INFORMS Journal on Computing
Vol. 19, No. 1, Winter 2007, pp. 14–26
ISSN 1091-9856 | EISSN 1526-5528 | 07 | 1901 | 0014

DOI 10.1287/ijoc.1050.0151

A Heuristic Approach to the Multi-Period Single-Sourcing Problem with Production and Inventory Capacities and Perishability Constraints

Ravindra K. Ahuja
Department of Industrial and Systems Engineering, University of Florida, P.O. Box 116595,
Gainesville, Florida 32611-6595, USA, ahuja@ufl.edu

Wei Huang
Innovative Scheduling, Gainesville Technology Enterprise Center (GTEC), 2153 Hawthorne Road Suite 128,
Gainesville, Florida 32641, USA, huang@innovativescheduling.com

H. Edwin Romeijn
Department of Industrial and Systems Engineering, University of Florida, P.O. Box 116595,
Gainesville, Florida 32611-6595, USA, romeijn@ise.ufl.edu

Dolores Romero Morales
Saïd Business School, University of Oxford, Park End Street, Oxford OX1 1HP, United Kingdom,
Dolores.Romero-Morales@sbs.ox.ac.uk

The multi-period single-sourcing problem that we address in this paper can be used as a tool for evaluating logistics network designs in a dynamic environment. We consider the assignment of retailers to facilities, taking into account the timing, location, and size of production and inventories, in the presence of various types of constraints. We formulate the problem as a nonlinear assignment problem, and develop efficient algorithms for solving the capacitated lot-sizing subproblems that form the objective function of this formulation. We propose a greedy heuristic, and prove that this heuristic is asymptotically optimal in a probabilistic sense when retailer demands share a common seasonality pattern. In addition, we develop an efficient implementation of the very-large-scale-neighborhood-search method that can be used to improve the greedy solution. We perform extensive tests on a set of randomly generated problem instances, and conclude that our approach produces very high quality solutions in limited time.

Key words: production and inventory planning; capacity constraints; heuristics
History: Accepted by Michel Gendreau, Area Editor for Heuristic Search and Learning; received April 2003; revised November 2004, March 2005; accepted April 2005.

1. Introduction

The tendency to move towards global supply chains and the shortening of the product life cycle cause companies to consider redesigning their logistics distribution network. Most quantitative models for the strategic problem of evaluating the layout of a logistics distribution network (usually with respect to costs) assume a static environment. Examples are Geoffrion and Graves (1974), Benders et al. (1986), and Fleischmann (1993). Duran (1987) studies a dynamic model for the planning of production, bottling, and distribution of beer, but focuses on the production process, and Muriel and Simchi-Levi (2004) and Muriel (1997) study a dynamic, but uncapacitated, distribution problem in an operational setting.

Unfortunately, the adequacy of those models is limited to situations where, in particular, the demand pattern is stationary over time. In addition, inventory decisions cannot be supported using stationary models. In this paper we study the multi-period single-sourcing problem (MPSSP), a model that was introduced as a tool for evaluating logistics distribution network designs with respect to costs in a dynamic environment. We consider a logistics distribution network consisting of facilities and retailers. Production and storage take place at the facilities, and the retailers' demand patterns for a single product are assumed to be known. This model is suitable for tactical and strategic purposes, in particular when we assume that the planning period is a typical future one, and will repeat itself over time. This means that the model is cyclic in nature. Our model could also be modified to apply to shorter-term, operational or tactical, purposes by assuming a fixed starting and ending period with corresponding initial and target inventory levels, yielding an acyclic model. We assume that there is no transportation between the facilities. In addition, we do not allow for inventories at the retailers. This situation is typical in, for instance, the food and beverage industry, where the retailers often are supermarkets and restaurants, which usually have very limited storage capacity.

The decisions that need to be made are (i) the assignment of retailers to facilities, (ii) the timing of production, and (iii) the location and size of inventories. Previous research has focused on variants of the MPSSP in which each facility has finite, and possibly time-varying, production capacities (Romeijn and Romero Morales 2001, 2003, 2004). However, in contrast to earlier models, in this paper we consider situations in which several additional types of capacity limitations are present (Romero Morales 2000). To account for the fact that warehouse capacities are limited, we include physical inventory capacity constraints. Throughput capacity constraints are included to account for situations in which operational constraints limit the amount of products that can flow through a facility in a particular period. Finally, many goods are perishable, either due to a physically limited lifetime, or due to fashion considerations. To account for this, we allow for a constraint on the number of periods that a good is stored at a facility before being transported to the retailer.

Service considerations often dictate that retailers are delivered by a single facility, leading to a model with a natural assignment structure. This structure has the added modeling advantage that the transportation cost functions may be arbitrary functions of demand and distance, which significantly increases the applicability of the model. We assume that the production and inventory costs are linear, and do not exhibit speculative motives.

Since even the problem of determining whether there exists a feasible solution to the basic MPSSP with production capacities only is NP-complete (Martello and Toth 1990, Romero Morales et al. 1999), it is unlikely that efficient methods exist that can solve large problem instances to optimality. Therefore, we focus on heuristic approaches to this problem. In particular, we

- formulate the generalization of the MPSSP as an assignment problem with a nonlinear objective function;
- develop efficient algorithms for evaluating this objective function, which requires the solution of subproblems that are capacitated lot-sizing problems that may also be interesting in their own right;
- provide an explicit characterization of the feasible region of this problem;
- propose a greedy heuristic solution approach (see, e.g., Martello and Toth 1981, Romeijn and Romero Morales 2003, 2004 for related heuristics), as well as a very-large-scale-neighborhood (VLSN) search improvement heuristic (see, e.g., Ahuja et al. 2000, 2002);
- propose a stochastic model on the problem data that allows us to

—prove that our greedy heuristic is asymptotically feasible and optimal in a probabilistic sense when the number of retailers grows large for the case where the retailer demands exhibit a common seasonality pattern;

—develop a model for generating problem instances that are highly capacitated yet likely to be feasible, and therefore challenging, for our computational experiments.

The remainder of the paper is organized as follows. In §2 we formulate the multi-period single-sourcing problem for the cyclic case as a mixed-integer linear-programming problem, reformulate the problem as a pure assignment problem with a nonlinear objective function, and discuss algorithms for efficiently evaluating the objective function. In §3 we present a greedy heuristic and describe the VLSN search approach. The results of extensive computational experiments are presented in §4. Some concluding remarks and issues for future research are discussed in §5. Four appendices can be found in the Online Supplement to this paper on the journal's website: Appendix A contains the analysis of a stochastic model for problem instances and in Appendix B we provide an asymptotic average-case analysis of the greedy heuristic using this model. In Appendix C we discuss the extension of the results from this paper to the acyclic case. Finally, in Appendix D we provide the tabulated results of some of our tests.

2. The Multi-Period Single-Sourcing Problem

In this section, we first formulate the MPSSP in a traditional way, as a mixed-integer linear program. Next, we show that the MPSSP can alternatively be formulated as an assignment problem with an objective function that is nonlinear and relatively expensive to evaluate since it involves the solution of a sub-optimization problem. We end the section by proposing efficient algorithms for solving these subproblems and outlining our solution approach to the MPSSP.

2.1. Traditional Formulation

We consider a model to evaluate the performance of a logistics distribution network that integrates production, inventory, and transportation decisions as well as resource capacities. In particular, let m denote the number of facilities (which perform both a production and a storage task), n the number of retailers whose demands need to be satisfied, and T the length of a typical future planning cycle that is expected to repeat itself. Note that this choice generalizes often-used single-period models, while at the same time allows for the modeling of dynamic demand patterns and the dynamics of inventories. At the same time, our model does not suffer from the typical start and end-of-study effect of traditional dynamic models with fixed starting and ending periods. (In Appendix C in the Online Supplement we describe how our models and algorithms can be modified to apply to such settings as well.)

The facilities face finite production capacities b_{it} in each period ($i = 1, \ldots, m$; $t = 1, \ldots, T$). The demand of retailer j in period t for a single product is d_{jt} ($j = 1, \ldots, n$). The production costs are assumed linear, with the unit production costs at facility i in period t denoted by p_{it}. The single-sourcing aspect of the model requires that the demand of a retailer is satisfied by a single facility only. The total transportation costs for supplying retailer j by facility i throughout the planning horizon are given by $c_{ij} \equiv \sum_{t=1}^{T} c_{ijt}(d_{jt})$, where c_{ijt} is an *arbitrary* transportation cost function. The unit inventory holding costs at facility i in period t are given by h_{it}. All parameters are nonnegative by definition.

The decisions to be made are the assignment of retailers to facilities, as well as the production quantities and inventory levels at the facilities. To this end, let x_{ij} be 1 if retailer j is assigned to facility i and 0 otherwise. In addition, let y_{it} represent the production quantity at facility i in period t, and I_{it} the quantity in storage at facility i at the end of period t. Hereafter $x \in \mathbb{R}^{mn}$ will denote the vector with components x_{ij} and similarly for y, $I \in \mathbb{R}^{mT}$.

The basic MPSSP is:

$$\text{minimize} \quad \sum_{t=1}^{T}\sum_{i=1}^{m} p_{it} y_{it} + \sum_{i=1}^{m}\sum_{j=1}^{n} c_{ij} x_{ij} + \sum_{t=1}^{T}\sum_{i=1}^{m} h_{it} I_{it}$$

$$\text{subject to} \quad y_{it} \le b_{it} \quad i = 1, \ldots, m;\ t = 1, \ldots, T \tag{1}$$

$$\sum_{j=1}^{n} d_{jt} x_{ij} + I_{it} = y_{it} + I_{i,t-1} \quad i = 1, \ldots, m;\ t = 1, \ldots, T \tag{2}$$

$$\sum_{i=1}^{m} x_{ij} = 1 \quad j = 1, \ldots, n \tag{3}$$

$$I_{i0} = I_{iT} \quad i = 1, \ldots, m \tag{4}$$

$$x_{ij} \in \{0, 1\} \quad i = 1, \ldots, m;\ j = 1, \ldots, n \tag{5}$$

$$y_{it}, I_{it} \ge 0 \quad i = 1, \ldots, m;\ t = 1, \ldots, T.$$

The constraints (1) model the production-capacity constraints, and (2) model the inventory-balance constraints. Constraints (3) and (5) enforce the single-sourcing structure, i.e., each retailer is assigned to exactly one facility. Finally, constraints (4) model the fact that T represents a typical sequence of periods that will repeat itself. In other words, in this model the initial and ending inventory levels are not fixed, but determined by the model to ensure that the ending inventory level prepares the system for future demands. We will make the common assumption that there are no speculative motives in the production and inventory costs, i.e., $p_{i[t+1]} \le p_{it} + h_{it}$ for all i and t, where $[t] \equiv (t-1) \bmod T + 1$. In particular, this means that, capacities permitting, demand should always be satisfied by production in recent periods, avoiding holding inventories as much as possible.

We will allow for various additional types of capacity constraints that often play a role in practice.

(i) *Throughput constraints.* Operational limitations often constrain the quantity of goods that can be handled at a particular facility during a given time period. Assuming that the finite *throughput capacity* at facility i in period t is given by r_{it}, we can model the corresponding constraints as

$$\sum_{j=1}^{n} d_{jt} x_{ij} \le r_{it} \quad i = 1, \ldots, m;\ t = 1, \ldots, T. \tag{6}$$

(ii) *Physical inventory constraints.* A finite storage capacity of $\bar{I}_{it}$ at facility i in period t can easily be modeled by including the following constraints:

$$I_{it} \le \bar{I}_{it} \quad i = 1, \ldots, m;\ t = 1, \ldots, T. \tag{7}$$

(iii) *Perishability constraints.* To account for the perishable nature of goods, we may constrain the number of periods that a good is stored at a facility. In particular, denoting the maximum number of periods that a product can be stored by k, we obtain the following constraints:

$$I_{it} \le \sum_{\tau=t+1}^{t+k} \sum_{j=1}^{n} d_{j[\tau]} x_{ij} \quad i = 1, \ldots, m;\ t = 1, \ldots, T \tag{8}$$

where we assume that $k \le T - 1$ (if not, we should consider a multiple of the cycle length T for which this assumption is satisfied). Perishability constraints have been considered in inventory control, but they can hardly be found in the literature on integrated production and distribution planning. A notable exception is Myers (1997) who proposes a linear-programming model to determine the maximum demand that a company dealing with perishability issues can accommodate.

We will refer to a specific instance of the MPSSP (including any desired capacity constraints) as (P), and denote its linear-programming relaxation by (LP).

2.2. Assignment Formulation

For a given assignment vector x, the MPSSP decomposes into a set of m structured minimum-cost network-flow models, one for each facility. This yields the following reformulation of the MPSSP as an assignment problem (see also Romeijn and Romero Morales 2003, 2004):

$$\text{(A)}\quad \text{minimize} \quad \sum_{i=1}^{m}\sum_{j=1}^{n} c_{ij}x_{ij} + \sum_{i=1}^{m} H_i(x_{i\cdot})$$

$$\text{subject to} \quad \sum_{j=1}^{n} d_{jt}x_{ij} \le r_{it}$$

$$i = 1, \ldots, m;\ t = 1, \ldots, T \quad (9)$$

$$\sum_{i=1}^{m} x_{ij} = 1 \quad j = 1, \ldots, n$$

$$x_{ij} \in \{0, 1\} \quad i = 1, \ldots, m;\ j = 1, \ldots, n$$

where $H_i(z)$, for $z \in \mathbb{R}^n_+$, is the optimal solution value of the following subproblem:

$$(\text{H}_i)\quad \text{minimize} \quad \sum_{t=1}^{T} p_{it}y_t + \sum_{t=1}^{T} h_{it}I_t$$

$$\text{subject to} \quad y_t \le b_{it} \quad t = 1, \ldots, T$$

$$\sum_{j=1}^{n} d_{jt}z_j + I_t = y_t + I_{[t-1]}$$

$$t = 1, \ldots, T \quad (10)$$

$$I_t \le \min\left(\bar{I}_{it},\ \sum_{\tau=t+1}^{t+k}\sum_{j=1}^{n} d_{j[\tau]}z_j\right)$$

$$t = 1, \ldots, T \quad (11)$$

$$I_T = I_0$$

$$y_t, I_t \ge 0 \quad t = 1, \ldots, T.$$

Clearly, this subproblem may be infeasible for some $z \in \mathbb{R}^n_+$, which would yield $H_i(z) = \infty$. Alternatively, we may impose additional constraints on (A) to ensure that the corresponding subproblems are feasible for all feasible assignments x. In addition, it is of independent interest to characterize the feasible assignment vectors explicitly. In the remainder of this section, we derive such feasibility conditions for the MPSSP.

2.2.1. Subproblem Feasibility Conditions. The subproblems (H_i) are actually capacitated minimum-cost network-flow problems. The corresponding graphs contain a single supply node (denoted by 0), as well as T demand nodes (denoted by $1, \ldots, T$). Demand node t has demand equal to the total demand in period t of all retailers assigned to facility i, i.e., $\sum_{j=1}^{n} d_{jt}z_j$, and the supply node has supply equal to the sum of all demands: $\sum_{t=1}^{T}\sum_{j=1}^{n} d_{jt}z_j$. There are production arcs of the form $(0, t)$ for all $t = 1, \ldots, T$, with unit costs p_{it} and capacities b_{it}. In addition, there are inventory arcs of the form $(t, [t+1])$ for all $t = 1, \ldots, T$, with unit costs h_{it} and capacities $\min(\bar{I}_{it}, \sum_{\tau=t+1}^{t+k}\sum_{j=1}^{n} d_{j[\tau]}z_j)$. Figure 1 illustrates the graph corresponding to this problem, where the arc labels denote the capacities (using the notation of Theorem 2.1), and the arc costs are omitted. The following theorem uses this reformulation to characterize the set of vectors $z \in \mathbb{R}^n_+$ for which $H_i(z) < \infty$ (or, equivalently, (H_i) is feasible). In the remainder, we will refer to this set as the *domain* of H_i.

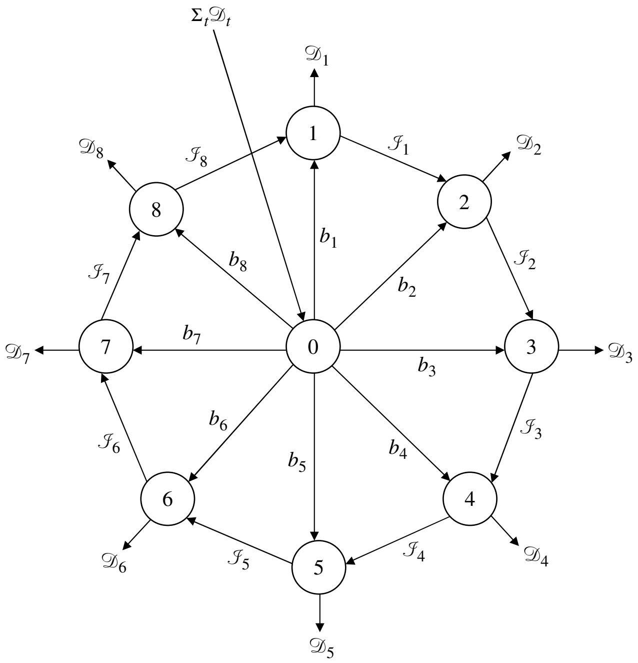

Figure 1 **Network Flow Formulation,** $T = 8$

Theorem 2.1. *The domain of the function H_i consists of all $z \in \mathbb{R}^n_+$ satisfying*

$$\sum_{t=1}^{T}\sum_{j=1}^{n} d_{jt}z_j \le \sum_{t=1}^{T} b_{it} \quad (12)$$

$$\sum_{t=\tau+k}^{\tau+r}\sum_{j=1}^{n} d_{j[t]}z_j \le \sum_{t=\tau}^{\tau+r} b_{i[t]}$$

$$\tau = 1, \ldots, T;\ r = k, \ldots, T-2 \quad (13)$$

$$\sum_{t=\tau}^{\tau+r}\sum_{j=1}^{n} d_{j[t]}z_j \le \sum_{t=\tau}^{\tau+r} b_{i[t]} + \bar{I}_{i[\tau-1]}$$

$$\tau = 1, \ldots, T;\ r = 0, \ldots, T-2. \quad (14)$$

Proof. For convenience, we will define

$$\mathcal{D}_t = \sum_{j=1}^{n} d_{jt}z_j \quad t = 1, \ldots, T$$

$$\mathcal{I}_t = \min\left(\bar{I}_{it},\ \sum_{\tau=t+1}^{t+k}\sum_{j=1}^{n} d_{j[\tau]}z_j\right) \quad t = 1, \ldots, T.$$

Theorem 6.12 in Ahuja et al. (1993) gives a general necessary and sufficient condition for feasibility of capacitated network-flow problems. For our problem, these conditions reduce to:

$$\sum_{t\in S}\mathcal{D}_t \le \sum_{t\in S} b_{it} + \sum_{t:\, t\notin S,\, [t+1]\in S} \mathcal{I}_t \qquad \text{for all } S\subseteq\{1,\ldots,T\} \quad (15)$$

$$-\sum_{t\in S}\mathcal{D}_t \le \sum_{t:\, t\in S,\, [t+1]\notin S} \mathcal{I}_t \quad \text{for all } S\subseteq\{1,\ldots,T\}. \quad (16)$$

It is clear that (16) is redundant. With respect to (15), note that any set S can be written as the union of disjoint sets, each of which contains consecutive integers (while observing the cyclic nature of the periods, i.e., period 1 follows T so that, for example, $\{T-1, T, 1, 2, 3\}$ is a set of consecutive integers). Constraints (15) corresponding to subset S are in fact the summation of the constraints corresponding to each of the sets in the union, and are thus implied by the latter constraints. Therefore, we can restrict ourselves to subsets S containing consecutive integers. This means that (H_i) is feasible if and only if

$$\sum_{t=1}^{T}\mathcal{D}_t \le \sum_{t=1}^{T} b_{it}$$

$$\sum_{t=\tau}^{\tau+r}\mathcal{D}_{[t]} \le \sum_{t=\tau}^{\tau+r} b_{i[t]} + \mathcal{I}_{[\tau-1]} \quad \tau=1,\ldots,T;\ r=0,\ldots,T-2.$$

Returning to the original notation, the first condition corresponds to (12). The second set of conditions yield

$$\sum_{t=\tau}^{\tau+r}\sum_{j=1}^{n} d_{j[t]}z_j \le \sum_{t=\tau}^{\tau+r} b_{i[t]} + \min\left(\bar{I}_{i[\tau-1]},\ \sum_{t=\tau}^{\tau+k-1}\sum_{j=1}^{n} d_{j[t]}z_j\right) \qquad \tau=1,\ldots,T;\ r=0,\ldots,T-2$$

which is equivalent to the pair of conditions

$$\sum_{t=\tau}^{\tau+r}\sum_{j=1}^{n} d_{j[t]}z_j \le \sum_{t=\tau}^{\tau+r} b_{i[t]} + \bar{I}_{i[\tau-1]} \qquad \tau=1,\ldots,T;\ r=0,\ldots,T-2 \quad (17)$$

$$\sum_{t=\tau}^{\tau+r}\sum_{j=1}^{n} d_{j[t]}z_j \le \sum_{t=\tau}^{\tau+r} b_{i[t]} + \sum_{t=\tau}^{\tau+k-1}\sum_{j=1}^{n} d_{j[t]}z_j \qquad \tau=1,\ldots,T;\ r=0,\ldots,T-2. \quad (18)$$

Now note that conditions (17) are precisely conditions (14). For $\tau=1,\ldots,T$ and $r=k,\ldots,T-2$ condition (18) can be rewritten as

$$\sum_{t=\tau+k}^{\tau+r}\sum_{j=1}^{n} d_{j[t]}z_j \le \sum_{t=\tau}^{\tau+r} b_{i[t]}$$

while for $r<k$ it can easily be seen to be redundant, which thus yields conditions (13). This completes the proof. □

2.2.2. Seasonal Demands. Since T denotes a typical sequence of demand periods that will repeat itself in the future, it will often correspond to the length of a typical demand cycle. It may then be reasonable to assume that retailers share a common seasonality pattern, i.e., d_j is the total demand of retailer j over T periods and $d_{jt}=\sigma_t d_j$ for a fixed vector of seasonality factors σ, such that $\sigma_t\ge 0$ for all $t=1,\ldots,T$, and $\sum_{t=1}^{T}\sigma_t=1$.

In this case, the throughput capacity constraints (9) reduce to

$$\sum_{j=1}^{n}\sigma_t d_j z_j \le r_{it} \quad i=1,\ldots,m;\ t=1,\ldots,T$$

or

$$\sum_{j=1}^{n} d_j z_j \le \frac{r_{it}}{\sigma_t} \quad i=1,\ldots,m;\ t=1,\ldots,T$$

where $z\in\mathbb{R}_+^n$ is the vector of assignment variables for a given facility i. Similarly, the constraints (12)–(14) reduce to

$$\sum_{j=1}^{n} d_j z_j \le \sum_{t=1}^{T} b_{it}$$

$$\sum_{j=1}^{n} d_j z_j \le \frac{\sum_{t=\tau}^{\tau+r} b_{i[t]}}{\sum_{t=\tau+k}^{\tau+r}\sigma_{[t]}} \quad \tau=1,\ldots,T;\ r=k,\ldots,T-2$$

$$\sum_{j=1}^{n} d_j z_j \le \frac{\sum_{t=\tau}^{\tau+r} b_{i[t]} + \bar{I}_{i[\tau-1]}}{\sum_{t=\tau}^{\tau+r}\sigma_{[t]}} \quad \tau=1,\ldots,T;\ r=0,\ldots,T-2.$$

We can thus summarize all capacity constraints by

$$\sum_{j=1}^{n} d_j z_j \le B_i \qquad i=1,\ldots,m \quad (19)$$

where

$$B_i=\min\left\{\min_{t=1,\ldots,T}\left(\frac{r_{it}}{\sigma_t}\right),\ \sum_{t=1}^{T} b_{it},\ \min_{\substack{\tau=1,\ldots,T\\ r=k,\ldots,T-2}}\left(\frac{\sum_{t=\tau}^{\tau+r} b_{i[t]}}{\sum_{t=\tau}^{\tau+r}\sigma_{[t]}}\right),\ \min_{\substack{\tau=1,\ldots,T\\ r=0,\ldots,T-2}}\left(\frac{\sum_{t=\tau}^{\tau+r} b_{i[t]} + \bar{I}_{i[\tau-1]}}{\sum_{t=\tau}^{\tau+r}\sigma_{[t]}}\right)\right\}.$$

This means that the feasible region of (A) is in fact the feasible region of a generalized assignment problem with agent-independent requirements (sometimes also called the single-sourcing problem, SSP). However, recall that in the MPSSP the objective function is nonlinear. In fact, the objective function is convex (Romero Morales 2000), implying that the optimal solution to the MPSSP will in general not be an extreme point of the feasible region of the SSP, so that results that are known about the SSP do not necessarily extend to the MPSSP with seasonal demands.

2.3. Evaluating the Objective Function

In this section we describe how the objective function of (A) for a given set of assignments, which requires solving the subproblems (H_i), can be computed efficiently. We will first develop an algorithm that solves the subproblem in $O(T^2)$ time, and next develop an $O(T)$ algorithm for the special case without physical inventory and perishability capacities. Note that these running times assume that all input data to the problem are given. If we include the time required to compute the input data, i.e., the aggregate demands of all retailers assigned to the corresponding facility and, if applicable, the inventory capacities implied by the perishability constraints, the running times become $O(T^2+nT)$ and $O(nT)$, respectively.

Note that we may apply these algorithms even on infeasible subproblem instances to determine whether the instance is feasible. It is interesting that the running time required for verifying feasibility using this approach is the same as using the feasibility conditions derived in §2.2.1. Nevertheless, these feasibility conditions may be of independent interest. In particular, they are essential in the asymptotic analysis of the heuristic that we will propose in §3.1, and they also enable us to construct a suitable data model for test-problem generation in §4.1.

2.3.1. General Approach. Note that, without loss of optimality we can assume that $\min_{t=1,\ldots,T} I_t = 0$. We propose to solve the problems (H_i) by, for each $t=1,\ldots,T$, fixing $I_t=0$ and treating period t as the "last" planning period. The cheapest one among the corresponding feasible solutions is then the optimal solution. If none of the T problems has a feasible solution, (H_i) is infeasible.

Given a period t for which $I_t=0$, we can use a backward recursion to obtain the best such solution as follows (where we will assume, without loss of generality, that this period is T). By the assumption that there are no speculative motives in the production and inventory costs, the optimal solution will avoid using the inventory arcs as much as possible. Therefore, we will produce the demand of each demand node as late as possible, thereby using as little inventory as possible. The optimal *candidate* inventories, for now disregarding the inventory capacity constraints, can then be found recursively as follows:

$$I_t = \begin{cases} 0 \quad \text{for } t=T \\ \max\left(0, \sum_{j=1}^{n} d_{j,t+1} z_j + I_{t+1} - b_{i,t+1}\right) \\ \quad \text{for } t=T-1,\ldots,0. \end{cases} \tag{20}$$

If all inventory levels satisfy the side constraints (including $I_0=0$), this solution is feasible and therefore optimal. On the other hand, if one or more inventory capacities are violated, the problem does not have a feasible solution. This follows directly from the fact that, by construction, the solution given above uses as little inventory as possible. (Note that the flows on the arcs of the form $(0,t)$ are simply equal to $\sum_{j=1}^{n} d_{jt}z_j + I_t - I_{t-1}$, which satisfy the capacity constraints by construction.) The running time of this procedure is $O(T)$, so that the total time required to solve T problems and find the best of the corresponding solutions is $O(T^2)$.

2.3.2. A Special Case. In the absence of physical inventory and perishability constraints, we can solve (H_i) more efficiently. For ease of exposition, we will always consider periods in a cyclic manner, i.e., $1,\ldots,T,1,\ldots$ etc. For example, $t=5,\ldots,2$ will mean that t takes on the values $5,6,\ldots,T,1,2$.

As in Theorem 2.1, define

$$\mathcal{D}_t = \sum_{j=1}^{n} d_{jt} z_j \quad t=1,\ldots,T.$$

Then define the partial sums of residual capacity, starting at the base period 1, as follows:

$$\Delta_t = \sum_{\tau=1}^{t} b_\tau - \sum_{\tau=1}^{t} \mathcal{D}_\tau \quad t=1,\ldots,T.$$

It is clear that the problem (H_i) is feasible if and only if $\Delta_T \geq 0$, i.e., total supply is no smaller than total demand. The following lemma finds a different base period, with respect to which all residual capacities are nonnegative.

Lemma 2.2. *Let $s \in \arg\min_{t=1,\ldots,T} \Delta_t$. Define*

$$\Delta'_t = \sum_{\tau=s+1}^{t} b_\tau - \sum_{\tau=s+1}^{t} \mathcal{D}_\tau \quad t=1,\ldots,T.$$

Then $\Delta'_t \geq 0$ for all $t=1,\ldots,T$.

Proof. It is easy to see that

$$\Delta'_t = \Delta_t + \Delta_T - \Delta_s \geq 0 \quad \text{for } t=1,\ldots,s$$
$$\Delta'_t = \Delta_t - \Delta_s \geq 0 \quad \text{for } t=s+1,\ldots,T. \quad \square$$

The following theorem now shows that, without loss of optimality, we can choose period s as defined in Lemma 2.2 to be the "last" planning period.

Theorem 2.3. *Let $s \in \arg\min_{t=1,\ldots,T} \Delta_t$. Then, without loss of optimality, we can assume that $I_s = 0$.*

Proof. We will show the result by contradiction. Let (y^*, I^*) be an optimal solution to the problem (H_i). Without loss of optimality, we may assume that there exists at least one t such that $I^*_t = 0$. Now suppose that $I^*_s = \varepsilon > 0$. Let t_1 be the last period *before* period s such that $I^*_{t_1} = 0$, and let t_2 be the first period *after* period s

such that $I^*_{t_2}=0$ (it may be that $t_1=t_2$). Since $I^*_{t_1}=0$ and $I^*_s=\varepsilon$, we know that

$$\sum_{t=t_1+1}^{s} y^*_t=\sum_{t=t_1+1}^{s}\mathcal{D}_t+\varepsilon$$

i.e., the total production in periods $t_1+1,\ldots,s$ exceeds the total demand in these periods by ε. Similarly, since $I^*_{t_2}=0$, we know that

$$\sum_{t=s+1}^{t_2} y^*_t=\sum_{t=s+1}^{t_2}\mathcal{D}_t-\varepsilon$$

i.e., the total production in periods $s+1,\ldots,t_2$ falls short of the total demand in these periods by ε.

Now note that, by Lemma 2.2,

$$\sum_{\tau=s+1}^{t} b_\tau\geq\sum_{\tau=s+1}^{t}\mathcal{D}_\tau\quad\text{for all } t=s+1,\ldots,t_2$$

i.e., the total production capacity in periods $s+1,\ldots,t$ is at least equal to the demand in these periods for all $t=s+1,\ldots,t_2$.

We can now conclude that we can find another feasible solution by decreasing the aggregate production in the periods $t_1+1,\ldots,s$ by ε, decreasing I^*_s to zero, and increasing the aggregate production in periods $s+1,\ldots,t_2$ by ε. Since the production and inventory costs exhibit no speculative motives, the cost of this modified solution is no worse than the solution we started with. Thus, we can assume without loss of optimality that $I_s=0$. □

Since the value of s in Theorem 2.3 can be computed in $O(T)$ time, and since the optimal solution to (H$_i$) given that $I_s=0$ can be computed in $O(T)$ time, (H$_i$) can be solved in $O(T)$ time.

3. A Heuristic Solution Approach

In this section we propose a heuristic approach for solving the MPSSP with all capacity constraints discussed in §2.1. The approach consists of two stages. In the first stage, we use a greedy heuristic to find an initial (partial) solution to the problem. In order to improve the quality of the solution found by the greedy heuristic (i.e., reduce the level of infeasibility if needed and reducing the costs of an initial feasible solution if one is found), we use a recently developed technique called VLSN search.

3.1. Greedy Heuristic

Martello and Toth (1981) proposed a greedy heuristic for the generalized assignment problem. This greedy heuristic was improved by Romeijn and Romero Morales (2000), who also showed that their improvement of the heuristic is asymptotically feasible and optimal with probability one under a very general stochastic model for the problem instances. Romeijn and Romero Morales (2003, 2004) subsequently generalized this heuristic and the analysis to the basic MPSSP as well as some extensions thereof. In this paper we further generalize the greedy heuristic to the MPSSP with the additional capacity constraints described above. In addition, we provide an asymptotic performance guarantee for particular stochastic models for the problem data (see Appendix B in the Online Supplement). Using the assignment formulation (A) of the MPSSP, the idea of the heuristic is to evaluate each possible assignment using some pseudo-cost function $f(i,j)$, which should measure the actual assignment costs, as well as the cost of using the limited capacities. For each assignment to be made, the difference between the second smallest and the smallest values of $f(i,j)$ (called the *desirability* of making the cheapest assignment with respect to the pseudo-cost) is computed, and assignments are made in decreasing order of this difference. Along the way the values of the desirabilities are updated to take into account the fact that the reduction in capacities caused by earlier assignments makes certain other assignments infeasible.

Greedy Heuristic

Step 0. Set $L=\{1,\ldots,n\}$ and $x^{\mathrm{G}}=0$.

Step 1. For all $j\in L$, let

$$\mathcal{F}_j=\{i\colon \text{retailer } j \text{ can feasibly be assigned to facility } i \text{ given } x^{\mathrm{G}}\}.$$

If $\mathcal{F}_j=\varnothing$ for some $j\in L$: let $L=L\backslash\{j\}$ and repeat Step 1. Otherwise, let

$$i_j\in\arg\min_{i\in\mathcal{F}_j} f(i,j)\quad\text{for } j\in L$$

$$\rho_j=\min_{\substack{s\in\mathcal{F}_j\\ s\neq i_j}} f(s,j)-f(i_j,j)\quad\text{for } j\in L.$$

Step 2. Let $\hat{\jmath}\in\arg\max_{j\in L}\rho_j$, and set

$$x^{\mathrm{G}}_{i_{\hat{\jmath}}\hat{\jmath}}=1$$

$$L=L\backslash\{\hat{\jmath}\}.$$

Step 3. If $L=\varnothing$: STOP, x^{G} is a (partial) solution to (A). Otherwise, go to Step 1.

The output of this greedy heuristic is a vector of assignments x^{G}, which is either a full or a partial solution of the reformulated problem (A).

For the MPSSP with capacity and perishability constraints we propose to use a generalization of a pseudo-cost function that was earlier developed for the basic version of the MPSSP (Romeijn and Romero Morales 2001, 2003, 2004). This pseudo-cost function is based on a study of the relationship between a basic optimal solution to (LP) and a corresponding dual optimal solution. The intuition behind it is that a measure of the cost of assigning retailer j to facility i should not only include the actual assignment costs from c_{ij}, but also a measure that reflects the cost of using resources. In particular, our analysis yields the following pseudo-cost function:

$$f(i,j) = c_{ij} + \sum_{t=1}^{T} \left(\lambda^*_{it} + \xi^*_{it} - \sum_{l=1}^{k} \alpha^*_{i[t-l]} \right) d_{jt}$$

where λ^*, ξ^*, and α^* are, respectively, the vectors of optimal dual multipliers for the inventory-balance constraints, throughput-capacity constraints, and perishability constraints in (LP) (where we have rewritten all capacity constraints as "$\geq$" constraints, so that the multipliers are all nonnegative). These dual multipliers are multiplied by the coefficient d_{jt} that appears in the left-hand side of the constraints, reflecting that $f(i,j)$ measures the cost with respect to the assignment variable x_{ij}. It is interesting to note that our pseudo-cost function does not explicitly contain the dual multipliers according to the production-capacity and physical-inventory constraints ((1) and (7)). However, note that the values of these dual multipliers are still implicitly part of the pseudo-cost through the dual constraints, which specify a relationship between *all* dual multipliers.

In Appendix B in the Online Supplement we provide a more formal derivation of the pseudo-cost function given above. In addition, we perform an average-case analysis of the greedy heuristic with this pseudo-cost function. In particular, using a stochastic model that is developed in Appendix A in the Online Supplement, we show that if the retailer demands follow a common seasonality pattern our heuristic yields, with probability one as the number of retailers increases to infinity, a feasible and optimal solution to the problem.

Despite this asymptotic-optimality result for the problem class with seasonal demand patterns, the greedy heuristic may yield suboptimal solutions—particularly for instances with general demand patterns and for instances where the number of retailers is relatively small. In fact, the heuristic may even fail to find a feasible solution. In order to improve the quality of the solution found by the greedy heuristic (i.e., reduce the level of infeasibility if needed and reducing the costs of an initial feasible solution if one is found), we use a recently developed technique called VLSN search, which will be described in the next section.

3.2. Very-Large-Scale-Neighborhood Search

The asymptotic performance guarantee enjoyed by the greedy heuristic discussed above suggests that this heuristic will provide good solutions to the MPSSP, particularly when the number of retailers is relatively large and when the retailer demands follow a common seasonality pattern. However, for fewer retailers, and for other demand patterns, we may need to enhance the heuristic in order to find good, or even feasible, solutions to the problem. A common approach is to use a local search procedure to enhance the initial solution found by the greedy heuristic. For example, one may consider a so-called *2-exchange* neighborhood, containing all solutions that can be obtained by interchanging the assignments of exactly two retailers that are currently assigned to different facilities. In particular, if retailer j_1 is assigned to facility i_1, and retailer j_2 is assigned to facility i_2, we may check if assigning retailer j_2 to facility i_1 and retailer j_1 to facility i_2 improves the quality of the solution. If no improvement is found for any of the possible exchanges of this type, the current solution is locally optimal.

It is well-known that as larger neighborhoods are considered in a local search procedure, it may be expected that higher-quality solutions will be obtained. On the other hand, the computational effort required to find a locally optimal solution may increase dramatically with the size of the neighborhood considered. The class of neighborhood-search techniques called VLSN search is aimed at using very large neighborhoods, while keeping the computational time required to find a local optimum limited. VLSN search has been successfully applied to many partitioning problems such as vehicle routing problems (Fahrion and Wrede 1990, Thompson and Psaraftis 1993, Gendreau et al. 2006), minimum-makespan machine scheduling (Frangioni et al. 2000) and other scheduling problems (Thompson and Psaraftis 1993), and the capacitated minimum spanning-tree problem (Ahuja et al. 2001b, 2003). Surveys of VLSN search can be found in Ahuja et al. (2000, 2002). Since the MPSSP can be viewed as a partitioning problem, VLSN search seems ideally suited for our problem.

The basic idea of VLSN search is to consider exchanges that involve more than two retailers and facilities, which we call *cyclic* and *path exchanges*. Two assignments x and x' are called *cyclic neighbors* if x' can be obtained from x by performing a cyclic exchange, e.g., by transferring a chain of retailers to different facilities. For instance, if x assigns retailers j_k, $k = 1, \ldots, \widehat{m}$ to distinct facilities i_k, $k = 1, \ldots, \widehat{m}$ (where $\widehat{m} \leq m$), and x' assigns retailer j_1 to facility i_2, retailer j_2 to facility $i_3, \ldots,$ retailer $j_{\widehat{m}-1}$ to facility $i_{\widehat{m}}$, and finally retailer $j_{\widehat{m}}$ to facility i_1, x and x' are cyclic neighbors. Similarly, x and x'' are called *path neighbors* if x'' is obtained from x by performing a path exchange, e.g., by transferring retailer j_1 to facility i_2, retailer j_2 to facility $i_3, \ldots,$ retailer $j_{\widehat{m}-1}$ to facility $i_{\widehat{m}}$ (see Figure 2 for examples).

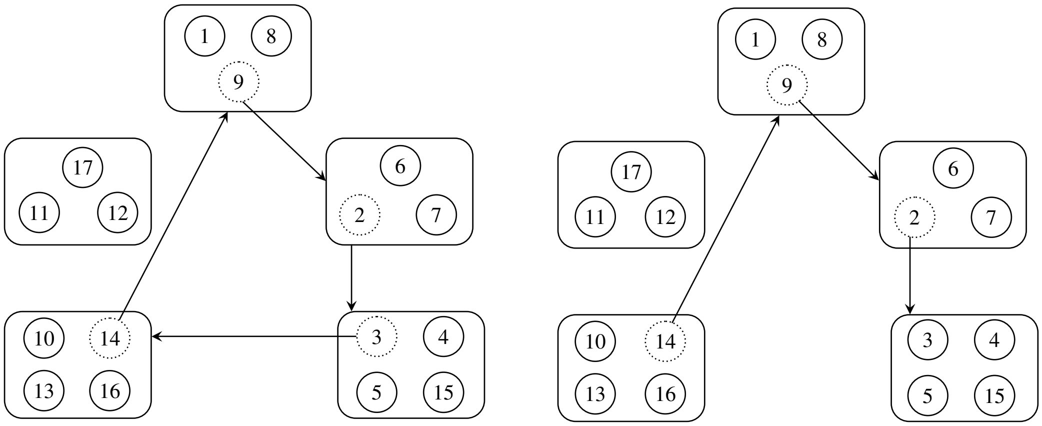

Figure 2 Cyclic and Path Exchange

3.2.1. The Improvement Graph. Thompson (1998), Thompson and Orlin (1989), Thompson and Psaraftis (1993), and Ahuja et al. (2001b) show how to find an improving neighbor in the cyclic exchange neighborhood by finding a *subset-disjoint cycle* in the so-called *improvement graph*. The improvement graph can be constructed as follows. Given a solution x to the MPSSP, for each retailer $j = 1, \ldots, n$, let $i(j)$ denote the facility to which retailer j is assigned. The improvement graph is a directed graph $G = (V, A)$ with the node set $V = \{1, \ldots, n\}$. The arc set A contains all pairs (j_1, j_2) such that $j_1, j_2 \in V$ with the property that transferring retailer j_1 from $i(j_1)$ to $i(j_2)$ and removing retailer j_2 from $i(j_2)$ yields a feasible assignment *for facility* $i(j_2)$. The arc costs of this arc are defined as the change in costs for facility $i(j_2)$ due to this transfer, i.e.

$$A(j_1, j_2) \\ = c_{i(j_2)j_1} - c_{i(j_2)j_2} + H_{i(j_2)}\big(x_{i(j_2)\cdot} + e_{j_1} - e_{j_2}\big) - H_{i(j_2)}\big(x_{i(j_2)\cdot}\big)$$

where $e_j \in \mathbb{R}^n$ denotes the jth unit vector. A cycle in G is called *subset-disjoint* if all nodes in the cycle correspond to retailers that are assigned to distinct facilities in the solution x. Thompson and Orlin (1989) and Thompson and Psaraftis (1993) show that there is a one-to-one correspondence between cyclic exchanges for a partitioning problem like the MPSSP and subset-disjoint cycles in the improvement graph. This means that a subset-disjoint cycle *with negative cost* yields an improved solution, where the cost of the cycle gives the change in cost. However, the problem of determining whether there exists a subset-disjoint cycle in the improvement graph is NP-complete, and the problem of finding a negative cost subset-disjoint cycle is NP-hard (Thompson 1988, Thompson and Orlin 1989, and Thompson and Psaraftis 1993). Heuristics that may be effective in practice have been developed by Ahuja et al. (2001a, b). In this paper, however, we chose to employ the exact dynamic-programming (or implicit-enumeration) algorithm that was developed and discussed in detail in Ahuja et al. (2003). In particular, this algorithm may be used to find the optimal negative-cost subset-disjoint cycle or, at a significant reduction in running time, the best cycle of a given maximum length. We have chosen to use the former when the number of facilities in the MPSSP is relatively small, and switch to the latter when the number of facilities is larger (see also §4).

To allow for path exchanges also, Ahuja et al. (2001b) show that a slight modification of the improvement graph will provide a one-to-one correspondence between subset-disjoint cycles in the modified improvement graph and cyclic or path exchanges, allowing for both types of exchanges using a single improvement graph.

3.2.2. Subproblems. Besides the problem of finding a negative-cost subset-disjoint cycle in the improvement graph, the other important issue is how to generate and update the improvement graph itself efficiently. Note that determining whether a candidate arc (j_1, j_2) is indeed present in the graph, and if so, what its cost is, involves solving an optimization subproblem of the type (H_i). In this section we will discuss the computational burden of solving these subproblems.

In addition to the running time of the procedure for solving the subproblems itself, we also need to compute the input data for the subproblems, i.e., the demands, which takes $O(nT)$ time when computed from scratch. However, after finding an improved neighbor, updating all demands for facility i takes $O(T)$ time, since at most one retailer leaves the facility, and at most one retailer enters the facility.

Since the value of s in Theorem 2.3 can be computed in $O(T)$ time, and since the optimal solution to (H_i) given that $I_s = 0$ can be computed in $O(T)$ time, (H_i) can be solved in $O(T)$ time. Note that, in the case with production capacities only, we may check feasibility of the assignments $x_{i\cdot}$ for facility i in $O(1)$ time by observing that only the constraint (12) is relevant. In the presence of other capacities, checking feasibility directly is computationally more expensive than applying the $O(T)$ method outlined above.

3.2.3. Initial Feasible Solution. Since VLSN search is an improvement heuristic, we need an initial feasible solution. If the greedy heuristic introduced in §3.1 yields a feasible solution, we will use that solution. If not, the greedy heuristic does yield a partial solution. We then complete this solution by assigning all retailers that are not assigned in the partial solution to their most desirable facility, disregarding the capacity constraints. This clearly yields an infeasible solution to the MPSSP. We then use a first-phase VLSN search (reminiscent of the phase I simplex method in linear programming) to convert this infeasible solution into a feasible solution. In particular, we will disregard the capacity constraints present in (A), including the constraints characterizing the domain of H_i. We can then measure the infeasibility of the problem instance (H_i) as follows, where $z \in \{0,1\}^n$ denotes the vector of assignments to facility i:

$$\Phi_i(z) = \sum_{t=1}^{T} \max\left\{0, \sum_{j=1}^{n} d_{jt} z_j - r_{it}\right\} + \min_{\tau=1,\ldots,T} \sum_{t=0}^{T} \max\left\{0, I_{it}^{\tau} - \min\left(\bar{I}_{it}, \sum_{\tau=t+1}^{t+k} \sum_{j=1}^{n} d_{j[\tau]} z_j\right)\right\}$$

where the values of I_{it}^{τ} are computed using the recursion (20), treating τ as the last period by setting $I_{i\tau}^{\tau} = 0$ and performing the backwards recursion in a cyclic manner from there. Using this measure of infeasibility, we then add the term

$$\nu \sum_{i=1}^{m} \Phi_i(x_{i\cdot})$$

to the objective function of (A), where ν is a positive penalty coefficient.

4. Computational Results

4.1. Generation of Problem Instances

Following the stochastic model described and analyzed in Appendix A in the Online Supplement, we tested our solution approach on problem instances generated as follows (where random variables are denoted by capital letters). For each problem instance, we generated a set of facilities and a set of retailers uniformly in the square $[0,10] \times [0,10]$. For the case where all retailers exhibit the same seasonality pattern (see §2.2.2), we generated an aggregate demand D_j from the uniform distribution on $[5,25]$ for each retailer j, and set $D_{jt} = \sigma_t D_j$ $(t = 1,\ldots,T)$. For the more general case, we generated a demand D_{jt} from the uniform distribution on $[5\sigma_t, 25\sigma_t]$ for each retailer in each time period. We fixed the number of time periods to $T = 6$, and, in most cases, chose the vector of seasonal factors to be $\sigma = (\frac{1}{9}, \frac{3}{18}, \frac{2}{9}, \frac{2}{9}, \frac{3}{18}, \frac{1}{9})^{\top}$. However, in this case perishability constraints, even with $k = 1$ or 2, often turn out to be non-binding. To obtain more illustrative results, we used a more extreme (albeit less realistic from a practical point of view) vector of seasonal coefficients when perishability constraints are present; in particular: $\sigma_1 = 6T^2/(T(T+1)(2T+1))$ and $\sigma_t = 6(t-1)^2/(T(T+1)\cdot(2T+1))$ $(t = 2,\ldots,T)$.

The costs C_{ij} are assumed to be proportional to demand and distance, i.e., $C_{ij} = \mathrm{dist}_{ij} \sum_{t=1}^{T} D_{jt}$, where dist_{ij} denotes the Euclidean distance between facility i and retailer j. We generated inventory holding costs h_{it} uniformly in the interval $[10,30]$ and, without loss of generality due to the absence of speculative motives, chose the production costs equal to zero.

Finally, we assumed that the capacities are equal for all facilities and all periods. To allow for sufficient capacity as the number of retailers grows, we let all capacities depend linearly on n, i.e., $b_{it} = \beta n$, $r_{it} = \rho n$, and $\bar{I}_{it} = \eta n$. Finally, we let

$$\beta = \delta \cdot \frac{\mathscr{E}(D_1)}{m} \cdot \max_{\substack{\tau=1,\ldots,T\\ r=k,\ldots,T-1}} \left(\frac{1}{r+1} \sum_{t=\tau}^{\tau+r} \sigma_{[t]}\right)$$

$$\rho = \delta' \cdot \frac{\mathscr{E}(D_1)}{m} \cdot \max_{t=1,\ldots,T} \sigma_t$$

$$\eta = \delta'' \cdot \max_{r=0,\ldots,T-2} \left(\frac{\mathscr{E}(D_1)}{m} \cdot \max_{\tau=1,\ldots,T} \sum_{t=\tau}^{\tau+r} \sigma_{[t]} - \beta(r+1)\right).$$

Using the analysis in Appendix A in the Online Supplement it can be shown that, if all retailer demands exhibit the same seasonality pattern, generated instances are feasible with probability one as the number of retailers approaches infinity if all and only if $\delta, \delta', \delta'' > 1$. In the absence of throughput, physical inventory, and perishability constraints we obtain

$$\beta = \delta \cdot \frac{\mathscr{E}(D_1)}{mT}.$$

Instances generated for this model are then asymptotically feasible with probability one if and only if $\delta > 1$, even in the case of general retailer demand patterns.

4.2. Results

We mainly considered cases with $m = 5$ facilities, and $n = 15$, 25, 50, 100, 150, 200, 250, and 300 retailers. For the production capacities, we mainly used a multiplier of $\delta = 1.1$. Recall that $\delta > 1$ ensures asymptotic feasibility; this value of δ thus provides relatively tight (and therefore relatively hard) problem instances, while still allowing for a reasonable fraction of feasible instances for small numbers of retailers. For

Table 1 Basic Case, Seasonal Demands ($m = 5$)

	Greedy heuristic			Including VLSN search				CPLEX
n	Feas.	Time (sec.)	Error (%)	Feas.	Time (sec.)	Error (%)	Iter.	Time (sec.)
15	19	0.001	10.61	22	0.08	1.69	2.1	3.99
25	16	0.006	16.49	21	0.16	0.95	3.0	30.93
50	21	0.012	8.73	24	0.41	0.89	3.5	12.85
100	25	0.046	3.06	25	1.79	0.37	4.0	26.00
150	25	0.098	2.27	25	5.55	0.21	4.7	115.95
200	25	0.171	1.63	25	10.78	0.15	4.3	$329.62^{(5)}$
250	25	0.268	1.45	25	22.80	0.12	5.1	$280.56^{(4)}$
300	25	0.389	1.17	25	34.50	0.08	4.7	$325.43^{(5)}$

Table 2 Throughput Constraints, Seasonal Demands ($m = 5, \delta' = 1.3$)

	Greedy heuristic			Including VLSN search			
n	Feas.	Time (sec.)	Error (%)	Feas.	Time (sec.)	Error (%)	Iter.
15	13	0.005	12.35	$16^{(2)}$	0.10	1.70	2.4
25	21	0.001	8.74	24	0.15	1.34	2.9
50	22	0.010	6.17	22	0.42	0.89	3.7
100	25	0.048	3.19	25	1.72	0.32	3.9
150	25	0.101	2.27	25	5.32	0.20	4.6
200	25	0.177	1.50	25	12.48	0.12	5.2
250	25	0.275	1.27	25	20.89	0.10	4.9
300	25	0.394	1.08	25	39.23	0.09	5.7

the throughput and physical-inventory constraints, choosing the tightness parameter too small would yield instances (and solutions) that are impractical, since they would allow for too little variety in the types of feasible solutions. Therefore, for the throughput constraints, we used $\delta' = 1.3$, and for the physical inventory constraints we used $\delta'' = 1.1, 1.5$. Finally, we considered perishability constraints with $k = 2$.

We first attempt to solve all problem instances using the greedy heuristic, after which VLSN search with both cyclic and path exchanges is employed to improve that solution. In case the greedy heuristic does not find a feasible solution, we use the first-phase VLSN approach discussed in §3.2.3. We initially set $\nu = 1{,}000$, and multiply it by a factor of ten when the first-phase VLSN search does not find an improving or feasible solution. As soon as VLSN search does find a feasible solution we disregard the penalty term in the objective function and return to the original VLSN algorithm, thereby ensuring feasibility for the remainder of the algorithm. If no feasible solution is obtained with $\nu = 10^{11}$, we stop the procedure without a solution to the problem.

For each class of problem instances we generated and solved 25 instances. Tables 1–5 report the main test results (additional results can be found in Tables 6–8 in the Online Supplement). The tables show, for both the greedy heuristic and VLSN search, the number of instances for which the heuristic found a feasible solution, the computation time, and the error (averaged over all instances for which a feasible solution was found). For VLSN search the average number of iterations, i.e., the number of improving solutions, is given. VLSN search was able to find a feasible solution to almost all instances that were in fact feasible (which we have verified by attempting to solve these problem instances using the MIP solver of CPLEX; see ILOG 2001). A superscript on the number of instances for which VLSN search found a feasible solution indicates the number of instances that were feasible, but for which VLSN search was not successful. In most cases, the error shown is an upper bound on the actual error, computed using the optimal solution value of the LP-relaxation. In Table 1, as well as for $n = 15, 25$ in all other tables, the error was computed using the optimal solution, or the best lower bound obtained by CPLEX within 15 minutes of CPU time. Finally, in Table 1 the average solution times using CPLEX are shown. The upper bound on the solution time was 15 minutes, and a superscript indicates the number of instances for which this limit was reached. Note that in all cases where VLSN was able to find a feasible solution, we provided this solution to CPLEX to speed up its solution time.

All tests were performed on a PC with a 667 MHz Pentium III processor with 128 MB RAM. All LP-problems and MIP problems were solved using CPLEX 7.0.

4.2.1. Main Results.

Tables 1–5 show results using the parameter choices indicated above. In our experiments, VLSN search was able to find a feasible solution whenever the problem instance was feasible, so the corresponding column in the tables actually indicates the number of feasible instances (out of the 25 generated). Only in 4.3% of the instances with $n = 15$ and 0.3% of the instances with $n = 25$ was VLSN search unsuccessful in finding a feasible solution to the problem.

Most tables clearly illustrate the asymptotic feasibility and optimality of the greedy heuristic. In Table 3,

Table 3 Physical Inventory Capacities, Seasonal Demands ($m = 5, \delta'' = 1.1$)

	Greedy heuristic			Including VLSN search			
n	Feas.	Time (sec.)	Error (%)	Feas.	Time (sec.)	Error (%)	Iter.
15	12	0.006	8.51	$17^{(2)}$	0.11	2.41	2.5
25	9	0.009	12.62	11	0.19	2.79	3.0
50	15	0.033	3.99	18	0.48	1.27	3.5
100	13	0.176	1.70	16	1.76	0.48	3.8
150	20	0.498	1.43	20	4.53	0.21	4.3
200	22	1.056	0.70	23	10.07	0.17	4.5
250	19	1.929	0.70	20	18.13	0.11	4.3
300	16	3.205	0.50	20	31.30	0.08	4.3

Table 4 Physical Inventory Capacities, Seasonal Demands ($m = 5, \delta'' = 1.5$)

	Greedy heuristic			Including VLSN search			
n	Feas.	Time (sec.)	Error (%)	Feas.	Time (sec.)	Error (%)	Iter.
15	18	0.003	11.44	20(1)	0.09	2.85	2.3
25	17	0.008	9.77	19	0.14	1.51	2.7
50	21	0.034	5.88	22	0.42	0.89	3.6
100	24	0.178	3.40	24	1.84	0.33	4.2
150	25	0.497	2.17	25	5.37	0.22	4.8
200	25	1.060	1.59	25	12.51	0.12	5.3
250	25	1.929	1.23	25	21.00	0.09	5.1
300	25	3.190	1.06	25	35.73	0.09	5.3

we see significantly fewer feasible instances than for all other cases. Additional experiments showed that the asymptotic nature of the feasibility result is more prominent here, and only for $n \geq 2{,}000$ do we stop seeing any infeasible instances. However, the ratio n/m between the number of retailers and facilities often needs to be quite large to obtain acceptable solution quality. VLSN search, however, is able to find good solutions to the problem in little time even for these apparently harder instances where the ratio n/m is small. The greedy heuristic and VLSN search seem to be complementary, in the sense that the problem instances for which the time required to obtain a good solution using VLSN search gets large often allow for a good solution using the greedy heuristic, but in far less time. In comparison, the time used by CPLEX to solve the problems to optimality is very problem-dependent, and increases quite rapidly in the size of the problem. For $n = 15$, CPLEX takes, on average, 4 to 230 times as much time as VLSN search. For $n = 25$, this increases to 8 to 270. The time used by CPLEX for problem instances with $n = 15$ are anywhere from less than a second to about half a minute. This indicates that for the smallest problems, VLSN combined with CPLEX could be a viable option. Combining these observations, it seems that VLSN/CPLEX should be used for the smallest instances, up to $n = 15$, VLSN search for the larger instances.

Table 5 Perishability Constraints, Seasonal Demands ($m = 5, k = 2$)

	Greedy heuristic			Including VLSN search			
n	Feas.	Time (sec.)	Error (%)	Feas.	Time (sec.)	Error (%)	Iter.
15	18	0.003	8.94	22	0.11	0.66	2.6
25	19	0.008	3.72	21(1)	0.18	0.29	3.5
50	21	0.032	2.71	24	0.41	0.44	4.7
100	25	0.194	1.13	25	1.28	0.18	5.2
150	25	0.533	0.77	25	2.98	0.09	5.3
200	25	1.122	0.50	25	5.94	0.06	5.8
250	25	2.033	0.46	25	9.41	0.04	5.2
300	25	3.346	0.37	25	16.14	0.04	6.0

The tables show that the computation times and solution quality for VLSN search are remarkably insensitive to the types of capacity constraints present. This can perhaps be explained by the fact that the feasible region of the MPSSP under seasonal demands is that of a SSP, regardless of the types of capacity constraints included. We conclude that the difference in the structure of the objective function in the different types of problem instances does not have a significant effect on the performance of the heuristics. As for the computation time involved, the operation that currently takes the most time is finding the negative cycle. Generally, around 50% of the time of VLSN search is spent there. It is interesting to note that, while only a small number of iterations of VLSN search is performed in this time, a local optimum that exhibits a dramatic improvement in solution quality is obtained.

4.2.2. General Demands in Basic Case and with Physical Inventory Capacities. Tables 6–8 in the Online Supplement show results obtained using the general demand model, for the basic MPSSP as well as the case with physical inventory capacities. In the latter case the feasibility condition for general demands is an open issue, but in the former the feasibility condition for seasonal and general demands coincide. We therefore decided to test the model including inventory capacities as well. Apart from somewhat larger errors and a lower success rate for instances with a small ratio of the number of retailers to the number of facilities, the similarity of the results for both demand models shows that the results in §4.2.1 are quite insensitive to the particular choice of demand model. The difference for small n is likely to be caused by apparently tighter capacities in the case of general demands. These results lead us to conjecture that the asymptotic performance guarantee of the greedy heuristic likely extends to the general demand case.

4.2.3. More Facilities in Basic Case. To study the impact of the number of facilities m, we tested the performance of the algorithms for the basic case. To avoid memory problems in the negative-cycle-detection procedure of VLSN search, we limited the cycle length to three retailers. The results in Table 8 in the Online Supplement indicate that, with this setting, the main difference between $m = 5$ and $m = 10$ is the error experienced by VLSN search. Recall that the errors are with respect to the best bound found by CPLEX within at most 15 minutes for $n = 25$, and with respect to the LP bound for all other instances. For these instances, CPLEX was rarely able to find an optimal solution even within a 30-minute time limit, and the errors are with respect to a probably weak lower bound. We conclude that problems with a very small ratio of the number of retailers to the number of facilities are particularly hard.

5. Summary and Future Research

We have considered the problem of evaluating a logistics-network design in a dynamic environment in the presence of production, inventory, and throughput capacities and perishability constraints. We have thoroughly analyzed feasibility properties of various subproblems, leading to stochastic models that are suitable for generating random problem instances, as well as efficiently implementable heuristic approaches. In particular, we have proposed a heuristic approach that enjoys attractive theoretical properties, and shown that it is very successful at obtaining high-quality solutions to the problem in limited time. We are currently investigating extensions of the approach to nonlinear production-cost structures, as well as cases that allow for inventories at the retailer level. Other interesting directions for future research would be to extend the models to allow for shipments between facilities (see Romeijn and Romero Morales 2002 for some initial results on such models), as well as the generalization to multiple products.

Acknowledgments

This work was supported in part by the National Science Foundation under Grants No. DMI-0085682 and DMI-0355533.

References

Ahuja, R. K., N. Boland, I. Dumitrescu. 2001a. Exact and heuristic algorithms for the subset disjoint minimum cost cycle problem. Working paper, Department of Industrial and Systems Engineering, University of Florida, Gainesville, FL.

Ahuja, R. K., T. L. Magnanti, J. B. Orlin. 1993. *Network Flows: Theory, Algorithms, and Applications*. Prentice-Hall, Englewood Cliffs, NJ.

Ahuja, R. K., J. B. Orlin, D. Sharma. 2000. Very large scale neighborhood search. *Internat. Trans. Oper. Res.* **7** 301–317.

Ahuja, R. K., J. B. Orlin, D. Sharma. 2001b. New neighborhood search structures for the capacitated minimum spanning tree problem. *Math. Programming* **91** 71–97.

Ahuja, R. K., J. B. Orlin, D. Sharma. 2003. A composite neighborhood search algorithm for the capacitated minimum spanning tree problem. *Oper. Res. Lett.* **31** 185–194.

Ahuja, R. K., O. Ergun, J. B. Orlin, A. Punnen. 2002. A survey of very large scale neighborhood search techniques. *Discrete Appl. Math.* **123** 75–102.

Benders, J. F., J. A. Keulemans, J. A. E. E. van Nunen, G. Stolk. 1986. A decision support program for planning locations and allocations with the aid of linear programming. C. B. Tilanus, O. B. de Gaus, J. K. Lenstra, eds. *Quantitative Methods in Management: Case Studies of Failures and Successes*, Chapter 4. Wiley, Chichester, UK, 29–34.

Duran, F. 1987. A large mixed integer production and distribution program. *Eur. J. Oper. Res.* **28** 207–217.

Fahrion, R., M. Wrede. 1990. On a principle of chain exchange for vehicle routing problems (I-VRP). *J. Oper. Res. Soc.* **41** 821–827.

Fleischmann, B. 1993. Designing distribution systems with transport economies of scale. *Eur. J. Oper. Res.* **70** 31–42.

Frangioni, A., E. Necciari, M. G. Scutellà. 2000. A multi-exchange neighborhood for minimum makespan parallel machine scheduling problems. *J. Combin. Optim.* **8** 195–220.

Gendreau, M., F. Guertin, J.-Y. Potvin, R. Séguin. 2006. Neighborhood search heuristics for a dynamic vehicle dispatching problem with pick-ups and deliveries. *Trans. Res. C.* **14** 157–174.

Geoffrion, A. M., G. W. Graves. 1974. Multicommodity distribution system design by Benders decomposition. *Management Sci.* **20** 822–844.

ILOG. 2001. *ILOG CPLEX 7.5 Reference Manual*. ILOG S. A., Gentilly, France.

Martello, S., P. Toth. 1981. An algorithm for the generalized assignment problem. J. P. Brans, ed. *Operational Research*. North-Holland, Amsterdam, The Netherlands, 589–603.

Martello, S., P. Toth. 1990. *Knapsack Problems, Algorithms and Computer Implementations*. Wiley, New York.

Muriel, A. 1997. On the effectiveness of set-partitioning approaches for large-scale machine scheduling and supply-chain management problems. Ph.D. thesis, Northwestern University, Evanston, IL.

Muriel, A., D. Simchi-Levi. 2004. Supply chain design and planning— Applications of optimization techniques for strategic and tactical models. A. G. de Kok, S. C. Graves, eds. *Handbooks in Operations Research and Management Science*, Vol. 11. *Supply Chain Management*: *Design*, *Coordination and Operation*. Elsevier, Amsterdam, The Netherlands.

Myers, D. C. 1997. Meeting seasonal demand for products with limited shelf lives. *Naval Res. Logist.* **44** 473–483.

Romeijn, H. E., D. Romero Morales. 2000. A class of greedy algorithms for the generalized assignment problem. *Discrete Appl. Math.* **103** 209–235.

Romeijn, H. E., D. Romero Morales. 2001. A probabilistic analysis of the multi-period single-sourcing problem. *Discrete Appl. Math.* **112** 301–328.

Romeijn, H. E., D. Romero Morales. 2002. A greedy heuristic for a three-level multi-period single-sourcing problem. A. Klose, M. G. Speranza, L. N. Van Wassenhove, eds. *Quantitative Approaches to Distribution Logistics and Supply Chain Management*. Springer-Verlag, Berlin, Germany, 191–214.

Romeijn, H. E., D. Romero Morales. 2003. An asymptotically optimal greedy heuristic for the multi-period single-sourcing problem: The cyclic case. *Naval Res. Logist.* **50** 412–437.

Romeijn, H. E., D. Romero Morales. 2004. Asymptotic analysis of a greedy heuristic for the multi-period single-sourcing problem: The acyclic case. *J. Heuristics* **10** 5–35.

Romero Morales, D. 2000. Optimization problems in supply chain management. Ph.D. thesis, Rotterdam School of Management, Erasmus University Rotterdam, Rotterdam, The Netherlands.

Romero Morales, D., J. A. E. E. van Nunen, H. E. Romeijn. 1999. Logistics network design evaluation in a dynamic environment. M. G. Speranza, P. Stähly, eds. *New Trends in Distribution Logistics*. Springer-Verlag, Berlin, Germany, 113–135.

Thompson, P. M. 1988. Local search algorithms for vehicle routing and other combinatorial problems. Ph.D. thesis, Operations Research Center, Massachusetts Institute of Technology, Cambridge, MA.

Thompson, P. M., J. B. Orlin. 1989. The theory of cyclic transfers. Working Paper OR 200-89, Operations Research Center, Massachusetts Institute of Technology, Cambridge, MA.

Thompson, P. M., H. N. Psaraftis. 1993. Cyclic transfer algorithms for multi-vehicle routing and scheduling problems. *Oper. Res.* **41** 935–946.