S^o$. If $\eta_j^t \geq \widetilde{\eta}$, with $\widetilde{\eta}$ a user-defined threshold, then $\eta_j^t = 1$, otherwise $\eta_j^t = 0$. Depending on the action implemented, one of the sets H_1 or H_0 is updated.

4.2 Repair and improvement phase

Three solution outcomes are possible at the end of the construction phase:

1. Problem P_H is infeasible, and therefore $\Delta_H = +\infty$.
2. Problem P_H is feasible and $\Delta_H > \underline{\Delta}$.
3. Problem P_H is feasible and $\Delta_H \leq \underline{\Delta}$.

The parameter $\underline{\Delta}$ denotes a pre-specified solution quality criterion. In the first case, a repair mechanism is needed to transform the initial infeasible solution into a feasible one. In the second case, although a feasible solution has been identified, its quality is unsatisfactory and thus an improvement scheme is required. In the third case, no further steps are applied since a feasible solution to the original problem (P) with a good LP gap is already available.

The aim of the second phase is to handle cases 1 and 2 simultaneously. Table 6 introduces the required notation. The algorithm is divided into two parts—2a and 2b—that are outlined below.

In the first part of the algorithm (Algorithm 2a), a non-exhaustive search of a simple neighborhood is performed. For the sake of simplicity, let us assume that the

Algorithm 2a: First part of repair and improvement phase

If $\Delta_H > \underline{\Delta}$ *then*
  For $\ell = 1, \ldots, \ell_{\max}$ *do*
    Set $t = 1$
    If $|H_1| > 0$ *then*
      Move h_ℓ from H_1 to H_0
      Set $t = t(h_\ell)$
    For $k = 1, \ldots, k_{\max}$ *do*
      Select at random facility $i \in S$ such that $i \neq i(h_\ell)$
      If $(i, t) \in H_0$ *then* move (i, t) from H_0 to H_1
      Check for new incumbent best solution and update Δ_H if necessary
      Restore H_0 and H_1

Algorithm 2b: Second part of repair and improvement phase

for $j = 2, \ldots, j_{\max}$ *do*
  If $\Delta_H > \underline{\Delta}$ *and* $|H_1| \neq 1$ *then*
    Set $t_1 = \cdots = t_j = 1$
    For $k = 1, \ldots, k_{\max}$ *do*
      If $|H_1| \geq j$ *then*
        Select at random j different elements $h_1, \ldots, h_j \in H_1$
        Move $h_1, \ldots, h_j$ from H_1 to H_0
        Set $t_1 = t(h_1), \ldots, t_j = t(h_j)$
      Select at random j different facilities $i_1, \ldots, i_j \in S$ such that $i_\ell \neq i(h_\ell)$
      for at least one ℓ $(1 \leq \ell \leq j)$
      For $\ell = 1, \ldots, j$ *do*
        If $(i_\ell, t_\ell) \in H_0$ *then* move (i_ℓ, t_ℓ) from H_0 to H_1
      Check for new incumbent best solution and update Δ_H if necessary
      Restore H_0 and H_1

current solution includes a non-empty set of facility status variables fixed at one, i.e., $\{\eta_i^t\}_{(i,t)\in H_1} \neq \emptyset$. In other words, set H_1 corresponds to facilities whose statuses change over the planning horizon. The neighborhood of this solution is defined by the following two types of exchanges:

M1. Undo the status change of some facility i in period t such that $(i, t) \in H_1$.
M2. Perform move M1 and at the same time enable another facility to have its status be changed in the same period t.

A move of type M2 corresponds in fact to a swap of two variables. The variable to be switched from zero to one is randomly selected from the set H_0. This procedure is repeated for each element of the set H_1. In case this set is empty, variable swapping is not possible; and therefore, only the second part of a move of type M2 can be performed for $t = 1$. In addition, a user-defined parameter ($k_{\max}$) controls the number of times this procedure is repeated. Whenever a new best feasible solution is identified, it becomes incumbent.

In the second part of the algorithm (Algorithm 2b), the neighborhood search is enlarged if the quality of the incumbent solution is still not satisfactory. This encompasses mutually exchanging the values of more than two facility status variables. Hence, k-swaps involve randomly selecting $k/2$ pairs from set H_1 and $k/2$ pairs from set H_0. Observe that similar to moves of type M1 in Algorithm 2a, it is possible to restrict the number of variables to be changed to one, and thus avoid drastic modifications of the network configuration which may cause constraint violation.

Algorithms 1 and 2a, 2b rely on a number of user-defined parameters whose values are dynamically modified during the whole procedure. As it is typical in heuristic development, the tuning of these parameters is a critical issue. Based on a number of empirical computational experiments, we present in Sect. 5.2 the parameter settings that best contributed to a good performance of our heuristic procedure.

5 Computational experiments

In this section, we examine the performance of the new heuristic based on considerable computational testing carried out on three sets of randomly generated instances. Although the heuristic procedure developed by Wolf and Merz (2007) also applies to model (P), a direct comparison to our heuristic is not possible as many details of the method are omitted. In addition, the nine test instances considered in Wolf and Merz (2007) are only briefly described, and so it is also not possible to reproduce them.

In the next section, we introduce the test problems. The parameter settings used by our heuristic are presented in Sect. 5.2, while the computational results are discussed in Sect. 5.3.

5.1 Test problems

We randomly generated three sets of instances varying in size, type of associated supply chain network, cost structure, and arc densities. The intervals for the random generation of input parameter values were selected with the purpose of obtaining a wide variety of instances close to real-life problems.

The first group includes 45 instances, denoted by P1–P45, and corresponds to simple networks comprising DCs and customers. The second set consists of 25 instances, denoted by P46–P70, and refers to two-echelon networks with plants and DCs. These two sets coincide with classes 1 and 2 used by Melo et al. (2006) and were generated following the procedure described by Melo et al. (2003).

Melo et al. (2006) also studied a third group of test problems associated with three-echelon networks comprising plants, central and regional DCs. However, since only a few instances of small size were considered, we decided to strengthen this group by generating 47 new test problems capturing realistic characteristics. Details about the random generation of these instances are given in Melo et al. (2009b). The parameters defining these new instances as well as their sizes can be found in Table 12 in the Appendix.

In all instances facility relocation decisions concern the DC layer(s). Furthermore, all costs follow a nondecreasing pattern over the time horizon since in our view this reflects real-life situations better, as supply chain networks are often redesigned to cope with rising costs driven by an expanding global economy. Moreover, a distinctive feature of the new set 3 is the magnitude of the facility closing costs compared

Table 7 Arc density used to generate three-echelon networks (set 3)

Source	Destination	Arc density (%)
Plants	Central DCs	70
Central DCs	Regional DCs	40
Central DCs	Central DCs	100
Central DCs	Customers	5
Regional DCs	Customers	50
Regional DCs	Regional DCs	40

Table 8 Parameter values for Algorithms 1, 2a, and 2b

Parameter	Value
$\underline{\eta}$	0.01 in the first iteration of Algorithm 1; 0.1 otherwise
$\overline{\eta}$	0.9
$\widetilde{\eta}$	0.5
$i_{\max}$	2
$\ell_{\max}$	2 if $\lvert H_1 \rvert = \emptyset$; $\lvert H_1 \rvert$ otherwise
$j_{\max}$	3
$k_{\max}$	20 in Algorithm 2a; 30 in Algorithm 2b

to facility opening costs. The former are significantly lower and may even take negative values to account for revenues due to the termination of leasing contracts or the selling of property.

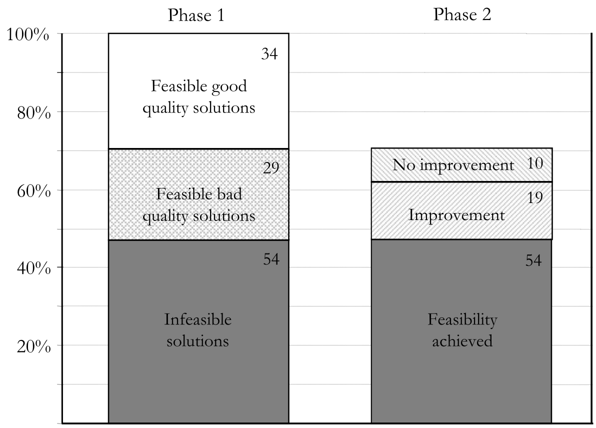
The arc density in the generated networks varies according to different criteria. Table 7 shows the baseline values for three-echelon networks (set 3). Simpler network structures (sets 1 and 2) are obtained by successively removing facility layers from this base case. Observe that in addition to inter-layer flows, products may also be shipped directly to customers as well as distributed among facilities belonging to the same layer. Finally, 70–80% of the commodities can actually be shipped over each generated arc. This limits the product flow through the network and thus mimics real-world situations. Observe that this feature results in a more tightly constrained problem (P) as the number of transportation channels available for product distribution is reduced. As a result, finding feasible solutions is an even more difficult task.

5.2 Parameter settings

To fine-tune the parameters used by the heuristic described in Sect. 4, we initially conducted a large number of empirical experiments. Table 8 presents the numerical values that best contributed to a good performance of our heuristic procedure.

In addition, the improvement phase (Algorithm 2a) is triggered by a feasible solution with an objective value that deviates more than 2.5% (i.e., $\underline{\Delta} = 0.025$) from the lower bound of the linear relaxation. The choice of this value stems from the computational experience of Melo et al. (2006) which indicates that for small instances the linear relaxation of (P) is very strong. On average the LP bound is within 2% of

Fig. 2 Outcome of phases 1 and 2 of the heuristic procedure



the corresponding optimal solution. The gap threshold $\underline{\Delta}$ is halved in Algorithm 2b when the neighborhood size is enlarged in an attempt to find a better solution.

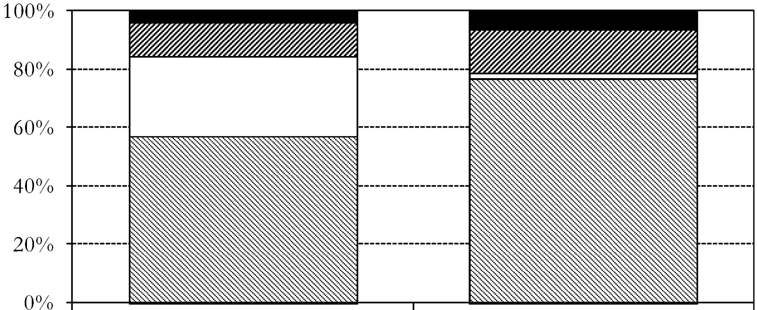
Finally, due to the random nature of the second phase of the heuristic, during which facility status variables are randomly chosen to have their values exchanged, 10 runs are performed for every instance.

5.3 Summary of results

In this section, we evaluate the performance of the new heuristic using the 117 instances described above. All experiments were conducted on a Pentium IV with a 3.2 GHz processor and 1 GB RAM. The heuristic algorithm was coded in C++. The linear relaxations LP_H associated with setting some of the facility status variables to given binary values were coded using ILOG Concert Technology 2.0 (2003) and solved with CPLEX 10.2 (ILOG CPLEX User's Manual 2007).

Figure 2 depicts the outcome of each phase of the heuristic procedure. Feasible solutions are found during the first phase for 63 of the 117 instances. Over 54% (34 out of 63) of these solutions satisfy the pre-defined quality criterion of 2.5% with respect to the deviation to the linear relaxation bound. For the remaining 54 instances, no feasible solutions were identified during the first phase. Nevertheless, the subsequent phase succeeded in delivering a feasible solution for each of these instances. In total, only 10 feasible solutions could not be further improved in the second phase of the heuristic.

Figure 3 displays the quality of the feasible solutions identified by the new heuristic. The relative percentage deviation ("LP-gap") between the objective value of these solutions and the lower bound produced by the linear relaxation of (P) is analyzed by grouping the results into four categories. The information is given separately for instances P1–P70 (sets 1 and 2) and P71–P117 (set 3) because there is an indication that the heuristic performs consistently better in the latter group. In 76.6% of the instances in set 3 (36 out of 47), the feasible solution is within 1% of the LP bound. In contrast, such high-quality solutions are only delivered to 57.1% of the instances (40 out of 70) belonging to the other two sets. Recall that instances P71–P117 refer to larger and more complex network structures whose generation was motivated by



	P1 - P70	P71 - P117
■ LP-gap > 10%	3	3
▨ 5% < LP-gap ≤ 10%	8	7
□ 1% < LP-gap ≤ 5%	19	1
▧ LP-gap ≤ 1%	40	36

Fig. 3 Deviation of the feasible solutions identified with the new heuristic from the corresponding LP bounds

the need to capture practical features of strategic supply chain planning. The number of feasible network configurations over the time horizon is usually much larger compared to sets 1 and 2 (P1–P70), thus offering diversity of choice to our heuristic procedure. In contrast, the single-echelon and two-echelon networks associated with P1–P70 comprise a limited number of feasible configurations, which seems to hinder the progress of the heuristic towards feasibility, particularly during the first phase.

Tables 9, 10, and 11 contain the detailed results obtained with the new heuristic approach. In addition, we also report our computational experience with CPLEX. In fact, over the past years, the effort invested in the development of commercial optimization engines such as CPLEX has significantly increased the possibility of tackling many complex problems. Even when an optimal solution cannot be found within acceptable computational time, it is often possible to identify a high-quality feasible solution. To verify if this also applies to our problem, we ran CPLEX with two stopping criteria: a time limit of 5 hours and a target gap. For the latter, the deviation between the best solution and the best lower bound delivered by CPLEX is at most 1%. Thus, we use CPLEX as an *alternative heuristic method*. Our choice is supported by Cordeau et al. (2006) who argue that solving a real-life problem to optimality is usually not meaningful due to errors contained in the data estimates. Since the error margin tends to be larger than 1%, these authors claim that it is adequate to run the optimization solver until a feasible solution within 1% optimality is identified. Ambrosino and Scutellà (2005) and Melkote and Daskin (2001) also use an optimization solver heuristically. Hence, for the best feasible solution that the solver produces there is a guarantee of the maximum deviation from optimality.

The first column of Tables 9–11 indicates the test instance. The columns under the heading *Heuristic* report the results delivered by the two-phase heuristic. The symbol * in column 2 highlights those instances in which the construction phase succeeded in identifying a feasible solution. When this does not occur, the repair mechanism of phase 2 is employed and repeated 10 times. In this case, the number of runs of phase 2 in which a feasible solution has been identified is reported in column 2. Column 3 presents the value of $(UB_H - LB)/LB \cdot 100\%$ with UB_H denoting the objective value of the identified feasible solution and LB the optimal LP value. Recall that since 10 runs of phase 2 are performed, the displayed values correspond to the average over the runs yielding a feasible solution. The CPU times (in seconds) required in phase 1 and phase 2 of the heuristic are reported in columns 4 and 5, respectively. If during the construction phase a solution is identified within 2.5% of the LP value then the improvement scheme in phase 2 is not performed. In this case, the symbol "–" appears in column 5. The total CPU time (in seconds) is given in column 6.

The columns under the heading *CPLEX* describe the results obtained with this optimization solver. Column 7 (*opt. gap*) presents the gap reported by CPLEX at the time of termination. Note that the values shown in this column may overestimate the true integrality gaps due to stopping the search as soon as a feasible solution within 1% optimality is identified. Column 8 shows the value of $(UB - LB)/LB \cdot 100\%$ with UB denoting the objective value of the best feasible solution and LB the optimal LP value. The total CPU time (in seconds) required by CPLEX is reported in column 9.

Finally, the columns under the heading *Heuristic/CPLEX* present a comparison between the two solution procedures. Column 10 displays the values of UB_H/UB, while column 11 indicates the ratios between the CPU times required by the heuristic and CPLEX.

A close examination of Tables 9–11 reveals that the heuristic delivers solutions within acceptable time. For single and two-echelon networks, a good feasible solution is obtained in less than one minute. Instances associated with three-echelon networks require, as expected, larger CPU time, but on average not more than seven minutes. These results support our view that the heuristic is suitable to be used when network re-optimization associated with "what-if" analyses needs to be performed. Note that in 4% of the instances the prescribed time limit of 5 hours is reached by CPLEX. These instances (P77, P90, P91, P98, and P99) belong to the problem set comprising three-echelon networks.

A further observation concerns the large LP gaps that the solutions obtained by the new heuristic approach exhibit in five instances (P26, P28, P77, P98, and P99). It seems that the effectiveness of the construction phase to find good initial solutions declines when the linear relaxation bound is poor. Another possible explanation is that the linear relaxation solution has variables very close to "1" or "0" while their optimal integer values are exactly the opposite. Although unlikely, this case may occur and is evidenced by instances P26 and P28. Instances P98 and P99, which have 10 products and a longer planning horizon of 8 periods in common, proved to be very hard to solve. The results obtained contrast significantly with all other instances belonging to set 3. Not only are the LP gaps of the feasible solutions identified by the heuristic above 100%, but also CPLEX reaches the prescribed time limit of 5 hours with very large MIP gaps (50.52%, resp. 43.17%). A careful examination of the branch-and-cut tree produced by CPLEX reveals that the optimization progress is hindered by the poor quality of the lower bounds obtained during the search. Nevertheless, it is interesting to see that the feasible solution identified by the heuristic for P99 is better

Table 9 Results obtained for instances in set 1 (single-echelon networks); all gaps in % and all CPU times in seconds

Instance	Heuristic					CPLEX			Heuristic/CPLEX	
	# sol.	LP gap	CPU 1	CPU 2	Total CPU	opt. gap	LP gap	CPU	sol. ratio	CPU ratio
P1	10	2.51	0.56	6.54	7.11	0.74	1.60	6.36	1.01	1.12
P2	*	3.96	0.55	7.90	8.45	0.98	3.15	8.94	1.01	0.95
P3	*	6.73	1.00	2.82	3.82	0.99	3.61	8.81	1.03	0.43
P4	*	4.33	0.83	25.35	26.18	0.33	3.31	9.36	1.01	2.80
P5	10	6.86	0.84	4.83	5.68	0.97	3.47	19.34	1.03	0.29
P6	10	8.59	1.38	9.21	10.59	0.99	5.33	121.17	1.03	0.09
P7	*	5.37	1.73	16.15	17.89	0.95	3.79	13.81	1.02	1.29
P8	10	13.38	0.77	4.74	5.50	0.21	0.46	10.88	1.13	0.51
P9	*	1.82	2.06	0.00	2.06	0.83	1.68	14.22	1.00	0.15
P10	10	0.07	0.53	1.10	1.63	0.03	0.05	6.36	1.00	0.26
P11	*	0.20	3.39	–	3.39	0.30	0.33	7.59	1.00	0.45
P12	10	0.63	1.83	3.43	5.25	0.74	0.02	8.56	1.00	0.61
P13	*	6.07	3.55	34.38	37.92	0.99	0.01	156.94	1.02	0.24
P14	*	1.46	4.15	–	4.15	0.74	0.01	42.33	1.00	0.10
P15	*	2.87	3.80	49.17	52.97	0.99	0.12	45.17	1.00	1.17
P16	*	0.28	5.30	7.45	12.75	0.49	0.49	9.83	1.00	1.30
P17	*	0.27	6.59	–	6.59	0.32	0.41	17.84	1.00	0.37
P18	*	2.17	6.22	34.48	40.70	0.60	1.47	63.31	1.01	0.64
P19	10	0.17	4.64	4.56	9.20	0.07	0.13	44.33	1.00	0.21
P20	*	2.50	6.77	30.30	37.07	0.69	1.54	71.88	1.01	0.52
P21	*	0.34	1.64	–	1.64	0.38	0.56	8.69	1.00	0.19
P22	*	2.24	2.67	15.19	17.87	0.93	1.58	15.89	1.01	1.12
P23	*	1.82	2.31	0.00	2.31	0.45	1.24	14.28	1.01	0.16
P24	*	0.45	1.61	–	1.61	0.48	0.56	4.20	1.00	0.38
P25	*	2.27	3.12	0.00	3.12	0.60	1.82	19.97	1.00	0.16
P26	9	78.09	3.20	10.52	13.72	0.95	1.85	19.14	1.75	0.72
P27	*	1.72	3.42	0.00	3.42	0.70	1.55	22.44	1.00	0.15
P28	*	34.86	3.25	7.65	10.90	0.91	0.47	12.45	1.33	0.88
P29	*	4.88	2.30	27.39	29.69	0.52	0.02	12.31	1.02	2.41
P30	*	0.01	0.69	–	0.69	0.04	0.02	4.34	1.00	0.16
P31	*	0.02	0.67	–	0.67	0.04	<0.01	4.34	1.00	0.15
P32	*	0.02	0.80	–	0.80	0.02	0.02	6.88	1.00	0.12
P33	*	0.01	0.49	–	0.49	0.02	0.01	3.94	1.00	0.12
P34	10	0.02	0.42	1.14	1.56	0.03	0.01	2.91	1.00	0.54
P35	*	0.41	2.69	3.36	6.05	0.47	0.12	21.03	1.00	0.29
P36	10	0.59	1.97	6.59	8.56	0.94	0.95	10.73	1.00	0.80
P37	10	6.60	4.80	25.17	29.97	0.87	4.67	124.33	1.02	0.24
P38	10	1.02	2.47	8.03	10.50	0.91	1.13	13.20	1.00	0.80
P39	*	6.84	6.30	29.69	35.99	1.00	4.81	352.31	1.02	0.10
P40	*	0.02	2.16	–	2.16	0.05	0.05	9.09	1.00	0.24

Table 9 (*Continued*)

Instance	Heuristic					CPLEX			Heuristic/CPLEX	
	# sol.	LP gap	CPU 1	CPU 2	Total CPU	opt. gap	LP gap	CPU	sol. ratio	CPU ratio
P41	*	0.01	2.22	–	2.22	0.01	0.01	26.14	1.00	0.08
P44	*	0.03	1.97	–	1.97	0.09	0.09	9.70	1.00	0.20
P43	10	0.03	2.12	2.76	4.88	0.03	0.03	9.33	1.00	0.52
P44	*	0.07	3.44	–	3.44	0.06	0.06	9.00	1.00	0.38
P45	*	0.80	0.14	–	0.14	0.79	0.80	1.31	1.00	0.11
Average		4.74	2.52	8.44	10.96	0.54	1.19	31.67	–	–
Min.		0.01	0.14	0.00	0.14	0.01	0.00	1.31	–	–
Max.		78.09	6.77	49.17	52.97	1.00	5.33	352.31	–	–

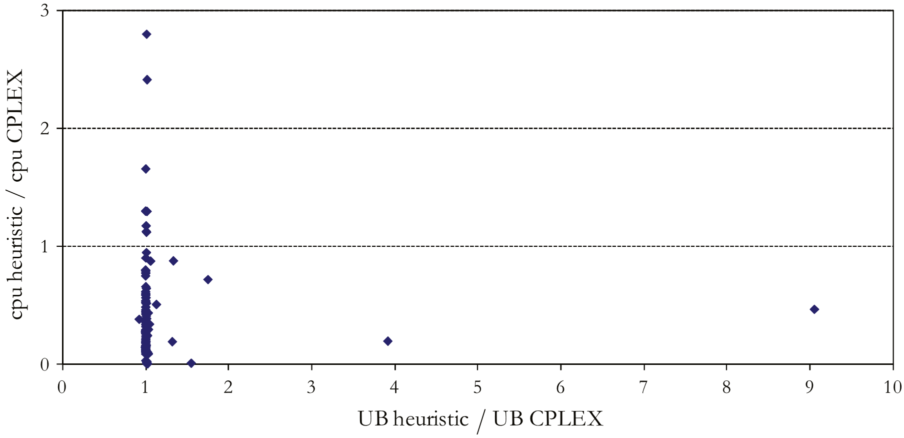


Fig. 4 Comparison between the feasible solutions identified by the heuristic and by CPLEX

than that delivered by CPLEX. Moreover, the heuristic requires significantly less time than CPLEX to find feasible solutions to these hard problems, P98 and P99 (see also Fig. 4).

In order to gain a better insight on how the results of the new methodology compare with those obtained with an optimization solver, Fig. 4 depicts the values presented in the last two columns of Tables 9–11. Recall that CPLEX is used as an *alternative heuristic method*. The horizontal axis of Fig. 4 represents the ratios between the objective values of the solutions identified by the heuristic ("UB heuristic") and those delivered by CPLEX. The vertical axis displays the ratios between the CPU times of the heuristic and those of CPLEX. Since phase 2 of the heuristic is performed 10 times, the average upper bound and the average CPU time obtained in this phase are considered. Moreover, note that a ratio lower than one indicates that the heuristic outperforms CPLEX.

Regarding the solution quality, Fig. 4 reveals that the performance of the heuristic and of CPLEX are comparable except for a few instances. Since CPLEX identifies a

Table 10 Results obtained for instances in set 2 (two-echelon networks), all gaps in % and all CPU times in seconds

Instance	Heuristic					CPLEX			Heuristic/CPLEX	
	# sol.	LP gap	CPU 1	CPU 2	Total CPU	opt. gap	LP gap	CPU	sol. ratio	CPU ratio
P46	*	0.01	1.53	–	1.53	0.02	0.02	5.91	1.00	0.26
P47	*	0.01	2.38	–	2.38	0.02	0.02	6.53	1.00	0.36
P48	9	0.02	0.70	2.62	3.33	0.47	0.47	12.02	1.00	0.28
P49	10	0.02	0.91	1.97	2.87	0.02	0.02	6.25	1.00	0.46
P50	10	0.01	1.70	1.84	3.55	0.02	0.02	6.00	1.00	0.59
P51	*	<0.01	1.22	–	1.22	0.00	<0.01	8.39	1.00	0.15
P52	10	0.01	1.08	2.88	3.95	0.01	0.02	9.11	1.00	0.43
P53	9	0.01	0.58	3.06	3.64	0.01	0.01	7.83	9.05	0.46
P54	10	0.09	1.08	1.75	2.83	0.01	0.01	5.48	1.00	0.52
P55	*	1.27	0.28	–	0.28	0.00	0.12	12.39	1.01	0.02
P56	10	<0.01	0.42	1.51	1.93	0.00	<0.01	4.56	1.00	0.42
P57	*	<0.01	1.30	–	1.30	0.00	<0.01	9.09	1.00	0.14
P58	10	<0.01	0.52	2.42	2.94	0.00	<0.01	9.28	1.00	0.32
P59	*	<0.01	1.22	–	1.22	0.00	<0.01	8.58	1.00	0.14
P60	10	<0.01	0.73	1.96	2.69	0.00	<0.01	9.24	1.00	0.29
P61	10	6.12	3.55	19.80	23.35	0.16	0.26	26.73	1.06	0.87
P62	*	1.40	3.17	–	3.17	0.88	1.52	24.28	1.00	0.13
P63	*	1.59	4.38	0.00	4.38	0.80	1.58	35.30	1.00	0.12
P64	*	1.31	4.99	12.95	17.93	0.55	1.11	31.84	1.00	0.56
P65	*	2.12	5.52	0.00	5.52	0.76	1.66	46.97	1.00	0.12
P66	10	0.01	2.50	5.40	7.90	0.01	0.01	13.16	1.00	0.60
P67	10	0.02	4.82	5.21	10.02	0.02	0.03	17.17	1.00	0.58
P68	10	0.02	2.34	4.31	6.65	0.03	0.03	14.80	1.00	0.45
P69	*	0.01	4.55	–	4.55	0.01	0.02	16.41	1.00	0.28
P70	*	0.01	4.51	–	4.51	0.00	0.01	17.17	1.00	0.26
Average		0.56	2.24	2.71	4.95	0.15	0.28	14.58	–	–
Min.		<0.01	0.28	0.00	0.28	0.00	0.00	4.56	–	–
Max.		6.12	5.52	19.80	23.35	0.88	1.66	46.97	–	–

feasible solution within 1% of optimality in 95.7% of the test problems, we realize that the new heuristic provides solutions within an acceptable optimality range. With respect to the CPU times, Fig. 4 indicates that the new methodology is significantly faster than CPLEX. In fact, the results reported in Tables 9–11 show that in each problem set, the CPU times are reduced by a factor of 3 when the heuristic is used. Although seven instances in set 1 (i.e., associated with single-echelon networks) exhibit CPU ratios above 1.0, the heuristic required less than one minute in each one of them. For the more complex networks (set 3), a single instance (P116) consumed more CPU time with the heuristic than with CPLEX. Nevertheless, a good solution was identified in this case within two minutes.

Table 11 Results obtained for instances in set 3 (three-echelon networks); all gaps in % and all CPU times in seconds

Instance	Heuristic					CPLEX			Heuristic/CPLEX	
	# sol.	LP gap	CPU 1	CPU 2	Total CPU	opt. gap	LP gap	CPU	sol. ratio	CPU ratio
P71	*	0.02	3.14	–	3.14	0.77	0.79	21.89	0.99	0.14
P72	*	0.25	12.59	–	12.59	0.64	0.66	46.81	1.00	0.27
P73	*	0.03	70.68	–	70.68	0.03	0.03	640.42	1.00	0.11
P74	*	0.03	323.77	–	323.77	0.27	0.28	432.88	1.00	0.75
P75	*	5.81	1.56	17.02	18.58	0.95	4.72	48.11	1.01	0.39
P76	*	0.11	4.45	–	4.45	0.10	0.13	150.84	1.00	0.03
P77	*	66.02	22.88	115.06	137.93	3.16	7.29	18000.13	1.55	0.01
P78	*	2.06	48.14	0.00	48.14	0.75	0.78	2344.24	1.01	0.02
P79	*	0.01	7.33	–	7.33	0.00	0.01	65.66	1.00	0.11
P80	7	0.01	17.98	36.50	54.48	0.01	0.01	148.33	1.00	0.37
P81	10	0.97	25.77	68.37	94.13	0.84	0.85	269.28	1.00	0.35
P82	7	0.25	196.37	63.25	259.62	0.01	0.01	1332.89	3.91	0.19
P83	3	0.02	4.20	15.24	19.45	0.01	0.02	36.91	1.00	0.53
P84	*	0.02	13.77	–	13.77	0.01	0.02	48.28	1.00	0.29
P85	10	0.06	51.98	10.94	62.92	0.02	0.03	236.22	1.00	0.22
P86	10	6.28	2.31	7.58	9.90	1.00	5.19	704.20	1.01	0.00
P87	10	6.30	2.36	7.67	10.03	1.00	5.19	707.25	1.01	0.01
P88	*	9.42	9.05	29.81	38.85	1.00	6.98	11376.80	1.02	0.00
P89	*	8.98	26.92	372.41	399.33	5.52	8.08	18001.03	1.01	0.02
P90	6	9.79	145.31	5683.02	5828.34	6.72	9.15	18001.08	1.01	0.32
P91	7	<0.01	8.97	7.74	16.71	0.00	<0.01	21.56	1.00	0.78
P92	4	<0.01	13.22	50.20	63.42	0.00	<0.01	80.45	1.00	0.79
P93	5	0.01	41.09	74.27	115.36	0.00	<0.01	128.23	1.00	0.90
P94	10	0.09	8.16	6.68	14.83	0.00	<0.01	97.34	1.00	0.15
P95	9	0.21	23.59	14.15	37.74	0.00	<0.01	348.17	1.00	0.11
P96	*	0.02	8.83	–	8.83	0.01	0.02	50.25	1.00	0.18
P97	*	6.95	30.26	135.48	165.74	0.99	2.38	490.53	1.04	0.34
P98	*	109.67	2278.66	4559.75	6838.41	50.52	127.66	18001.05	0.92	0.38
P99	*	160.05	933.99	2443.33	3377.32	43.17	97.54	18000.42	1.32	0.19
P100	10	0.05	4.70	6.99	11.69	0.00	<0.01	113.03	1.00	0.10
P101	10	<0.01	15.91	21.50	37.41	0.00	<0.01	204.98	1.00	0.18
P102	4	<0.01	9.74	17.32	27.05	0.00	<0.01	71.84	1.00	0.38
P103	*	<0.01	14.09	–	14.09	0.00	<0.01	116.24	1.00	0.12
P104	*	0.01	6.80	–	6.80	0.01	0.01	45.25	1.00	0.15
P105	9	0.01	18.78	28.50	47.28	0.01	0.01	84.27	1.00	0.56
P106	2	<0.01	14.05	30.70	44.75	0.00	<0.01	68.13	1.00	0.66
P107	4	<0.01	24.19	75.65	99.84	0.00	<0.01	278.53	1.00	0.36
P108	2	0.55	15.23	55.62	70.85	0.00	<0.01	174.53	1.01	0.41
P109	10	<0.01	42.20	16.57	58.77	0.00	<0.01	377.98	1.00	0.16
P110	*	<0.01	21.17	–	21.17	0.00	<0.01	161.67	1.00	0.13

Table 11 (*Continued*)

Instance	Heuristic					CPLEX			Heuristic/CPLEX	
	# sol.	LP gap	CPU 1	CPU 2	Total CPU	opt. gap	LP gap	CPU	sol. ratio	CPU ratio
P111	9	0.33	49.29	40.86	90.15	0.00	<0.01	469.17	1.00	0.19
P112	6	<0.01	19.91	40.57	60.48	0.00	<0.01	103.44	1.00	0.58
P113	1	<0.01	73.16	155.21	228.37	0.00	<0.01	296.59	1.00	0.77
P114	6	<0.01	21.69	22.35	44.04	0.00	<0.01	91.19	1.00	0.48
P115	4	<0.01	55.70	79.50	135.21	0.00	<0.01	403.91	1.00	0.33
P116	1	<0.01	41.00	75.02	116.01	0.00	<0.01	70.11	1.00	1.65
P117	1	<0.01	55.55	115.76	171.31	0.00	<0.01	353.44	1.00	0.48
Average		8.39	102.99	414.30	411.12	2.50	5.91	2410.97	–	–
Min.		0.01	1.56	0.00	2.31	0.00	0.00	21.56	–	–
Max.		160.05	2278.66	5683.02	6838.41	50.52	127.66	18001.08	–	–

6 Conclusions

In this paper, we proposed an LP-based heuristic approach for a multi-period logistics network redesign problem. The underlying model captures key features relevant to strategic supply chain planning. Our results suggest that the new heuristic approach is able to solve realistically sized problem instances within acceptable computational time. In particular, an effective repair mechanism is employed when a feasible solution is not identified during the construction phase. Moreover, a local improvement step is able to deliver good-quality solutions.

A clear advantage of the new heuristic procedure is its flexibility to handle model extensions related to changing capacity requirements over the planning horizon. When growing future demand is anticipated, additional network restructuring measures need to be adopted. The latter may result in extending the capacity of existing facilities and/or establishing additional facilities. Observe that in our model the overall capacity of the network does not change over the time periods. Although capacity shifts from existing to new locations are modeled, capacity expansion scenarios are not addressed by formulation (P). Melo et al. (2006) propose an extension to (P) to deal with this case. The opposite occurs when markets face declining demand, e.g., due to economic downturns or because products reach their end of life. Melo et al. (2006) also extend model (P) to handle the network capacity reduction case. In some applications, capacity transfer sizes are restricted to discrete amounts as opposed to the continuous case addressed in (P). Modular capacity shifts are also a natural extension to model (P) and are briefly discussed in Melo et al. (2006).

Finally, the inclusion of uncertainty in the problem data, namely, in the costs, demands and interest rates, is a relevant research topic that deserves further investigation. Within the context of facility location problems in SCM, several papers can be found in the literature that propose exact approaches (e.g., Ahmed et al. 2003; Alonso-Ayuso et al. 2003; Mitra et al. 2006; Santoso et al. 2005; Schütz et al. 2009). Ahmed and Sahinidis (2003) developed a linear programming based scheme for a capacity expansion problem, although no location decisions were explicitly addressed.

The possibility of considering such a type of approach for a stochastic version of the problem addressed in this paper is still an open issue.

Acknowledgements This research was partly supported by the German Academic Exchange Service (DAAD) under the program *PPP-Acções Integradas Luso-Alemãs/DAAD-GRICES*. The research of the third author was also partly supported by National Funding from FCT—Fundação para a Ciência e a Tecnologia, under the project PEst-OE/MAT/UI0152 (CIO/FCUL) and SFRH/BSAB/799/2008. This support is gratefully acknowledged. The authors also thank Rafael Velásquez for his valuable help in generating the test instances and implementing the heuristic procedure.

Appendix: Characteristics of the instances

Table 12 describes the characteristics and sizes of the new instances (set 3).

Table 12 Characteristics and sizes of instances associated with three-echelon networks with 5 plants

Instance	Customers	Selectable facilities				Periods	Products	Size		
	$\|C\|$	Existing DCs		New DCs		$\|T\|$	$\|P\|$	# Variables		
		Central	Regional	Central	Regional			Cont.	Integer	# Constraints
		$\|S_c^c\|$	$\|S_r^c\|$	$\|S_c^o\|$	$\|S_r^o\|$					
P71	100	4	10	8	20	3	5	28960	126	2985
P72	100	4	10	8	20	4	5	38613	168	3966
P73	100	4	10	8	20	6	5	57919	252	5928
P74	100	4	10	8	20	8	5	77225	336	7890
P75	50	4	10	8	20	3	5	19273	126	2085
P76	50	4	10	8	20	4	5	25697	168	2766
P77	50	4	10	8	20	6	5	38545	252	4128
P78	50	4	10	8	20	8	5	51393	336	5490
P79	200	4	10	8	20	3	5	48343	126	4785
P80	200	4	10	8	20	4	5	64457	168	6366
P81	200	4	10	8	20	6	5	96685	252	9528
P82	200	4	10	8	20	8	5	128913	336	12690
P83	100	8	20	12	30	3	5	55294	210	3727
P84	100	8	20	12	30	4	5	73725	280	4946
P85	100	8	20	12	30	6	5	110587	420	7384
P86	50	8	20	12	30	3	5	40147	210	2827
P87	50	8	20	12	30	3	5	40147	210	2827
P88	50	8	20	12	30	4	5	53529	280	3746
P89	50	8	20	12	30	6	5	80293	420	5584
P90	50	8	20	12	30	8	5	107057	560	7422
P91	200	8	20	12	30	3	5	85597	210	5527
P92	200	8	20	12	30	4	5	114129	280	7346
P93	200	8	20	12	30	6	5	171193	420	10984
P94	100	4	10	8	20	3	10	56743	126	5190

Table 12 (*Continued*)

Instance	Customers	Selectable facilities				Periods	Products	Size		
	$\lvert C\rvert$	Existing DCs		New DCs		$\lvert T\rvert$	$\lvert P\rvert$	# Variables		
		Central	Regional	Central	Regional			Cont.	Integer	# Constraints
		$\lvert S_c^c\rvert$	$\lvert S_r^c\rvert$	$\lvert S_c^o\rvert$	$\lvert S_r^o\rvert$					
P95	100	4	10	8	20	4	10	75657	168	6906
P96	50	4	10	8	20	3	10	37366	126	3540
P97	50	4	10	8	20	4	10	49821	168	4706
P98	50	4	10	8	20	8	10	99641	336	9370
P99	100	4	10	8	20	8	10	151313	336	13770
P100	200	4	10	8	20	3	10	95506	126	8490
P101	200	4	10	8	20	4	10	127341	168	11306
P102	100	8	20	12	30	3	10	107059	210	6352
P103	100	8	20	12	30	4	10	142745	280	8446
P104	50	8	20	12	30	3	10	76762	210	4702
P105	50	8	20	12	30	4	10	102349	280	6246
P106	200	8	20	12	30	3	10	167662	210	9652
P107	200	8	20	12	30	4	10	223549	280	12846
P108	100	4	10	8	20	3	20	112309	126	9600
P109	100	4	10	8	20	4	20	149745	168	12786
P110	50	4	10	8	20	3	20	73552	126	6450
P111	50	4	10	8	20	4	20	98069	168	8586
P112	100	8	20	12	30	3	20	210589	210	11602
P113	100	8	20	12	30	4	20	280785	280	15446
P114	50	8	20	12	30	3	20	149992	210	8452
P115	50	8	20	12	30	4	20	199989	280	11246
P116	200	8	20	12	30	3	20	331792	210	17902
P117	50	8	20	12	30	3	50	369682	210	19702

References

Ahmed S, Sahinidis NV (2003) An approximate scheme for stochastic integer programming arising in capacity expansion. Oper Res 51:461–471

Ahmed S, King A, Parija G (2003) A multi-stage stochastic integer programming approach for capacity expansion under uncertainty. J Glob Optim 26:3–24

Alonso-Ayuso A, Escudero LF, Garín A, Ortuño MT, Pérez G (2003) An approach for strategic supply chain planning under uncertainty based on stochastic 0–1 programming. J Glob Optim 26:97–124

Ambrosino D, Scutellà MG (2005) Distribution network design: new problems and related models. Eur J Oper Res 165:610–624

Ballou RH (1968) Dynamic warehouse location analysis. J Mark Res 5:271–276

Canel C, Khumawala BM, Law J, Loh A (2001) An algorithm for the capacitated, multi-commodity multi-period facility location problem. Comput Oper Res 28:411–427

Chudak F, Shmoys D (2004) Improved approximation algorithms for the uncapacitated facility location problem. SIAM J Comput 33:1–25

Cordeau JF, Pasin F, Solomon MM (2006) An integrated model for logistics network design. Ann Oper Res 144:59–82

Daskin M, Snyder L, Berger R (2005) Facility location in supply chain design. In: Langevin A, Riopel D (eds) Logistics systems: design and optimization. Springer, Berlin, pp 39–65. Chap 2
Goetschalckx M, Vidal CJ, Dogan K (2002) Modeling and design of global logistics systems: a review of integrated strategic and tactical models and design algorithms. Eur J Oper Res 143:1–18
Gonzalez TF (ed) (2007) Handbook of approximation algorithms and metaheurististics. Chapman & Hall/CRC, London/Boca Raton
Hammami R, Frein Y, Hadj-Alouane AB (2008) Supply chain design in the delocalization context: relevant features and modeling tendencies. Int J Prod Econ 113:641–656
Harrison TP (2004) Principles for the strategic design of supply chains. In: Harrison TP, Lee HL, Neale JJ (eds) The practice of supply chain management: where theory and application converge. Springer, New York, pp 3–12. Chap 1
Hinojosa Y, Puerto J, Fernández FR (2000) A multiperiod two-echelon multi-commodity capacitated plant location problem. Eur J Oper Res 123:271–291
Hinojosa Y, Kalcsics J, Nickel S, Puerto J, Velten S (2008) Dynamic supply chain design with inventory. Comput Oper Res 35:373–391
ILOG Concert Technology 2.0 User's Manual (2003) ILOG, Inc, Incline Village, Nevada: www.ilog.com/products/optimization/tech/concert.cfm
ILOG CPLEX User's Manual (2007) ILOG, Inc, Incline Village, Nevada: www.ilog.com/products/cplex
Julka N, Baines T, Tjahjono B, Lendermann P, Vitanov V (2007) A review of multi-factor capacity expansion models for manufacturing plants: searching for a holistic decision aid. Int J Prod Econ 106:607–621
Ko HJ, Evans GW (2007) A genetic algorithm-based heuristic for the dynamic integrated forward/reverse logistics network for 3PLs. Comput Oper Res 34:346–366
Meixell MJ, Gargeya VB (2005) Global supply chain design: a literature review and critique. Transp Res, Part E, Logist Transp Rev 41:531–550
Melachrinoudis E, Min H (2000) The dynamic relocation and phase-out of a hybrid, two-echelon plant/warehousing facility: a multiple objective approach. Eur J Oper Res 123:1–15
Melkote S, Daskin MS (2001) Capacitated facility location/network design problems. Eur J Oper Res 129:481–495
Melo MT, Nickel S, Saldanha da Gama F (2003) Large-scale models for dynamic multi-commodity capacitated facility location. Technical Report 58, Fraunhofer Institute for Industrial Mathematics (ITWM), Kaiserslautern, Germany. Available at www.itwm.fraunhofer.de/fileadmin/ITWM-Media/Zentral/Pdf/Berichte_ITWM/2003/bericht58.pdf
Melo MT, Nickel S, Saldanha da Gama F (2006) Dynamic multi-commodity capacitated facility location: a mathematical modeling framework for strategic supply chain planning. Comput Oper Res 33:181–208
Melo MT, Nickel S, Saldanha da Gama F (2009a) Facility location and supply chain management—a review. Eur J Oper Res 196:401–412
Melo MT, Nickel S, Saldanha da Gama F (2009b) An LP-rounding heuristic to solve a multi-period facility relocation problem. Technical Report 168, Fraunhofer Institute for Industrial Mathematics (ITWM), Kaiserslautern, Germany. Available at www.itwm.fraunhofer.de/fileadmin/ITWM-Media/Zentral/Pdf/Berichte_ITWM/2009/bericht_168.pdf
Mitra G, Poojari C, Sen S (2006) Strategic and tactical planning models for supply chains: an application of stochastic mixed integer programming. In: Appa G, Pitsoulis L, Williams HP (eds) Handbook on modelling for discrete optimization. Springer, Berlin, pp 227–264. Chap 8
Santoso T, Ahmed S, Goetschalckx M, Shapiro A (2005) A stochastic programming approach for supply chain network design under uncertainty. Eur J Oper Res 167:96–115
Schütz P, Tomasgard A, Ahmed S (2009) Supply chain design under uncertainty using sample average approximation and dual decomposition. Eur J Oper Res 199:409–419
Shmoys D (2004) The design and analysis of approximation algorithms: facility location as a case study. In: Hosten S, Lee J, Thomas R (eds) Trends in optimization, proceedings of symposia in applied mathematics, vol 61. American Mathematical Society, Providence, pp 85–97
Simchi-Levi D, Chen X, Bramel J (2005) The logic of logistics: theory, algorithms and applications for logistics and supply chain management. Springer series in operations research and financial engineering. Springer, New York
Srivastava SK (2008) Network design for reverse logistics. Omega 36:535–548
Thanh PN, Bostel N, Péton O (2008a) A dynamic model for facility location in the design of complex supply chains. Int J Prod Econ 113:678–693

Thanh PN, Bostel N, Péton O (2008b) A linear relaxation-based heuristic for logistics network design. In: 7e Conférence Francophone de MOdélisation et SIMulation—MOSIM'08, 31 March–2 April, Paris, France

Troncoso JJ, Garrido RA (2005) Forestry production and logistics planning: an analysis using mixed-integer programming. For Policy Econ 7:625–633

Vazirani V (2001) Approximation algorithms. Springer, New York

Vila D, Martel A, Beauregard R (2006) Designing logistics networks in divergent process industries: a methodology and its application to the lumber industry. Int J Prod Econ 102:358–378

Wilhelm W, Liang D, Rao B, Warrier D, Zhu X, Bulusu S (2005) Design of international assembly systems and their supply chains under NAFTA. Transp Res, Part E, Logist Transp Rev 41:467–493

Wolf S, Merz P (2007) A hybrid method for solving large-scale supply chain problems. In: Cotta C, van Hemert J (eds) Proceedings of the 7th European conference on evolutionary computation in combinatorial optimization. Lecture notes in computer science, vol 4446. Springer, Berlin, pp 219–228