2010

The Stochastic Multiperiod Location Transportation Problem

Walid Klibi
Universite de Montreal

Francis Lasalle
universite de montreal

Alain Martel
Universite de Montreal

Soumia Ichoua
Embry-Riddle Aeronautical University, ichouasoumia@msn.com

Transportation Science

Publication details, including instructions for authors and subscription information:
http://pubsonline.informs.org

The Stochastic Multiperiod Location Transportation Problem

Walid Klibi, Francis Lasalle, Alain Martel, Soumia Ichoua,

TRANSPORTATION SCIENCE
Vol. 44, No. 2, May 2010, pp. 221–237
ISSN 0041-1655 | EISSN 1526-5447 | 10 | 4402 | 0221

DOI 10.1287/trsc.1090.0307

The Stochastic Multiperiod Location Transportation Problem

Walid Klibi, Francis Lasalle, Alain Martel
Interuniversity Research Centre on Enterprise Networks, Logistics and Transportation (CIRRELT),
Université de Montréal, Montréal, Quebéc H3C 3J7, Canada, and
Faculté des Sciences de l'Administration, Département Opérations et Systèmes de Décision,
Université Laval, Québec City, Québec G1V 0A6, Canada
{walid.klibi@cirrelt.ca, francis.lasalle@cirrelt.ca, alain.martel@cirrelt.ca}

Soumia Ichoua
Interuniversity Research Centre on Enterprise Networks, Logistics and Transportation (CIRRELT),
Université de Montréal, Montréal, Quebéc H3C 3J7, Canada, and
Department of Computer Science and Engineering, Johnson C. Smith University, Charlotte, North Carolina 28216,
soumia.ichoua@cirrelt.ca

This paper studies a stochastic multiperiod location-transportation problem (SMLTP) characterized by multiple transportation options, multiple demand periods, and a stochastic demand. We consider the determination of the number and location of the depots required to satisfy customer demand as well as the mission of these depots in terms of the subset of customers they must supply. The problem is formulated as a stochastic program with recourse, and a hierarchical heuristic solution approach is proposed. It incorporates a tabu search procedure, an approximate route length formula, and a modified procedure of Clarke and Wright (Clarke, G., J. W. Wright. 1964. Scheduling of vehicles from a central depot to a number of delivery points. *Oper. Res.* **12** 568–581). Three neighbourhood exploration strategies are proposed and compared with extensive experiments based on realistic problems.

Key words: location problem; transportation problem; stochastic customer order process; stochastic programming; Monte Carlo scenarios; tabu search
History: Received: June 2008; revisions received: December 2008, September 2009; accepted: September 2009. Published online in *Articles in Advance* March 5, 2010.

Introduction

Depot location decisions arise at the strategic planning level of distribution networks. Fundamentally, in a business context, location-allocation problems involve the determination of the number and location of the depots required to satisfy customer demand as well as the mission of these depots in terms of the subset of customers they must supply. Deterministic, dynamic, and stochastic models have been proposed in the last decade to solve variants of the depot location problem. Comprehensive reviews of the literature on these models are found in Owen and Daskin (1998) and Klose and Drexl (2005). Three formulations of the problem under uncertainty were studied based, respectively, on stochastic programming (Birge and Louveaux 1997; Snyder and Daskin 2005), robust optimization (Kouvelis and Yu 1997), and queuing theory (Berman, Krass, and Xu 1995). In most of these formulations, the demand is the main random variable considered.

When the distribution network designed is in operation, on a daily basis, the depots must ship the products ordered by their customers to specified ship-to points. Moreover, nowadays, an increasing number of companies rely on external transportation resources to ship their products to customers and they do not have their own vehicle fleet. In this context, depending on the size of the orders received, they may be shipped in single-customer partial truckloads (STL) or full truckloads (FTL), on multidrop truckload routes (MTL), or via less-than-truckload (LTL) transportation. The profits of the distribution network depend heavily on the efficiency of the transportation decisions made. These transportation options are defined more precisely in the following pages. The vehicle routing problems (VRP) encountered for the MTL case were studied extensively in the literature (see Laporte and Osman 1995 for a review). Although exact solution methods were developed for deterministic routing problems (Toth and Vigo 1998), their complexity led most researchers in this field to propose heuristic solution methods (Laporte et al. 2000). A few stochastic versions of the VRP problem were also studied (Laporte and Louveaux 1998). Despite the fact that location and transportation problems are clearly interrelated, the large classical literature on these problems assumes that they are independent. In the last few years, however, major efforts have been devoted to the development of integrated models such as location-routing and inventory-routing models. A recent review of these integrated models is found in Shen (2007).

The first classification of location-routing problems (LRPs) is found in Laporte (1988), who proposes various deterministic formulations of the problem. More recent papers (Chien 1993; Tuzun and Burke 1999; Barreto et al. 2007; Prins et al. 2007) present heuristic methods to solve deterministic LRPs. A multiechelon version with inventory is addressed in Ambrosino and Scutella (2005). An uncapacitated LRP with distance constraints is studied in Berger, Coullard, and Daskin (2007); they propose a set-partitioning formulation and a branch-and-price solution approach. The hierarchical structure of the problem is stressed in Nagy and Salhi (1996a, b), who develop an heuristic in which a routing phase is embedded into the location method. A dynamic LRP defined over a planning horizon is examined in Laporte and Dejax (1989) and in Salhi and Nagy (1999). A stochastic LRP is studied in Laporte, Louveaux, and Mercure (1989) and in Albareda-Sambola, Fernandez, and Laporte (2007). In the stochastic program considered, depot locations and a priori routes must be specified in the first stage, and second-stage recourse decisions deal with first-stage route failures. Recently, Shen (2007) proposed a stochastic LRP model based on routing cost estimations. An approach to solve continuous LRPs is presented in Salhi and Nagy (2009). Comprehensive reviews of location-routing models and of their applications are provided in Min, Jayaraman, and Srivastava (1998) and Nagy and Salhi (2007).

The LRP models found in the literature have two main shortcomings when compared to the strategic needs of distribution businesses using external transportation resources.

First, most LRP models assume that the distributor has its own vehicle fleet and that the transportation problems encountered are purely VRPs. When common or contract carriers are used, more options are available, namely, single-customer full (FTL) or partial (STL) truckload, multidrop truckload (MTL), and less-than-truckload (LTL) transportation. The FTL, STL, and MTL options are provided by truckload (TL) carriers. The distinction comes from the way the truck is used by the shipper: FTL refers to the case where a full trailer is delivered to a single destination, STL to the case where the trailer shipped is not loaded to full capacity, and MTL to the case where the truck delivery route involves more than one destination. In the last case, the route is elaborated by the shipper as if it were its own truck. This leads to *location-transportation problems* (LTP) instead of location-routing problems. These problems must not be confused with the class of problems known as *transportation-location problems*, which are, in fact, transshipment-location problems as pointed out by Nagy and Salhi (2007).

Second, most LRP models implicitly assume that location decisions and routing decisions can be made simultaneously for the planning horizon considered (a year, for example), i.e., that the routes do not change on a daily basis and that customer demand is static. In the business context considered here, the location and mission of depots must be fixed for the planning horizon but transportation decisions are made on a daily basis in reaction to the customer orders received. This gives rise to what we call *stochastic multiperiod location-transportation problems* (SMLTP). Most LRP models do not consider the customer order arrival process explicitly. Salhi and Nagy (1999) introduce a deterministic multiperiod location-routing problem to analyse the consistency of the solutions provided by static location-routing methods. Laporte and Dejax (1989) consider a related dynamic location-routing problem but they do not require that the location and mission of the depots is fixed for the planning horizon. In our context, a random number of customers order a random amount of products on a daily basis, and these orders must be delivered the next day. The dynamic customer demand pattern is therefore not known when the problem has to be solved. We assume, however, that the demand process is stationary.

The aim of this paper is to provide a more precise definition of the SMLTP, to formulate it as a stochastic program with recourse, and to propose an heuristic method to solve it.

Laporte (1988) has shown that the LRP is NP-hard. For each time period, the transportation subproblem considered in the SMLTP is an open VRP having the same NP-hardness property as a basic VRP (Sariklis and Powell 2000). Thus, the SMLTP is an NP-hard stochastic combinatorial optimization problem. Consequently, although exact solution approaches may be used to solve small size instances of the SMLTP optimally, they are not capable of handling realistic instances. This justifies the development of heuristics that better achieve a good trade-off between computation time and solution quality.

The paper is organized as follows. The next section provides a detailed description of the SMLTP and formulates it as a stochastic program with recourse. The following section shows how this stochastic program can be solved with a sample average approximation (SAA) mixed-integer program (MIP) based on a Monte Carlo scenario sampling scheme. In the following section, a solution approach is proposed to solve the SMLTP: It combines efficiently, in a nested schema, a Monte Carlo scenario generation procedure, a transportation heuristic, and a location-allocation tabu search procedure. The section also proposes various neighbourhood exploration strategies. Finally, in the last section, experiments are designed to evaluate the various versions of the tabu heuristic proposed for problems with different realistic characteristics. Computational results are also presented and discussed.

Problem Description and Formulation

The Business Context

To start with, let us examine the business context of the SMLTP more closely. A company purchases (or manufactures) a family of similar products, considered as a single product, from a number of supply sources. This product is sold to customers located in a large geographical area and, hence, must be shipped to a large number of ship-to points. To provide good service, the company cannot satisfy customer orders directly from the supply sources and must operate a number of uncapacitated distribution centers (also referred to as depots). Customers order a varying quantity of product on a daily basis and the company wants to provide next-day delivery using common or contract carriers. For a given day t, at each depot, when all the orders are in, the company plans its transportation for the next day and requests from its carriers the trucks required to deliver products to ship-to points through FTL, STL, MTL, or LTL transportation. When the orders are delivered, the trucks need not return to the depot. Let L be the set of depots considered, P the set of ship-to points where orders can be delivered, $L_p \subset L$ the subset of depots that are able to provide next-day delivery service to ship-to point $p \in P$, and, conversely, $P_l \subset P$ the subset of ship-to points that could be served by depot l. Also, for day t, let K_{lt}^{TL} be the set of feasible TL (STL or MTL) delivery routes from depot $l \in L$, considering ship-to point demands, service requirements, and vehicle characteristics. Let $P_k \subset P$ be the ordered set of ship-to points in route $k \in K_{lt}^{\mathrm{TL}}$. The TL tariff w_k charged by the carriers to assign a specific truck to route k is based on the following formula:

$$w_k = \max(r_k m_k; \mathrm{TL}_l) + a_l(|P_k| - 1),$$

where

- m_k = total mileage of route k,
- r_k = transportation cost rate per mile for the vehicle associated to route k,
- TL_l = minimum transportation charge for any TL route starting at depot l,
- a_l = drop charge for any additional stop on a route starting at depot l.

Note that the charge w_k must be paid to the carrier independently of the load shipped in the vehicle. When a customer orders more than a truckload on a given day, we assume that the depot ships as much as possible in full truckloads. Note also that several routes $k \in K_{lt}^{\mathrm{TL}}$ could have identical ship-to point sets P_k if vehicle types with different capacities and rates can be used. However, if this occurs, there is no reason not to use the route with the lowest charge w_k. We therefore assume, in what follows, that the sets K_{lt}^{TL} contain only nondominated TL routes.

If the vehicle on route k has a load close to its capacity, then multidrop TL transportation is usually cheaper than LTL transportation. For the orders received on a given day, if it is not possible to construct routes with a good load or capacity ratio for all orders, then it may be cheaper to send some orders by LTL transportation. In particular, if a TL route is more expensive than LTL shipments, then it should not be used. More precisely, a route $k \in K_{lt}^{\mathrm{TL}}$ would not be used if $w_k > \sum_{p \in P_k} \mathrm{LTL}(l, p; d_{pt})$, where d_{pt} is the load to be dropped at ship-to point $p \in P_k$ on day t, and $\mathrm{LTL}(l, p; d)$ is the LTL charge for the shipment of a load d from depot l to ship-to point p. If its alternative LTL routes dominate such a TL route, it can be removed from the set of possible routes a priori. We assume from now on that all the routes in the set K_{lt}^{TL} are also nondominated by their alternative LTL routes. Conversely, on a day t, an LTL shipment on lane (l, p) would be considered only if it is cheaper than the least-cost STL shipment $k_{(l,p)}$ on this lane, i.e., only if $\mathrm{LTL}(l, p; d_{pt}) < \max(r_{k_{(l,p)}} m_{k_{(l,p)}}; \mathrm{TL}_l)$. This implies that in the daily transportation planning process at depot l, in addition to the nondominated feasible TL routes $k \in K_{lt}^{\mathrm{TL}}$, all the economic depot to ship-to point LTL routes $k \in K_{lt}^{\mathrm{LTL}}$ must be considered, i.e., the set of routes to consider is $K_{lt} = K_{lt}^{\mathrm{TL}} \cup K_{lt}^{\mathrm{LTL}}$. The LTL tariff paid for a single destination route $k \in K_{lt}^{\mathrm{LTL}}$ is given by $w_k = \mathrm{LTL}(l, p_k; d_{p_k t})$, where p_k is the ship-to point of route k.

Given the distribution network user context described previously, the strategic decisions to make here involve the selection of a subset of the depots $\underline{L}^* \subset L$ to operate during the planning horizon T considered as well as the assignment of ship-to points $\underline{P}_l^* \subset P_l$, $l \in \underline{L}^*$, to these depots in order to maximize total expected profits. Note that, in what follows, the notation $\underline{L}$ is used to represent a subset of open depots in L and $\bar{L} = L \backslash \underline{L}$ is its complement. Similarly, $\underline{P}_l \subset P_l$ is used to represent the subset of ship-to points assigned to depot $l \in \underline{L}$. An important aspect of the problem is that the mission of the selected depots, defined by their customer sets $\underline{P}_l^*$, $l \in \underline{L}^*$, must remain the same for each day $t \in T$ of the planning horizon. When a depot $l \in L$ is used, a fixed operating cost A_l is incurred and the unit value of

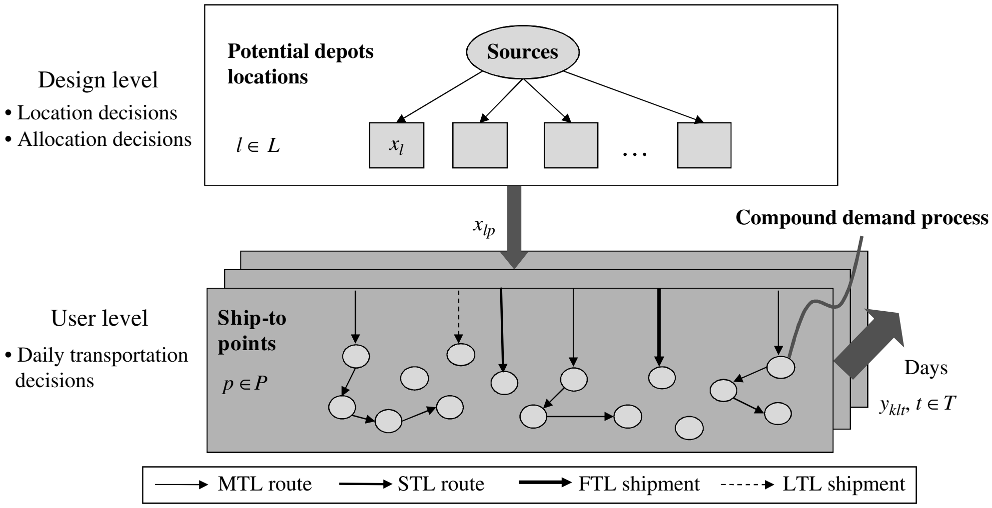

Figure 1 **Multiperiod Location Transportation Network**

products shipped from that depot is v_l. The value v_l takes into account the product production and procurement costs, inbound shipment costs, warehousing costs, and inventory holding costs. The unit price of products sold to ship-to point p is u_p. The structure of the multiperiod location-transportation network examined is illustrated in Figure 1, where the design and user problems are separated according to the temporal hierarchy between location and transportation decisions.

The Distribution Network User Problem

As previously explained, for a given distribution network with depots set $\underline{L} \subset L$ and mission $\underline{P}_l$, $l \in \underline{L}$, on a daily basis, the depots $l \in \underline{L}$ receive orders from their customers $p \in \underline{P}_l$ and they make shipping decisions for the next day. It is assumed that the demands of the ship-to points $p \in P$ follow a compound stationary stochastic process with a random order interarrival time q_p and a random order size o_p. The cumulative distribution functions of interarrival times and order sizes are denoted, respectively, by $F_p^q(\cdot)$ and $F_p^o(\cdot)$. A possible realization of these compound stochastic processes over planning horizon T is illustrated in Figure 2 for exponential interarrival times and normal order sizes. As can be seen, on a given day, only a subset of the ship-to points may require products. Such realizations constitute demand scenarios and the set of all demand scenarios associated with the compound demand processes considered is denoted by Ω. The probability that demand scenario $\omega \in \Omega$ will eventually be observed is denoted by $\pi(\omega)$. For a given scenario $\omega \in \Omega$, the set of ship-to points ordering products on day t is denoted by $P_t(\omega)$, and the shipments to make on day $t \in T$ at depot $l \in \underline{L}$ are defined by the loads $d_{pt}(\omega)$, $p \in \underline{P}_{lt}(\omega)$, where $\underline{P}_{lt}(\omega) = \underline{P}_l \cap P_t(\omega)$ is the set of depot l ship-to points that order products on day t.

Given the loads $d_{pt}(\omega)$, $p \in \underline{P}_{lt}(\omega)$, $l \in \underline{L}$, to deliver on day t, shipping decisions are made by the network depots *users* in two steps. First, for the loads that are larger than a truckload, a decision is made to ship as much as possible in full truckloads. Let $K_{plt}^{\text{FTL}}(\omega)$ be the set of vehicle types (routes) selected to make full truckload shipments to point p. Then, the residual loads to be inserted in the STL, MTL, or LTL shipments of depot l on day t are

$$\bar{d}_{pt}(\omega) = d_{pt}(\omega) - \sum_{k \in K_{plt}^{\text{FTL}}(\omega)} b_k y_{klt}^{\text{FTL}}(\omega), \quad p \in \underline{P}_{lt}(\omega), \quad (1)$$

where $y_{klt}^{\text{FTL}}(\omega)$ is the number of truckloads shipped to point p from depot l on route k, and b_k is the capacity of route k vehicles. We assume that $y_{klt}^{\text{FTL}}(\omega)$ is selected to ensure that $\bar{d}_{pt}(\omega)$ is nonnegative.

Next, the best delivery routes must be constructed. Let

$K_{lt}(\omega)$ = set of nondominated feasible delivery routes (i.e., such that $P_k \subset \underline{P}_{lt}(\omega)$ and $\sum_{p \in P_k} \bar{d}_{pt}(\omega) \le b_k$, $k \in K_{lt}(\omega)$) from depot l on day t under scenario ω;

δ_{kp} = binary coefficient taking the value one if ship-to point p is covered by route k (i.e., if $p \in P_k$), and zero otherwise;

$y_{klt}(\omega)$ = binary decision variable equal to one if route k is used from depot l on day t under scenario ω, and equals to zero otherwise.

For demand scenario ω, the best routes are obtained at depot l on day t by solving the following transportation subproblem:

$$C_{lt}^u(\omega) = \underset{\mathbf{y}}{\text{Min}} \sum_{k \in K_{lt}(\omega)} w_k y_{klt}(\omega) \quad (2)$$

subject to

$$\sum_{k \in K_{lt}(\omega)} \delta_{kp} y_{klt}(\omega) = 1 \quad p \in \underline{P}_{lt}(\omega), \quad (3)$$

$$y_{klt}(\omega) \in \{0, 1\} \quad k \in K_{lt}(\omega), \quad (4)$$

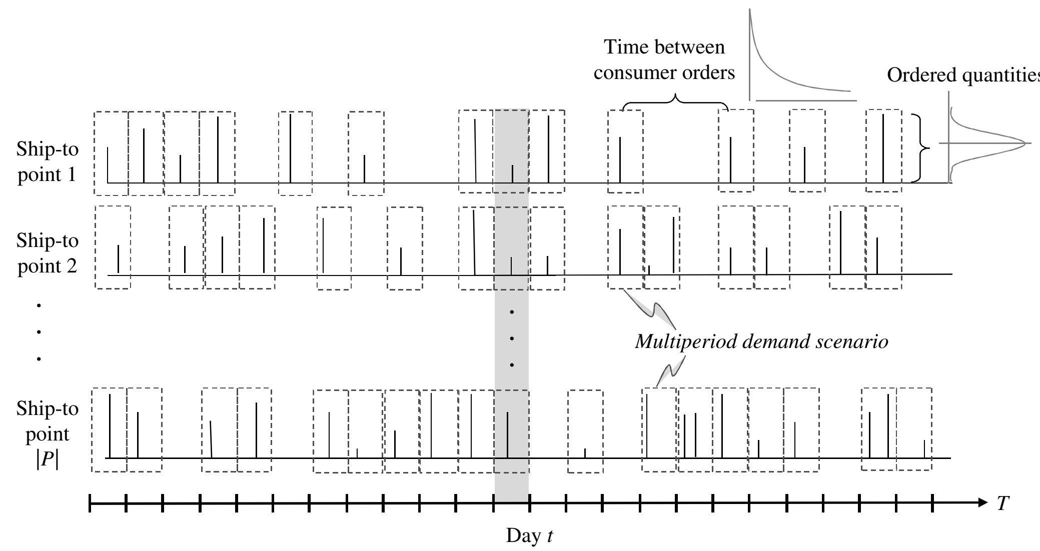

Figure 2 **Ship-to Points Stochastic Demand Process**

where $\mathbf{y}$ denotes the vector of all the routing decisions and $C^u_{lt}(\omega)$ is the cost of the optimal shipments made by depot l on day t under scenario ω. This model is similar to the classical set partitioning formulation of the deterministic VRP (Toth and Vigo 1998).

Furthermore, the shipments made on a daily basis generate sales revenues. Taking these into account as well as depot production and procurement, warehousing, inventory holding, and customer shipment costs, the net revenues $R^u(\omega)$ generated at the distribution network user level for demand scenario ω are given by

$$R^u(\omega) = \sum_{l\in \underline{L}}\sum_{t\in T}\Bigg[\sum_{p\in \underline{P}_{lt}(\omega)} [(u_p - v_l)d_{pt}(\omega)] - \sum_{p\in \underline{P}_{lt}(\omega)}\sum_{k\in K^{\mathrm{FTL}}_{plt}(\omega)} w_k y^{\mathrm{FTL}}_{klt}(\omega) - C^u_{lt}(\omega)\Bigg]. \quad (5)$$

Because u_p is the unit price of products sold to ship-to point p and v_l is the unit value of products shipped from depot l, $(u_p - v_l)$ provides the margin of products shipped from depot l to ship-to point p. These net revenues are important elements to take into account in the distribution network design model.

The Distribution Network Design Problem

The SMLTP is a hierarchical decision problem due to the temporal hierarchy between the location decisions and the transportation decisions. At the strategic level, the only decisions made *here and now* are the selection of the subset of facilities $\underline{L}^* \subset L$ to use during the planning horizon T considered and the mission $\underline{P}^*_l$ $l \in \underline{L}^*$ of these facilities. After a deployment lead time, on a daily basis, the transportation decisions discussed previously are made by the network users. However, the network design decisions must be considered when making daily transportation decisions and, conversely, adequate network design decisions cannot be made without anticipating the net revenues (5) generated by daily sales and transportation decisions for a given distribution network during the network usage horizon T. The best possible anticipation involves the explicit inclusion, in the design model, of the transportation model (2–4) and of the net revenue expression (5) but with the information available at the time the network design decisions are made. Because the ship-to point demands for the horizon T are not known when the design decisions are made, this information takes the form of the set of potential demand scenarios Ω previously defined. This leads to the formulation of the SMLTP as a two-stage stochastic program with recourse (Ruszczynski and Shapiro 2003), where the first stage deals with depot location and mission decisions and the second stage deals with daily transportation decisions. The following first-stage decision variables are required to formulate the model:

x_l = binary variable equal to one if depot l is opened, and to zero otherwise;

x_{lp} = binary variable equal to one if ship-to point p is assigned to depot l, and to zero otherwise.

The notation $\mathbf{x}$ is used to denote the vector of all these decision variables and $\mathbf{x}_l$ is used to denote the vector of depot l ship-to point assignment variables. Note that a given binary vector $\mathbf{x}$ specifies unique design sets $\underline{L}$ and $\underline{P}_l$, namely, $\underline{L} = \{l \mid x_l = 1\}$, $\bar{L} = \{l \mid x_l = 0\}$, and $\underline{P}_l = \{p \mid x_{lp} = 1\}$, $l \in \underline{L}$.

The stochastic programming model to solve is the following:

$$R^* = \operatorname*{Max}_{\mathbf{x}} R(\mathbf{x}) = \sum_{\omega\in\Omega} \pi(\omega) R^{du}(\mathbf{x}, \omega) - \sum_{l\in L} A_l x_l \quad (6)$$

subject to

$$\sum_{l\in L_p} x_{lp} = 1 \quad p \in P, \tag{7}$$

$$x_{lp} \le x_l \quad l \in L,\ p \in P_l, \tag{8}$$

$$x_l, x_{lp} \in \{0,1\} \quad l \in L,\ p \in P_l, \tag{9}$$

where, based on (2–5), the optimal value $R^{du}(\mathbf{x},\omega)$ of the second-stage program for design $\mathbf{x}$ and scenario ω is given by

$$R^{du}(\mathbf{x},\omega) = \sum_{l\in L}\sum_{t\in T}\Bigg\{\sum_{p\in P_t(\omega)}\Bigg[(u_p - v_l)d_{pt}(\omega) - \sum_{k\in K_{plt}^{\mathrm{FTL}}(\omega)} w_k y_{klt}^{\mathrm{FTL}}(\omega)\Bigg]x_{lp} - C_{lt}^{du}(\mathbf{x}_l,\omega)\Bigg\} \tag{10}$$

with

$$C_{lt}^{du}(\mathbf{x}_l,\omega) = \min_{\mathbf{y}} \sum_{k\in K_{lt}(\omega)} w_k y_{klt}(\omega) \tag{11}$$

subject to

$$\sum_{k\in K_{lt}(\omega)} \delta_{kp} y_{klt}(\omega) = x_{lp} \quad p \in P_t(\omega), \tag{12}$$

$$y_{klt}(\omega) \in \{0,1\} \quad k \in K_{lt}(\omega). \tag{13}$$

In the first term of objective function (6), expected net revenues are calculated, and in the second term, depot fixed costs are subtracted to get expected profits. Constraints (7) in the first-stage program enforce single depot assignments for ship-to points and constraints (8) limit ship-to point assignments to opened depots. Constraints (12) in the second-stage program are coupling relations, ensuring that daily route selections respect depot mission decisions.

Notwithstanding the inherent combinatorial complexity of this model, it would be virtually impossible to solve because the set of demand scenarios Ω is usually extremely large. In fact, when the interarrival times and order size distribution functions $F_p^q(\cdot)$ and $F_p^o(\cdot)$ are continuous, there is an infinite number of possible demand scenarios. This is the case, for example, when interarrival times are exponential and order sizes are log normal, a frequent case in practice. This difficulty can be alleviated, however, through the use of Monte Carlo scenario sampling methods.

Sample Average Approximation Model

The approach proposed to reduce the stochastic complexity of our problem is based on the Monte Carlo sampling methods presented in Shapiro (2003) and applied to the VRP in Verweij et al. (2003) and Rei, Gendreau, and Soriano (2007) and to supply chain network design problems in Santoso et al. (2005) and Vila, Martel, and Beauregard (2007). A random sample of scenarios is generated outside the optimization procedure and then a sample average approximation (SAA) program is constructed and solved. The idea is to first generate an independent sample of n equiprobable scenarios $\{\omega^1,\ldots,\omega^n\} = \Omega^n \subset \Omega$ from the initial probability distributions of order interarrival times and order sizes, which also removes the necessity of explicitly computing the scenario probabilities $\pi(\omega)$. Then, based on (6–13), the SAA program obtained is the following:

$$\bar{R}_n = \max_{\mathbf{x},\mathbf{y}} \frac{1}{n}\sum_{\omega\in\Omega^n}\sum_{l\in L}\sum_{t\in T}\Bigg\{\sum_{p\in P_t(\omega)}\Bigg[(u_p - v_l)d_{pt}(\omega) - \sum_{k\in K_{plt}^{\mathrm{FTL}}(\omega)} w_k y_{klt}^{\mathrm{FTL}}(\omega)\Bigg]x_{lp} - \sum_{k\in K_{lt}(\omega)} w_k y_{klt}(\omega)\Bigg\} - \sum_{l\in L} A_l x_l \tag{14}$$

subject to

$$\sum_{l\in L_p} x_{lp} = 1 \quad p \in P, \tag{15}$$

$$x_{lp} \le x_l \quad l \in L,\ p \in P_l, \tag{16}$$

$$\sum_{k\in K_{lt}(\omega)} \delta_{kp} y_{klt}(\omega) = x_{lp}, \quad \omega \in \Omega^n,\ l \in L,\ t \in T,\ p \in P_t(\omega), \tag{17}$$

$$x_l,\ x_{lp},\ y_{klt}(\omega) \in \{0,1\} \quad \omega \in \Omega^n,\ l \in L,\ p \in P,\ t \in T,\ k \in K_{lt}(\omega). \tag{18}$$

The scenarios in sample $\Omega^n \subset \Omega$ used in the model are generated directly from the cumulative distribution functions of interarrival times and order sizes $F_p^q(\cdot)$ and $F_p^o(\cdot)$, $p \in P$. Assuming that the customer orders are independent of each other, to sample a scenario $\omega \in \Omega$, we generate independent pseudorandom numbers u_q and u_o uniformly distributed on the interval $[0;1]$ and we compute the inverse, $F_p^{q-1}(u_q)$ and $F_p^{o-1}(u_o)$, of the distributions of interarrival times and order sizes. The **MonteCarlo** procedure used to generate the daily demands $d_{pt}(\omega)$, $p \in P$, $t \in T$ of the ship-to points for scenario ω is presented in Figure 3. In this procedure, the continuous variable τ is used to denote order-arrival times. Order arrivals are generated in the interval $]0,|T|]$ and mapped onto the corresponding planning periods $t \in T$. More than one order can arrive in a given planning period. Repeating this Monte Carlo sampling procedure n times yields the required sample of scenarios Ω^n. Note that all the **Procedures** presented in this paper use the following syntax:

Procedure(*input_variable1*, …; *procedure_parameter1*, …; *output_variable1*, …).

MonteCarlo$((F_p^q(\cdot), F_p^o(\cdot), p \in P), T; d_{pt}(\omega), p \in P, t \in T)$
For all $p \in P$, do:
$\tau = 0$; $d_{pt}(\omega) = 0$, $t \in T$
While $\tau \le |T|$, do:
Generate the Uniform $[0, 1]$ random numbers u_q and u_o
Compute the next order arrival time $\tau = \tau + F_p^{q-1}(u_q)$ and $t = \lceil \tau \rceil$
Compute the planning period t demand $d_{pt}(\omega) = d_{pt}(\omega) + F_p^{o-1}(u_o)$
End while
End do

Figure 3 Procedure Monte Carlo for the Generation of Scenario ω

Clearly, the quality of the solution obtained with this approach improves as the size n of the sample of scenarios used increases. The SAA model above has a structure similar to the deterministic location-routing problem (Berger, Coullard, and Daskin 2007), but it separates explicitly assignment and routing decisions and it is much larger. It uses an exponential number of binary decision variables for route selection. A preestablished set of nondominated routes can be generated as input to the model, but only small problem instances can be solved to optimality this way with commercial solvers. A better approach is to use column generation, which avoids the explicit consideration of all the possible routes. When $|T| * |\Omega^n|$ is large, however, this optimal approach can be used only for relatively small problems. Our aim in the next section is to propose an heuristic method to solve realistic problem instances.

Solution Approach

The General Scheme

The SMLTP is a hierarchical decision problem for which a hierarchical heuristic solution approach is a natural fit. Nagy and Salhi (2007) present a review of sequential, clustering, iterative, and hierarchical heuristic approaches to solve the LRP. The difference between these heuristics relates mainly to how the solution method treats the relationship between the location and the routing subproblems. The heuristic solution approach proposed in this section is a nested method that integrates location-allocation and transportation decisions in a hierarchical manner. In sync with the bilevel problem definition provided in Figure 1, the solution approach proposed builds on a *user level* transportation heuristic and a *design level* location-allocation heuristic combined into an efficient nested procedure. For the design problem, the tabu search heuristic proposed locates depots and assigns ship-to points to the opened depots. It is a local search procedure that explores neighbouring depot configurations and perturbs the ship-to point assignments using a set of restricted moves. For the user transportation problem, a modified and extended Clarke and Wright (1964) savings heuristic is proposed. This user heuristic is also used to evaluate potential moves in the design heuristic.

Clearly, several hundred moves may be considered during the solution process and an exact evaluation of each potential solution, using the user heuristic, would be too time-consuming. This is particularly true given the stochastic nature of our problem. To solve the SAA model (14–18), the evaluation of the solutions considered must be based on an estimation of their expected value for the scenario sample Ω^n generated. To reduce the calculation effort, we propose a fast approximate move evaluation procedure based on a route cost estimation formula. In the taxonomy proposed by Talbi (2002), our solution approach could be classified as a low-level hybrid heuristic. The following sections present our *user* level heuristic and our *design* level heuristic.

The User Problem Heuristic

Consider the user problem for a given distribution network with depots set $\underline{L} \subset L$ and mission $\underline{P}_l$, $l \in \underline{L}$. At the user level, under scenario ω, for all the ship-to point orders $\underline{P}_{lt}(\omega)$ received by depot l on day t, the objective of depot l is to select the FTL, STL, MTL, and LTL shipments and minimize its transportation costs. As explained previously, this is achieved with a two-step procedure. The first step determines the number of full truckload shipments to make and the residual ship-to point loads. The number of full trucks of each type $k \in K_{plt}^{\text{FTL}}(\omega)$ to ship on day t from depot l is found by solving the following simple integer programs by inspection:

$$[y_{klt}^{\text{FTL}}(\omega)]_{k \in K_{plt}^{\text{FTL}}(\omega)} = \arg\left(\max_{\mathbf{y}} \sum_{k \in K_{plt}^{\text{FTL}}(\omega)} b_k y_k \,\middle|\, \sum_{k \in K_{plt}^{\text{FTL}}(\omega)} b_k y_k \le d_{pt}(\omega),\; y_k = 0, 1, \ldots\right), \quad p \in \underline{P}_{lt}(\omega). \tag{19}$$

Then, the residual loads $\bar{d}_{pt}(\omega)$, $p \in \underline{P}_{lt}(\omega)$ to be shipped are computed with (1).

The second step solves program (2–4), i.e., it finds the best TL (STL or MTL) or LTL shipments to deliver the residual loads. To solve this transportation problem, we propose a modification of the simple and efficient VRP heuristic, based on perturbed Clarke and Wright (CW) (1964) savings and 2-opt improvements and developed by Girard, Renaud, and Boctor (2006). The main differences between our problem and a classical VRP are that (i) two different direct delivery transportation modes can be used (STL or LTL), and (ii) for all modes considered, the vehicle used does not return to the depot after its last drop. Clearly, the best direct delivery mode for a given ship-to point can

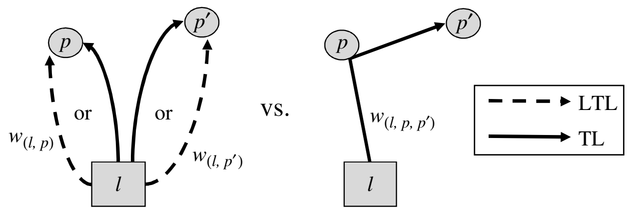

Figure 4 Alternative Routes Considered in the Savings Calculations for Depot *l*

be determined a priori by comparing their respective costs, so that the cost of the best direct shipment from depot l to ship-to point p is

$$w_{(l,p)} = \min[\mathrm{LTL}(l,p;\bar{d}_{pt}(\omega)); \max(r_{(l,p)}m_{(l,p)};\mathrm{TL}_l)], \quad p \in \underline{P}_{lt}(\omega). \quad (20)$$

Then, as illustrated in Figure 4, for two ship-to points p and p', the savings associated with using a multi-drop TL route (l,p,p') instead of the best direct shipments (l,p) and (l,p') can be calculated with the following expression:

$$e_{pp'} = w_{(l,p)} + w_{(l,p')} - w_{(l,p,p')}, \quad p \in \underline{P}_{lt}(\omega),\ p' \in \underline{P}_{lt},(\omega)\backslash\{p\}, \quad (21)$$

where $w_{(l,p,p')} = \max(r_{(l,p,p')}m_{(l,p,p')};\mathrm{TL}_l) + a_l$. The perturbed CW heuristic works as the original CW algorithm but uses the following modified savings formula:

$$e_{pp'} = w_{(l,p)} + w_{(l,p')} - \lambda_{pp'} w_{(l,p,p')}, \quad p \in \underline{P}_{lt}(\omega),\ p' \in \underline{P}_{lt}(\omega)\backslash\{p\}, \quad (22)$$

where the weight $\lambda_{pp'}$ is randomly selected between two predetermined limits λ^- and λ^+ for every pair (p,p'). The 2-opt heuristic is then applied to improve each route of the solution obtained. The procedure is repeated γ times and the best solution found is retained. The total transportation cost of the best solution found is denoted by $\hat{C}^{du}_{lt}(\underline{P}_{lt}(\omega))$. The net revenue generated by depot $l \in \underline{L}$ on day $t \in T$ of scenario $\omega \in \Omega^n$ is given by

$$\hat{R}^{du}_{lt}(\underline{P}_{lt}(\omega)) = \sum_{p\in \underline{P}_{lt}(\omega)} \left[(u_p - v_l)d_{pt}(\omega) - \sum_{k\in K^{\mathrm{FTL}}_{plt}(\omega)} w_k y^{\mathrm{FTL}}_{kt}(\omega)\right] - \hat{C}^{du}_{lt}(\underline{P}_{lt}(\omega)). \quad (23)$$

The user problem heuristic proposed is summarised in Figure 5, in the procedure **User**. This procedure can be used to evaluate any given network design $\mathbf{x}$ under any given scenario $\omega \in \Omega$. The total net revenues generated by design $\mathbf{x}$ for the scenario ω considered is obtained simply by summing net revenues over all depots and days, i.e., by calculating $\hat{R}^{du}(\mathbf{x},\omega) = \sum_{l\in\underline{L}}\sum_{t\in T}\hat{R}^{du}_{lt}(\underline{P}_{lt}(\omega))$. When a sample Ω^n of n Monte Carlo scenarios is used, an estimate $\bar{R}_n(\mathbf{x})$ of the expected value of the design considered is thus given by

$$\bar{R}_n(\mathbf{x}) = \frac{1}{n}\sum_{\omega\in\Omega^n}\sum_{l\in\underline{L}}\sum_{t\in T}\hat{R}^{du}_{lt}(\underline{P}_{lt}(\omega)) - \sum_{l\in\underline{L}} A_l. \quad (24)$$

User$((b_k, k \in K^{\mathrm{FTL}}_{plt}(\omega)), d_{pt}(\omega), p \in \underline{P}_{lt}(\omega); \gamma, \lambda^-, \lambda^+; \hat{R}^{du}_{lt}(\underline{P}_{lt}(\omega)))$

S1: (a) Solve (19) by inspection to obtain the FTL shipments

$$y^{\mathrm{FTL}}_{klt}(\omega),\ k \in K^{\mathrm{FTL}}_{plt}(\omega),\ p \in \underline{P}_{lt}(\omega).$$

(b) Compute the residual loads $\bar{d}_{pt}(\omega)$, $p \in \underline{P}_{lt}(\omega)$, with (1).

S2: (a) Select the best direct delivery transportation modes (LTL vs. STL), and compute their costs $w_{(l,p)}$, $p \in \underline{P}_{lt}(\omega)$, with (20).

(b) Solve the resulting *open* routing problem γ times with the modified Clarke and Wright (1964) algorithm using the savings $e_{pp'}$, $p \in \underline{P}_{lt}(\omega)$, $p' \in \underline{P}_{lt}(\omega)\backslash\{p\}$ computed with (22), and retain the best transportation solution.

S3: Compute the net revenue $\hat{R}^{du}_{lt}(\underline{P}_{lt}(\omega))$ with (23).

Figure 5 Procedure User for Depot *l* in Period *t* Under Scenario ω

Tabu Search Heuristic for the Distribution Network Design Problem

The heuristic proposed to solve the SAA program (14–18) is based on tabu search, a local iterative approach that is able to escape from local optima by allowing a degradation of the objective function as opposed to pure descent methods. Because nonimproving moves are allowed, solutions previously encountered during the search may be revisited. To avoid this cycling phenomenon, a short-term memory called a tabu list keeps track of the most recent moves. The interested reader will find more details about this approach in Glover and Laguna (1997).

At each iteration of the tabu heuristic, the current design $\mathbf{x}^c$ available is improved. Potential nontabu solutions in the neighbourhood of $\mathbf{x}^c$ are evaluated with a fast but approximate route cost estimation formula. The best potential solution found is then evaluated more precisely with **User** to determine if it is better than the best solution $\mathbf{x}^*$ found to date. The following paragraphs describe the main features of the tabu search heuristic proposed as well as three alternative neighbourhood exploration strategies.

Neighbourhood Structure. At each iteration, a new design $\mathbf{x}^m$ is generated from the current design $\mathbf{x}^c$ using one of the following three moves: (i) drop (close an opened distribution center and reassign its ship-to points); (ii) add (open a closed distribution center and assign some ship-to points to it); or (iii) shift (close an opened distribution center and open a closed one while modifying some ship-to point assignments).

Let M^c be the set of all possible moves from the current design $\mathbf{x}^c$ at an iteration of the algorithm. Then, using the operator $\oplus$ to denote a move, the *general neighbourhood* of design $\mathbf{x}^c$ is defined by the feasible solution set $GN(\mathbf{x}^c) = \{\mathbf{x}^m \mid \mathbf{x}^m = \mathbf{x}^c \oplus m,\ m \in M^c\}$. Unfortunately, the size of $GN(\mathbf{x}^c)$ increases rapidly with the number of ship-to points and potential locations, and it would be too time-consuming to assess all the potential solutions in $GN(\mathbf{x}^c)$. We therefore restrict the search for a better design to moves associated to dropping, adding, or shifting depots combined with a greedy reassignment of ship-to points based on a unitary net revenue maximisation rule. Also, using the concept of *area of influence* introduced by Nagy and Salhi (1996a), the moves considered are restricted to adjacent depots. Two depots are neighbours if there exists at least one ship-to point for which they are, respectively, the nearest and the second-nearest depots. Let $\mathcal{N}(l)$ be the set of neighbours of depot l. Then, for each open depot $l \in \underline{L}^c$ specified by the current design $\mathbf{x}^c$, we define a *region* $\mathcal{R}(l)$ including depot l, the depots in $\mathcal{N}(l)$, and the ship-to points $\underline{P}^c_{l'}$, $l' \in \{l\} \cup (\mathcal{N}(l) \cap \underline{L}^c)$ assigned to these depots. In our algorithm, only drop, add, or shift moves within regions are considered. These features limit the search for a better design to a *restricted neighbourhood* $RN(\mathbf{x}^c) \subset GN(\mathbf{x}^c)$ defined by limited move subsets $\mathcal{M}(l) \subset M^c$, $l \in \underline{L}^c$. For a given region $\mathcal{R}(l)$, the feasible moves considered are:

(i) *Drop* move $(m_{\text{drop}}(l))$: If $\mathcal{N}(l) \cap \underline{L}^c \neq \varnothing$, close depot l and assign each ship-to point $p \in \mathcal{R}(l) \cap P$ to depot $l_p = \arg\max_{l' \in \mathcal{N}(l) \cap \underline{L}^c_p} (u_p - v_{l'}) - w_{(l', p)}$.

(ii) *Add* moves $(\mathcal{M}_{\text{add}}(l))$: If $\mathcal{N}(l) \cap \bar{L}^c \neq \varnothing$, for all $l' \in \mathcal{N}(l) \cap \bar{L}^c$, open depot l' and assign each ship-to point $p \in \mathcal{R}(l) \cap P$ to depot $l_p = \arg\max_{l'' \in (\mathcal{R}(l) \cap \underline{L}^c_p) \cup \{l'\}} (u_p - v_{l''}) - w_{(l'', p)}$.

(iii) *Shift* moves $(\mathcal{M}_{\text{shift}}(l))$: If $\mathcal{N}(l) \cap \bar{L}^c \neq \varnothing$, for all $l' \in \mathcal{N}(l) \cap \bar{L}^c$, open depot l' and close depot l simultaneously and assign each ship-to point $p \in \mathcal{R}(l) \cap P$ to depot $l_p = \arg\max_{l'' \in (\mathcal{N}(l) \cap \underline{L}^c_p) \cup \{l'\}} (u_p - v_{l''}) - w_{(l'', p)}$.

In what follows, the following sets of possible moves are used:

$$\mathcal{M}_{\text{drop}} = \{m_{\text{drop}}(l)\}_{l \in \underline{L}^c}, \quad \mathcal{M}_{\text{add}} = \bigcup_{l \in \underline{L}^c} \mathcal{M}_{\text{add}}(l),$$

$$\mathcal{M}_{\text{shift}} = \bigcup_{l \in \underline{L}^c} \mathcal{M}_{\text{shift}}(l) \quad \mathcal{M} = \mathcal{M}_{\text{drop}} \cup \mathcal{M}_{\text{add}} \cup \mathcal{M}_{\text{shift}},$$

$$\mathcal{M}(l) = \{m_{\text{drop}}(l)\} \cup \mathcal{M}_{\text{add}}(l) \cup \mathcal{M}_{\text{shift}}(l).$$

Note that only the moves leading to a feasible solution are considered. For example, if a drop move creates an isolated ship-to point, i.e., a point too far from other depots to permit next-day delivery, then it is not considered. Note also that during the neighbourhood exploration, the moves in $\mathcal{M}$ are evaluated only if they are not tabu.

Neighbourhood Evaluation. The evaluation of a design $\mathbf{x}^m \in RN(\mathbf{x}^c)$ involves the estimation of its expected value for the sample Ω^n of Monte Carlo scenarios generated. This could be done by applying the **User** heuristic to all $(l, \omega, t) \in \underline{L}^m \times \Omega^n \times T$ and then computing $\bar{R}_n(\mathbf{x}^m)$ with (24). However, evaluating all possible moves using the **User** heuristic would be too time-consuming. Possible moves must thus be evaluated using a fast approximation. The evaluation function proposed is based on a linear regression route length estimator (introduced by Daganzo 1984 and extended by Nagy and Salhi 1996b) modified to account for LTL transportation. It estimates the total transportation cost $\tilde{C}^{du}_{lt}(\underline{P}^m_{lt}(\omega))$ of depot l on day t under scenario ω as a function of the set of ship-to points $\underline{P}^m_{lt}(\omega)$ visited from depot l on that day under design $\mathbf{x}^m$. The function is obtained with a multiple regression model based on three explanatory variables: (i) a *linehaul* variable $(\xi_l(\underline{P}^m_{lt}(\omega))/NS_l)$, where $\xi_l(P)$ is the sum of the distances from depot l to all $p \in P$, and NS_l the average number of stops on the routes from depot l; (ii) a *detour* variable $(\xi_l(\underline{P}^m_{lt}(\omega))/\sqrt{|\underline{P}^m_{lt}(\omega)|})$; and (iii) an LTL variable $(\rho^{\text{LTL}}_l \varphi_l(\underline{P}^m_{lt}(\omega)))$, where $\varphi_l(P)$ is the sum of the load distances from depot l to all $p \in P$ and is ρ^{LTL}_l the proportion of load distances from depot l on LTL routes. This leads to the following cost approximation formula:

$$\begin{aligned}\hat{C}^{du}_{lt}(\underline{P}^m_{lt}(\omega)) &\approx \tilde{C}^{du}_{lt}(\underline{P}^m_{lt}(\omega)) \\ &= \hat{\beta}_1\left(\frac{\xi_l(\underline{P}^m_{lt}(\omega))}{NS_l}\right) + \hat{\beta}_2\left(\frac{\xi_l(\underline{P}^m_{lt}(\omega))}{\sqrt{|\underline{P}^m_{lt}(\omega)|}}\right) \\ &\quad + \hat{\beta}_3(\rho^{\text{LTL}}_l \varphi_l(\underline{P}^m_{lt}(\omega))),\end{aligned} \tag{25}$$

where $\hat{\beta}_1$, $\hat{\beta}_2$, and $\hat{\beta}_3$ are regression coefficients associated to the *linehaul*, *detour*, and LTL variables, respectively. These regression coefficients are estimated using a sample of historical daily delivery routes or a sample of daily routes obtained with **User** for different network designs. The parameters NS_l and ρ^{LTL}_l are initially estimated with the same route sample, but they are updated in the algorithm every time the **User** heuristic is used to evaluate a new design.

An adequate approximation of $\bar{R}_n(\mathbf{x}^m)$ is then obtained by replacing $\hat{C}^{du}_{lt}(\underline{P}^m_{lt}(\omega))$ in (23) by $\tilde{C}^{du}_{lt}(\underline{P}^m_{lt}(\omega))$ and then by substituting resulting expression in (24), to get:

$$\begin{aligned}\tilde{R}^{du}_{lt}(\underline{P}^m_{lt}(\omega)) &= \sum_{p \in \underline{P}^m_{lt}(\omega)} \left[(u_p - v_l) d_{pt}(\omega) - \sum_{k \in K^{\text{FTL}}_{plt}(\omega)} w_k y^{\text{FTL}}_{kt}(\omega)\right] \\ &\quad - \tilde{C}^{du}_{lt}(\underline{P}^m_{lt}(\omega)),\end{aligned} \tag{26}$$

$$\tilde{R}_n(\mathbf{x}^m) = \frac{1}{n} \sum_{\omega \in \Omega^n} \sum_{l \in \underline{L}^m} \sum_{t \in T} \tilde{R}^{du}_{lt}(\underline{P}^m_{lt}(\omega)) - \sum_{l \in \underline{L}^m} A_l. \tag{27}$$

Restricted Neighbourhood Exploration. Given the restricted neighbourhood structure previously defined, three different strategies are considered to explore the neighbourhood $RN(\mathbf{x}^c)$ of the current solution $\mathbf{x}^c$. These strategies specify the order in which the drop, add, and shift moves are made during each iteration of the algorithm. These three strategies have the following characteristics.

Strategy 1 (S1): This strategy is inspired from the tabu search approach proposed by Nagy and Salhi (1996a) to solve a deterministic location-routing problem. It is a straightforward version of the tabu search method. At each iteration, all the drop, add, and shift moves in $\mathscr{M}$ are considered to generate a new solution.

Strategy 2 (S2): This strategy is an extension of the three-phase hill-climbing method proposed by Kuehn and Hamburger (1963) to solve a deterministic location-allocation problem. Assuming that a minimum number of depots are initially opened, the first phase explores add moves $\mathscr{M}_{\text{add}}$ only, the second phase considers drop moves $\mathscr{M}_{\text{drop}}$ exclusively, and the third phase concentrates on shift moves $\mathscr{M}_{\text{shift}}$. In our implementation, only nontabu shift moves are considered.

Strategy 3 (S3): This strategy starts with the initial solution obtained by assigning each ship-to point to the feasible depot, with the maximum marginal net revenue. Clearly, in this solution, all the potentially interesting depots are opened. Then, at each iteration, a drop move in $\mathscr{M}_{\text{drop}}$ is performed, followed by shift moves in $\mathscr{M}_{\text{shift}}$. This process continues until the specified maximum number of iterations is reached. It is worth noting that the shift moves made at each iteration can be viewed as an intensification phase in the region of the search space that contains solutions having the same number of opened depots as the current solution. In addition, following each shift move, a reassignment procedure for borderline points is applied. This intensification phase is a tabu search because nonimproving moves are allowed.

Note that the moves $m(l) \in \mathscr{M}$ considered are based on changes to the status of depots, followed by a greedy adjustment to ship-to point assignments to ensure that the resulting design is feasible. There is no guarantee, however, that the assignments adjustment made is optimal. It may therefore be profitable, when all moves are performed, to refine the adjustments made to get the new solution $\mathbf{x}^c$. To do this, we elaborated a reassignment procedure for borderline points, inspired from an heuristic proposed by Zainuddin and Salhi (2007) to solve the capacitated Weber problem. A point p assigned to depot $l \in \underline{L}^m$ (i.e., in $\underline{P}_l^m$) is considered borderline if its distance from depot l exceeds a predefined target distance $m_{\max}$, i.e., if $m_{(l,p)} > m_{\max}$. Let $\mathscr{B}$ be the set of borderline ship-to points. A ship-to point p is reassigned from depot l to

Reassign$(\mathbf{x}^c; m_{\max}, \Upsilon_{\max}; \mathbf{x}', \tilde{R}_n(\mathbf{x}'))$
Set $\mathbf{x}' = \mathbf{x}^c$ and compute $\tilde{R}_n(\mathbf{x}')$ with (27)
Construct the borderline point set $\mathscr{B}$
For all $\rho_{\max} \in \Upsilon_{\max}$, do
  $\mathbf{x}^m = \mathbf{x}'$
  For all $p \in \mathscr{B}$, do

$$l_p = \arg\min_{l' \in \underline{L}_p^m} \{\rho'_l = m_{(l',p)}/m_{(l,p)} \mid \rho'_l \le \rho_{\max}\}, \quad x^m_{lp} = 0,\ x^m_{l_p,p} = 1.$$

  Compute $\tilde{R}_n(\mathbf{x}^m)$ with (27).
  If $(\tilde{R}_n(\mathbf{x}^m) > \tilde{R}_n(\mathbf{x}'))$, then $\mathbf{x}' = \mathbf{x}^m$ and $\tilde{R}_n(\mathbf{x}') = \tilde{R}_n(\mathbf{x}^m)$
End do

Figure 6 Reassignment Procedure

depot $l_p \in \underline{L}_p^m$ if its distance ratio $\rho_{l_p} = m_{(l_p,p)}/m_{(l,p)}$ is the lowest among the eligible depots, provided that it does not exceed a predefined value $\rho_{\max}$. When this is done for all the borderline points $p \in \mathscr{B}$ associated to $\mathbf{x}^m$, a new solution vector $\mathbf{x}'$ results. Several alternative solution vectors $\mathbf{x}'$ can be generated by considering a set $\Upsilon_{\max}$ of $\rho_{\max}$ values. The solutions thus generated can then be evaluated with (27) to determine which one is the best. This gives rise to the reassignment procedure presented in Figure 6 and applied in the three strategies.

Tabu Lists. During the search procedure, two tabu lists of varying length are kept. When a depot l is added or dropped in the new current solution, l is inserted into a drop or add list $\mathscr{T}_1$. On the other hand, when the new current solution is obtained by shifting two depots l and l', the pair (l, l') is inserted into a shift list $\mathscr{T}_2$. In both cases, the oldest element of the list is removed. Note that when a depot l is inserted in $\mathscr{T}_1$, all the shifts involving this depot are also inserted in $\mathscr{T}_2$. At each iteration, the length of the relevant list is randomly generated in the interval $[\alpha_1|L|/2, |L|/2]$ for $\mathscr{T}_1$ and interval $[\alpha_2|L|, 2|L|]$ for $\mathscr{T}_2$, where $\alpha_1, \alpha_2 \in]0,1[$ are predefined parameters. Note that these interval bounds have been adequately fixed after several preliminary tests and are close to those considered in Nagy and Salhi (1996a).

Solution Algorithm Initialization. Before the tabu search is started, the solution process must be initialized. This involves first fixing the various parameters required by the procedures used. In addition to the parameters already defined, the following two parameters are required to control the tabu search:

MI = maximum number of iterations,
MNI = the maximum number of iterations without improvement.

Next, a sample of n demand scenarios $\mathbf{d}(\omega) = [d_{pt}(\omega)]_{p\in P,\, t\in T}$, $\omega \in \Omega^n$ must be generated using procedure **MonteCarlo** n times, and the neighbour sets $\mathscr{N}(l)$, $l \in L$ must be created. Initial solutions must then be constructed for the neighbourhood exploration

Initialize(strategy; $n, \gamma, \lambda^-, \lambda^+, \alpha_1, \alpha_2, m_{\max}, \Upsilon_{\max}$, MI, MNI, $(\mathbf{d}(\omega), \omega \in \Omega^n), (\mathcal{N}(l), l \in L), \mathbf{x}^o, \bar{R}_n(\mathbf{x}^o)$)
Set the heuristic parameters ($n, \gamma, \lambda^-, \lambda^+, \alpha_1, \alpha_2, m_{\max}, \Upsilon_{\max}$, MI, MNI)
For all $\omega \in \Omega^n$, do **MonteCarlo**($(F_p^q(\cdot), F_p^o(\cdot), p \in P), T; d_{pt}(\omega), p \in P, t \in T$)
Obtain the neighbour sets $\mathcal{N}(l), l \in L$
If strategy $= 2$, then
Set $L^0 = \varnothing$, $\bar{P} = P$
While $\bar{P} \neq \varnothing$, do

$$l' = \arg\max_{l \in \bar{L}^0} \sum_{p \in \bar{P} \cap P_l} (u_p - v_l) - w_{(l,p)}$$

Set $L^0 = L^0 \cup \{l'\}$, $\underline{P}_{l'} = \bar{P} \cap P_{l'}$ and $\bar{P} = \bar{P} \backslash \underline{P}_{l'}$
End while
Else set $\underline{L}^0 = L$
Construct an initial design $\mathbf{x}^o$ by assigning each ship-to point $p \in P$ to the depot
$l_p = \arg\max_{l \in \underline{L}_p^0} (u_p - v_l) - w_{(l,p)}$
For all $(l, \omega, t) \in \underline{L}^o \times \Omega^n \times T$, do
User($(b_k, k \in K_{plt}^{\mathrm{FTL}}(\omega)), d_{pt}(\omega), p \in \underline{P}_{lt}^o(\omega); \gamma, \lambda^-, \lambda^+; \hat{R}_{lt}^{du}(\underline{P}_{lt}^o(\omega))$)
Compute the expected value $\bar{R}_n(\mathbf{x}^o)$ with (24)

Figure 7 Initialization Procedure

strategy considered. For Strategy 2, an initial solution $\mathbf{x}^0$ is obtained by sequentially opening the closed depot $l \in \bar{L}$ maximizing the marginal net revenues of the ship-to points $\bar{P}$ not yet allocated. For Strategies 1 and 3, $\underline{L}^0$ is initially set to L. Then, for all cases, the missions are obtained by assigning each ship-to point $p \in P$ to the depot $l_p \in L_p$, with the maximum marginal net revenues. The resulting design $\mathbf{x}_0$ is then evaluated with the **User** heuristic and the expected value function (24). Figure 7 presents the initialization procedure thus obtained.

Tabu Search Procedure. The **Tabu** search procedure proposed appears in Figure 8. The iterations of the algorithm are controlled by two parameters: *iter*, the current number of iterations, and *iter_ni*, the current number of iterations without improvement. In Figure 8, Step 2 examines different moves in the neighbourhood of the current solution $\mathbf{x}^c$. Step 3 applies the reassignment procedure to the current solution $\mathbf{x}^c$. Step 4 evaluates the best move and updates the parameters of the transportation costs estimation function. Step 5 manages the tabu lists. Step 6 performs a shift move intensification when the parameter "Improve" equals one. Step 7 checks if the last iteration improved the best solution to date $\mathbf{x}^*$. Finally, Steps 8 and 9 control the iterations of the algorithm. If either of the iteration limits is not reached, the exploration continues from Step 2 with the current solution $\mathbf{x}^c$. If *MNI* iterations were made without improvement, the exploration continues from Step 1 with the best solution to date $\mathbf{x}^*$. Otherwise, the algorithm stops. Note that to simplify the exposition, we

Tabu($\mathbf{x}^o, \bar{R}_n(\mathbf{x}^o), \mathcal{M}$; MI, MNI, improve; $\mathbf{x}^*, \bar{R}_n(\mathbf{x}^*)$)
Step 0. Set $\mathbf{x}^* = \mathbf{x}^o$ and $\bar{R}_n(\mathbf{x}^*) = \bar{R}_n(\mathbf{x}^o)$ and initialize the tabu lists ($\mathcal{T}_1$ and $\mathcal{T}_2$).
Step 1. Set $\mathbf{x}^c = \mathbf{x}^*$ and $\bar{R}_n(\mathbf{x}^c) = \bar{R}_n(\mathbf{x}^*)$
Step 2. $\tilde{R}_n(\mathbf{x}') = 0$
For all $l \in \underline{L}^c$, do
Construct the region $\mathcal{R}(l)$
For all $m(l) \in \mathcal{M}(l)$ and $m(l)$ not tabu, do
Set $\mathbf{x}^m = \mathbf{x}^c \oplus m(l)$ and compute $\tilde{R}_n(\mathbf{x}^m)$ with (27)
If $(\tilde{R}_n(\mathbf{x}^m) > \tilde{R}_n(\mathbf{x}'))$, then $\mathbf{x}' = \mathbf{x}^m$ and $\tilde{R}_n(\mathbf{x}') = \tilde{R}_n(\mathbf{x}^m)$
End do
End do
$\mathbf{x}^c = \mathbf{x}'$
Step 3. **Reassign**($\mathbf{x}^c; m_{\max}, \Upsilon_{\max}; \mathbf{x}', \tilde{R}_n(\mathbf{x}')$) and set $\mathbf{x}^c = \mathbf{x}'$
Step 4. For all $(l, \omega, t) \in \underline{L}^c \times \Omega^n \times T$, do
User($(b_k, k \in K_{plt}^{\mathrm{FTL}}(\omega)), d_{pt}(\omega), p \in \underline{P}_{lt}^c(\omega); \gamma, \lambda^-, \lambda^+; \hat{R}_{lt}^{du}(\underline{P}_{lt}^c(\omega))$)
Compute the expected value $\bar{R}_n(\mathbf{x}^c)$ with (24)
Update the means NS_l and ρ_l^{LTL}, $l \in \underline{L}^c$, using the augmented set of generated routes
Step 5. Update the tabu lists ($\mathcal{T}_1$ and $\mathcal{T}_2$)
Step 6. If (improve $= 1$), then
Tabu($\mathbf{x}^c, \bar{R}_n(\mathbf{x}^c), \mathcal{M}_{\text{shift}}$; MI, MNI, 0; $\mathbf{x}', \bar{R}_n(\mathbf{x}')$)
$\mathbf{x}^c = \mathbf{x}'$ and $\bar{R}_n(\mathbf{x}^c) = \bar{R}_n(\mathbf{x}')$
Step 7. If $(\bar{R}_n(\mathbf{x}^c) > \bar{R}_n(\mathbf{x}^*))$, then $\mathbf{x}^* = \mathbf{x}^c$, $\bar{R}_n(\mathbf{x}^*) = \bar{R}_n(\mathbf{x}^c)$ and $iter\_ni = 0$
Else $iter\_ni = iter\_ni + 1$
Step 8. $iter = iter + 1$
Step 9. If ($iter <$ MI) and ($iter\_ni < MNI$), then go to Step 2
Else if ($iter < MI$), then set $iter\_ni = 0$ and go to Step 1
Else stop

Figure 8 Tabu Procedure

used the parameters MI and MNI when calling **Tabu** in Step 6 as in the main procedure. In the implementation, however, the iteration limits parameters that are used in Step 6 are not the same as in the main procedure.

Using the **Initialize** and **Tabu** procedures described previously, the neighbourhood exploration strategies considered are implemented as follows:

Strategy 1 (S_1):
Initialize(1; $n, \gamma, \lambda^-, \lambda^+, \alpha_1, \alpha_2, m_{\max}, \Upsilon_{\max}$, MI, MNI, $(\mathbf{d}(\omega), \omega \in \Omega^n), (\mathcal{N}(l), l \in L), \mathbf{x}^o, \bar{R}_n(\mathbf{x}^o)$)
Tabu($\mathbf{x}^o, \bar{R}_n(\mathbf{x}^o), \mathcal{M}$; MI, MNI, 0; $\mathbf{x}^*, \bar{R}_n(\mathbf{x}^*)$)
Strategy 2 (S_2):
Initialize(2; $n, \gamma, \lambda^-, \lambda^+, \alpha_1, \alpha_2, m_{\max}, \Upsilon_{\max}$, MI, MNI, $(\mathbf{d}(\omega), \omega \in \Omega^n), (\mathcal{N}(l), l \in L), \mathbf{x}^o, \bar{R}_n(\mathbf{x}^o)$)
Tabu($\mathbf{x}^o, \bar{R}_n(\mathbf{x}^o), \mathcal{M}_{\text{add}}$; MI, MNI, 0; $\mathbf{x}^{\text{add}}, \bar{R}_n(\mathbf{x}^{\text{add}})$)
Tabu($\mathbf{x}^{\text{add}}, \bar{R}_n(\mathbf{x}^{\text{add}}), \mathcal{M}_{\text{drop}}$; MI, MNI, 0; $\mathbf{x}^{\text{drop}}, \bar{R}_n(\mathbf{x}^{\text{drop}})$)
Tabu($\mathbf{x}^{\text{drop}}, \bar{R}_n(\mathbf{x}^{\text{drop}}), \mathcal{M}_{\text{shift}}$; MI, MNI, 0; $\mathbf{x}^*, \bar{R}_n(\mathbf{x}^*)$)
Strategy 3 (S_3):
Initialize(3; $n, \gamma, \lambda^-, \lambda^+, \alpha_1, \alpha_2, m_{\max}, \Upsilon_{\max}$, MI, MNI, $(\mathbf{d}(\omega), \omega \in \Omega^n), (\mathcal{N}(l), l \in L), \mathbf{x}^o, \bar{R}_n(\mathbf{x}^o)$)
Tabu($\mathbf{x}^o, \bar{R}_n(\mathbf{x}^o), \mathcal{M}_{\text{drop}}$; MI, MNI, 1; $\mathbf{x}^*, \bar{R}_n(\mathbf{x}^*)$).

The next section evaluates and compares these strategies.

Computational Results

Plan of Experiments

To test the heuristic approach proposed to solve the SMLTP, several problem instances were generated based on the following four dimensions: problem size, cost structure, demand process, and network characteristics. The problem instances were generated randomly but they were based on realistic parameter value ranges obtained partially from the "Usemore" case documented in Ballou (1992) and from the data of a real case. The problems were defined over various U.S. regions and all distances were calculated with PC∗MILER (www.alk.com) for the current U.S. road network. For all cases, it was assumed that the order interarrival times are exponentially distributed with a mean inter-arrival time λ and that order sizes are log normal with a mean μ and a standard deviation σ. A one-year planning horizon, assumed to include $|T| = 200$ working days, was used.

Problems of three different sizes were tested—small (P_1), medium (P_2), and large (P_3)—as defined in Table 1. In each case, the problem size varies in terms of the number of depots and ship-to points considered, and in terms of the U.S. geographical regions covered. Based on the realistic industrial problems examined, the number of ship-to points in the problems is much larger than the number of potential depots, and the latter was fixed at about 3% $|P|$. To capture different cost structures, two levels of fixed and variable costs were considered (see Table 2). The fixed operating costs A_l and the unit value of products v_l were selected randomly in the interval specified in Table 2. The product prices on the market u_p was fixed equal to the value in Table 2 for all ship-to points.

Next, demand processes are associated to the geographical coordinates of the ship-to points in a problem. These demand processes are calibrated to represent large, medium, or small customers. Two types of network are generated: (1) those composed mainly of large- and medium-sized customers (LN), and (2) those including mainly small- and medium-sized customers (SN). Table 3 provides the proportion of each type of customer in LN and SN networks, as well as the (λ, μ, σ) parameter values range used to generate specific instances. Finally, two random replications $(\mathrm{DS}^1, \mathrm{DS}^2)$ are generated for each network structure considered.

Table 1 Test Problems Sizes

Problem instance	Geographical area	Number of depots	Number of ship-to points
P_1	Central Northeastern United States	7	206
P_2	Northeastern and Midwest United States	15	706
P_3	Northeastern and Midwest United States	28	1,206

Table 2 Test Problems Cost Structures

	High product value and price	Low product value and price
High fixed costs	(a) [230K, 250K]; [19, 21]; 23	(b) [230K, 250K]; [9, 11]; 13
Low fixed costs	(c) [130K, 150K]; [19, 21]; 23	(d) [130K, 150K]; [9, 11]; 13

Note. (Cost structure): [fixed cost (A_l) range]; [product value (v_l) range]; product price (u_p).

The combination of these four dimensions yields 48 problem instances. Each instance is denoted as follows:

$$(i, j, k, l) \quad i \in \{P_1, P_2, P_3\},\ j \in \{a, b, c, d\},\ k \in \{\mathrm{LN}, \mathrm{SN}\},\ l \in \{\mathrm{DS}^1, \mathrm{DS}^2\}.$$

The three neighbourhood exploration strategies proposed in the previous section were tested for all of these problem instances and the numerical results obtained are presented in the next paragraphs.

Numerical Results

The heuristics proposed were implemented in VB.Net 2005 and the experiments reported in this section were performed on a 2 GHz Dual Core workstation with 3 GB of RAM. This section starts with a discussion of the calibration of the several procedures used in the heuristics: Preliminary tests on various problem instances were performed to fix the algorithm parameters and to study the stochastic behaviour of the solutions obtained. We then provide a comprehensive analysis of performances of the 3 neighbourhood exploration strategies considered for the 48 problem instances. In addition, for the small problem (P_1) instances, the heuristic is compared to the optimal solution of the SAA model (14–18) obtained with CPLEX-11 when using all possible nondominated routes. These problems were solved on a 64-bit server with a 2.5 GHz Intel XEON processor and 16 GB of RAM.

Procedures Calibration. The solution approach is based on several procedures including a number of

Table 3 Ship-to Point Demand Structure

Ship-to point sizes and characteristics	Large	Medium	Small
Larger ship-to points network (LN) (%)	**15**	**65**	**20**
Smaller ship-to points network (SN) (%)	**10**	**30**	**60**
μ (cwt)	[480, 580]	[300, 400]	[120, 220]
σ (% μ) (%)	7	10	16
λ (days)	[2.5, 4.5]	[5.5, 15.5]	[20.5, 35.5]

Table 4 Heuristic Parameter Values

Procedure	User			Reassign		Tabu			
Parameters	γ	λ^-	λ^+	$m_{\max}$	$\Upsilon_{\max}$	α_1	α_2	MI	MNI
Values range	[1, 50]	[0.5, 1.5]	[1, 2]	[50, 400]	[0, 1.5]	[0, 1]	[0, 1]	[20, 200]	[0, 20]
Selected value	10	1	1.2	250	{1, 0.75, 0.5}	0.25	0.25	100	10

parameters that were calibrated with a set of preliminary experiments. Using several P_1 and P_2 instances, the three strategies were executed alternatively to fix the heuristic parameters specified in the initialization procedure. Table 4 presents the range of values tested for each parameter and provides the value selected. Note that these values were fixed for the three strategies and present the best trade-off in terms of algorithm search speed and solution accuracy.

Another important parameter to calibrate is the number n of scenarios to generate (with the **Monte-Carlo** procedure) to obtain good solutions. It is important to note that because our demand process is stationary, the user problem must be solved by each depot on a daily basis, and the planning horizon includes 200 days, when n scenarios are used, 200 n user model instances sampled from the same probability distributions are solved by each depot. For this reason, a relatively small number of scenarios is required to obtain good results. To determine the best value of n, different sample sizes were tested and the quality of the solutions obtained was evaluated using a statistical optimality gap, as is usually done when the SAA method is used to solve stochastic programs. This calibration was done using problem P_1 with larger ship-to points (LN), and it was based on the optimal solution of the SAA model (14–18) obtained with CPLEX-11. Values of $n = 1, 2, 4, 6, 8$ were tested. We were not able to solve larger problems to optimality without truncating the set of non-dominated routes.

To estimate the statistical optimality gap for a sample size n, several SAA models based on independent samples of size n must be solved. Let $\bar{R}_n^j$ and $\hat{\mathbf{x}}_n^j$, $j = 1, \ldots, m$ be the optimal value and an optimal solution of the SAA model for the m samples used. Well-known results in stochastic programming are that $\mathbb{E}[\bar{R}_n] \geq R^*$, where $\mathbb{E}[]$ denotes the expected value, $\bar{R}_n$ is the optimal value of (14), and R^* is the optimal value of (6), and that $\bar{R}_{n,m} = \sum_{j=1}^m \bar{R}_n^j/m$ is an unbiased estimator of $\mathbb{E}[\bar{R}_n]$ (Shapiro 2003). A statistical upper bound on the optimal value is thus provided by $\bar{R}_{n,m} \geq R^*$. Also, the true objective function value $R(\hat{\mathbf{x}}_n^j) \leq R^*$ of the feasible solutions $\hat{\mathbf{x}}_n^j$, $j = 1, \ldots, m$ can be estimated with

$$\bar{R}_{n'}(\hat{\mathbf{x}}_n^j) = \frac{1}{n'} \sum_{\omega \in \Omega^{n'}} R^{du}(\hat{\mathbf{x}}_n^j, \omega) - \sum_{l \in L} A_l(\hat{x}_l)_n^j, \tag{28}$$

where $\Omega^{n'} \subset \Omega$ is a sample of n' scenarios generated independently of the sample used to obtain $\hat{\mathbf{x}}_n^j$, and with $n' \gg n$. A statistical lower bound on the optimal value is thus provided by $\bar{R}_{n'}(\hat{\mathbf{x}}_n^j) \leq R^*$. Furthermore, if the **User** heuristic is used to solve the second-stage program, its value $R^{du}(\hat{\mathbf{x}}_n^j, \omega)$ in (28) must be replaced by the value $\hat{R}^{du}(\hat{\mathbf{x}}_n^j, \omega) \leq R^{du}(\hat{\mathbf{x}}_n^j, \omega)$ calculated with **User** to get $\hat{R}_{n'}(\hat{\mathbf{x}}_n^j) \leq \bar{R}_{n'}(\hat{\mathbf{x}}_n^j)$. An estimate of the optimality gap of the solution $\hat{\mathbf{x}}_n^j$ can thus be calculated as follows: $\text{gap}_{n,m,n'}(\hat{\mathbf{x}}_n^j) = \bar{R}_{n,m} - \hat{R}_{n'}(\hat{\mathbf{x}}_n^j)$. Because m SAA models with a sample of size n are solved, an estimate of the average gap for a sample of size n is given by $\text{gap}_{n,n'} = \sum_{j=1}^m \text{gap}_{n,m,n'}(\hat{\mathbf{x}}_n^j)/m$.

To evaluate alternative sample sizes, we calculated this average gap with $m = 4$ sample replications, and we evaluated (28) with the **User** heuristic using scenario samples of size $n' = 100$. The average gap values obtained for different sample sizes are provided in Table 5. These values are expressed as a percentage of the objective function value of the best design found. It can be seen that samples of six or eight scenarios provide very good results. When inspecting the design found with these sample sizes, we noted that they were all very similar: They opened the same depots and only one or two ship-to point assignments were different. However, the solution time increases considerably with the sample size. For these reasons, a sample size of $n = 6$ scenarios has been selected as the best trade-off to use in the experiments. This means that 1,200 user model instances, sampled from the same probability distributions, are solved by each depot every time a design is evaluated. The gaps in Table 5 suggest that the approach proposed is extremely accurate. This is due partly to the fact that for the problem instance considered, the profits generated are relatively high and the objective function value has low variability, which could mean that the value of stochastic solution (Birge and Louveaux 1997) is relatively low.

Analysis of Results. This section discusses the quality of the supply network designs obtained with

Table 5 Statistical Optimality Gap Values

Sample size (n)	1 (%)	2 (%)	4 (%)	6 (%)	8 (%)
$\text{gap}_{n,100}$ (in %)	2.32	3.42	2.21	0.67	0.07

Table 6 Comparison with CPLEX-11 Solution of SAA Model for Problem P_1

Problem	(a, LN)	(b, LN)	(c, LN)	(d, LN)	(a, SN)	(b, SN)	(c, SN)	(d, SN)
S1	**2,689,489**	**2,493,083**	**4,204,796**	**4,145,389**	**1,430,919**	**1,306,351**	**2,396,894**	**2,360,981**
S2	2,689,162	2,492,898	**4,204,796**	**4,145,389**	**1,430,919**	1,306,142	**2,396,894**	**2,360,981**
S3	2,632,805	2,439,453	4,203,591	4,144,280	**1,430,919**	1,306,142	**2,396,894**	**2,360,981**
CPLEX-11	2,689,481	2,493,069	4,204,851	4,145,394	1,431,210	1,306,349	2,397,181	2,361,277
%-difference (%)	0.000	−0.001	0.001	0.000	0.020	0.000	0.012	0.013

the three neighbourhood exploration strategies proposed as well as their respective solution times. First, to determine how close to optimum the solutions obtained are, the SAA model (14–18) was solved with CPLEX-11 for problem P_1 instances, using an *MIP relative tolerance* of 0.001 and a sample size $n = 6$. These are the largest SAA problems we were able to solve to optimality. The results obtained with our three exploration strategies and with CPLEX are provided in Table 6. The values shown in Table 6 are the estimation of the true objective function value $R(\mathbf{x})$ provided by $\hat{R}_{n'}(\mathbf{x})$ with a sample size $n' = 100$. The last row provides the **%-difference** between the value of the CPLEX solution and the best design obtained with our heuristic search strategies. It can be seen that the differences are very small. For problem (P_1, b, LN), the design provided by S1 is slightly better than the solution provided by CPLEX. This is possible because the SAA is solved to optimality with a sample size $n = 6$, which is an approximation. For a given sample size, our heuristic can therefore provide a better solution of the original stochastic programming model (6–13) than the SAA model. The SAA models solved include 2,501,456 binary variables for the larger ship-to point networks (LN), and they were solved on average in 939 seconds by CPLEX-11 on our 64-bit server. It took on average 10 seconds to obtain the best solution found with our heuristics on a 32-bit workstation.

Next, our three exploration strategies were compared for all problem instances. The true objective function value of a design $\mathbf{x}$ obtained was estimated with $\hat{R}_{n'}(\mathbf{x})$ using a sample of $n' = 100$ scenarios. For a given problem instance, to ensure that solution strategies are compared on the same basis, the sample of $n = 6$ scenarios used in the heuristic and the sample of $n' = 100$ scenarios used to estimate the design value were the same for the three exploration strategies. The mean design value obtained for different subsets of problem instances are presented in Table 7. The dot in the problem subset labels in Table 7 denotes all instances corresponding to a problem attribute. (P_1, a, ., .), for example, is the subset of four problems of size P_1 with cost structure *a*. The value of the best search strategy for each problem subset is highlighted. More detailed comparisons of the three strategies' design value and solution times are provided in Figure 9. In the three first plots of this figure, for P_1, P_2, and P_3, respectively, the design value %-deviation from the best-known solution is given for all the instances solved. The fourth plot provides a comparison of average time (in seconds) required to find the best solution by problem size.

When looking at the detailed results, the first observation that, comes out is that even if none of the proposed strategies is completely dominated, search strategies S1 and S2 provide better results. Strategy 1 provides the best solution for 65% of the problem instances solved and it is within 0.5% of the best

Table 7 Mean Design Values for All Problem Types

P_1	(P_1, a, ., .)	(P_1, b, ., .)	(P_1, c, ., .)	(P_1, d, ., .)	(P_1, ., LN, .)	(P_1, ., SN, .)	(P_1, ., ., .)
S1	**2,119,638**	**1,957,886**	**2,426,406**	**2,884,254**	**3,255,870**	**1,908,016**	**2,581,943**
S2	2,119,556	1,957,788	2,426,062	2,883,577	3,255,103	1,907,990	2,581,546
S3	2,048,156	1,927,984	2,412,939	2,883,854	3,204,709	1,907,974	2,556,341
P_2	(P_2, a, ., .)	(P_2, b, ., .)	(P_2, c, ., .)	(P_2, d, ., .)	(P_2, ., LN, .)	(P_2, ., SN, .)	(P_2, ., ., .)
S1	**9,578,586**	**8,845,539**	**8,599,984**	9,631,837	**10,712,395**	5,780,462	**8,246,428**
S2	9,521,073	8,836,155	8,595,897	**9,633,208**	10,677,916	**5,782,614**	8,230,265
S3	9,560,879	8,809,231	8,570,404	9,616,201	10,691,160	5,758,618	8,224,889
P_3	(P_3, a, ., .)	(P_3, b, ., .)	(P_3, c, ., .)	(P_3, d, ., .)	(P_3, ., LN, .)	(P_3, ., SN, .)	(P_3, ., ., .)
S1	**20,752,448**	15,848,841	16,485,836	**17,554,079**	**23,447,483**	13,701,596	18,574,540
S2	20,645,080	**15,971,828**	**16,540,209**	17,547,339	23,437,516	**13,726,087**	**18,581,802**
S3	20,640,702	15,663,930	16,389,856	17,530,314	23,325,136	13,638,200	18,481,668

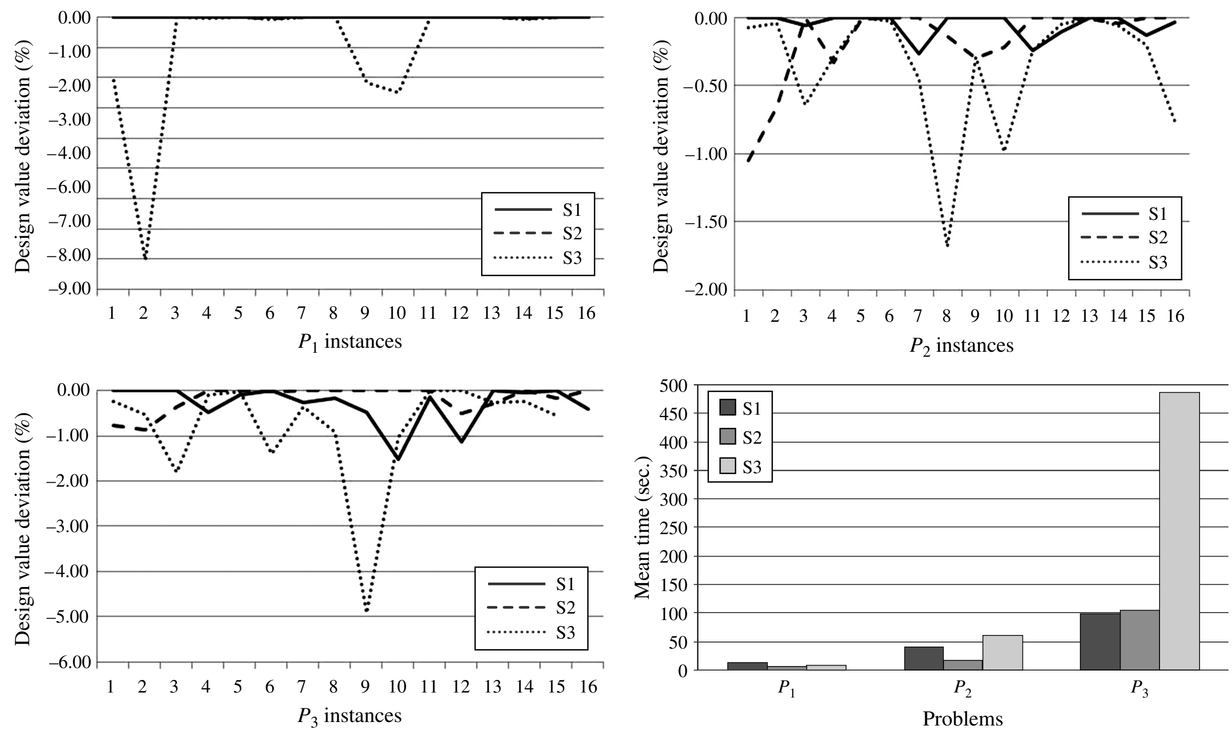

Figure 9 **Exploration Strategies Design Value and Solution Time Comparisons**

solution found for 96% of the problems solved. This is mainly because this strategy converges quickly to a solution with a good number of depots and then performs several moves to improve this solution. It usually finds the best solution during the initial search iterations. Strategy 1 is also very fast: The average solution times for P_1, P_2, and P_3 are 13, 43, and 292 seconds, respectively. Strategy 2 provides very good results for P_2 and P_3, but it is relatively dependent on the quality of its initial solution. It provides the best solution for 52% of the problem instances solved and it is within 0.5% of the best solution found for 90% of the problems solved. It gives excellent results for large problems (P_3) and for instances with smaller ship-to points (SN). It is also the fastest strategy. Strategy 3 finds the best design for only 27% of the problem instances. It gives better results for small networks with smaller ship-to points (SN). Due to the intensification phase added in Strategy 3, the number of shift moves grows rapidly. For large problems, the solution time is significantly larger than for the two other strategies.

When the network density is low, the number of borderline ship-to points is high and the **Reassign** procedure improves the design obtained significantly. When fixed depot costs are low, the design obtained includes more depots to save on transportation costs. Conversely, when ship-to points are smaller, the design obtained includes fewer depots. The cost structure does not have a marked impact on the performance of the search strategies. For large problems (P_3), however, S1 performs better for instances with large fixed and variable costs (a), and S2 performs better for problems with extreme fixed to variable costs ratios (b, c).

Our results also show that the route length estimation formula (27) is very accurate. Using it to evaluate trial moves provides very good designs. It gives values between [99%, 104%] of the exact transportation costs, computed with the user model, in the case of larger ship-to point networks (LN) and between [98%, 108%] in the case of smaller ship-to point networks (SN).

Conclusions

This paper defines and formulates an important strategic planning problem for distribution businesses using external transportation resources: the *stochastic multiperiod location-transportation problem* (SMLTP). The problem is characterized as a hierarchical decision problem involving a design level taking network location and allocation decisions and a user level taking transportation decisions, and it is formulated as a two-stage stochastic program with recourse. We show how a sample of multiperiod demand scenarios can be generated from the stochastic demand processes of customers and used to solve the problem. Because the resulting sample average approximation MIP is extremely large, it cannot be solved to optimality with commercial solvers for realistic problems and a hierarchical heuristic solution approach is proposed to solve it. It is based on a user-level transportation heuristic and a design-level location-allocation heuristic. Three different strategies based on drop, add, and shift moves were proposed to explore the neighbourhood of a solution. A revised route length approximation formula is also proposed to speed up calculations.

To test the quality of the heuristic developed, several industrial cases were examined and used to construct 48 realistic test-problem instances. The experiments made showed that the solution approach proposed provides good results in terms of solution quality and solution time. We found that neighbourhood search strategies S1 and S2 give the best results. The excellent performance of the nested solution approach proposed is principally due to the good quality of the anticipation of transportation costs in the design heuristic and to the efficient combination of several procedures and heuristics. We believe that the approach as it stands is sufficiently evolved to solve most practical cases efficiently and effectively. Some room is left, however, for additional research. Following are some future research avenues that could yield significant payoffs:

- We considered a context where a ship-to point must be allocated to the same depot for the entire planning horizon. In other contexts, a flexible allocation of ship-to points to depots may be possible. The user problem would then become a multidepot transportation problem and would be more difficult to solve.
- We assumed that the demand process is stationary. If it were nonstationary, the size n of the scenario samples to use to obtain good statistical optimality gaps would be much larger and the SAA model would be more difficult to solve.
- We were able to solve the SAA model to optimality with CPLEX-11 only for small problems (P_1), and we did not develop a specialized decomposition method to try to solve larger problems to optimality. This may be very difficult because, as indicated earlier, the number of binary variables for route selection grows exponentially, but it deserves further investigation.
- The three search strategies examined could be refined and fine tuned and other search methods, such as variable neighbourhood search, could be used to elaborate alternative heuristics.

Acknowledgments

This research was supported in part by Natural Sciences and Engineering Research Council of Canada Grant DNDPJ 335078-05; by Defence R&D Canada; and by Modellium Inc.

References

Albareda-Sambola, M., E. Fernandez, G. Laporte. 2007. Heuristic and lower bound for a stochastic location-routing problem. *Eur. J. Oper. Res.* **179** 940–955.

Ambrosino, D., M. G. Scutella. 2005. Distribution network design: New problems and related models. *Eur. J. Oper. Res.* **165** 610–624.

Ballou, R. H. 1992. *Business Logistics Management*, 3rd ed. Prentice Hall, Englewood Cliffs, NJ.

Barreto, S., C. Ferreira, J. Paixao, B. S. Santos. 2007. Using clustering analysis in a capacitated location-routing problem. *Eur. J. Oper. Res.* **179** 968–977.

Berger, R. T., C. R. Coullard, M. S. Daskin. 2007. Location-routing problems with distance constraints. *Transportation Sci.* **41**(1) 29–43.

Berman, O., D. Krass, C. W. Xu. 1995. Location discretionary service facilities based on probabilistic customer flows. *Transportation Sci.* **29**(3) 276–290.

Birge, J. R., F. Louveaux. 1997. *Introduction to Stochastic Programming*. Springer, New York.

Chien, T. W. 1993. Heuristic procedures for practical-sized uncapacitated location-capacitated routing problems. *Decision Sci.* **24**(5) 995–1021.

Clarke, G., J. W. Wright. 1964. Scheduling of vehicles from a central depot to a number of delivery points. *Oper. Res.* **12** 568–581.

Daganzo, C. F. 1984. The distance traveled to visit N points with a maximum of C stops per vehicle: An analytic model and an application. *Transportation Sci.* **18**(4) 331–350.

Girard, S., J. Renaud, F. F. Boctor. 2006. Heuristiques rapides pour le problème de tournées de véhicules. *Administrative Sci. Assoc. Canada 2006 Proc.*, Banff, Alberta, Canada.

Glover, F., M. Laguna. 1997. *Tabu Search*. Kluwer Academic Publishers, Boston.

Klose, A., A. Drexl. 2005. Facility location models for distribution system design. *Eur. J. Oper. Res.* **162** 4–29.

Kouvelis, P., G. Yu. 1997. *Robust Discrete Optimization and Its Applications*. Kluwer Academic Publishers, Dordrecht, The Netherlands.

Kuehn, A. A., M. J. Hamburger. 1963. A heuristic program for locating warehouses. *Management Sci.* **9** 643–666.

Laporte, G. 1988. Location-routing problems. B. L. Golden, A. A. Assad, eds. *Vehicle Routing: Methods and Studies*. North-Holland, Amsterdam, 163–198.

Laporte, G., P. J. Dejax. 1989. Dynamic location routing problems. *J. Oper. Res. Soc.* **40**(5) 471–482.

Laporte, G., F. V. Louveaux. 1998. Solving stochastic routing problems. T. G. Crainic, G. Laporte, eds. *Fleet Management and Logistics*. Kluwer Academic Publishers, Boston.

Laporte, G., I. H. Osman. 1995. Routing problems: A bibliography. *Ann. Oper. Res.* **61** 227–262.

Laporte, G., F. V. Louveaux, H. Mercure. 1989. Models and exact solutions for a class of stochastic location-routing problems. *Eur. J. Oper. Res.* **39** 71–78.

Laporte, G., M. Gendreau, J.-Y. Potvin, F. Semet. 2000. Classical and modern heuristics for the vehicle routing problem. *Internat. Trans. Oper. Res.* **7** 285–300.

Min, H., V. Jayaraman, R. Srivastava. 1998. Combined location-routing problems: A synthesis and future research directions. *Eur. J. Oper. Res.* **108** 1–15.

Nagy, G., S. Salhi. 1996a. Nested heuristic methods for the location-routing problem. *J. Oper. Res. Soc.* **47**(9) 1166–1174.

Nagy, G., S. Salhi. 1996b. A nested location-routing heuristic using route length estimation. *Stud. Locational Anal.* **10** 109–127.

Nagy, G., S. Salhi. 2007. Location-routing: Issues, models and methods. *Eur. J. Oper. Res.* **177** 649–672.

Owen, S. H., M. S. Daskin. 1998. Strategic facility location: A review. *Eur. J. Oper. Res.* **111** 423–447.

Prins, C., P. Prodhon, A. Ruiz, P. Soriano, R. W. Calvo. 2007. Solving the capacitated location-routing problem by a cooperative Lagrangian relaxation-granular tabu search heuristic. *Transportation Sci.* **41**(4) 470–483.

Rei, W., M. Gendreau, P. Soriano. 2007. A hybrid Monte Carlo local branching algorithm for the single vehicle routing problem with stochastic demands. CIRRELT Research Document 2007–24, CIRRELT, Montreal.

Ruszczynski, A., A. Shapiro, eds. 2003. *Stochastic Programming*, Vol. 10. Handbooks in Operations Research and Management Science, Elsevier, Amsterdam.

Salhi, S., G. Nagy. 1999. Consistency and robustness in location-routing. *Stud. Locational Anal.* **13** 3–19.

Salhi, S., G. Nagy. 2009. Local improvement in planar facility location using vehicle routing. *Ann. Oper. Res.* **167**(1) 287–296.

Santoso, T., S. Ahmed, M. Goetschalckx, A. Shapiro. 2005. A stochastic programming approach for supply chain network design under uncertainty. *Eur. J. Oper. Res.* **167** 96–115.

Sariklis, D., S. Powell. 2000. A heuristic method for the open vehicle routing problem. *J. Oper. Res. Soc.* **51** 564–573.

Shapiro, A. 2003. Monte Carlo sampling methods. A. Ruszczynski, A. Shapiro, eds. *Stochastic Programming*, Vol. 10, *Handbooks in Operations Research and Management Science*. Elsevier, Amsterdam.

Shen, Z.-J. 2007. Integrated supply chain design models: A survey and future research directions. *J. Indust. Management Optim.* **3**(1) 1–27.

Snyder, L. V., M. S. Daskin. 2005. Reliability models for facility location: The expected failure cost case. *Transportation Sci.* **39** 400–416.

Talbi, E. G. 2002. A taxonomy of hybrid metaheuristics. *J. Heuristics* **8** 541–564.

Toth, P., D. Vigo. 1998. Exact solution of the vehicle routing problem. T. G. Crainic, G. Laporte, eds. *Fleet Management and Logistics*. Springer, Berlin.

Tuzun, D., L. I. Burke. 1999. A two-phase tabu search approach to the location routing problem. *Eur. J. Oper. Res.* **116** 87–99.

Verweij, B., S. Ahmed, A. J. Kleywegt, G. Nemhauser, A. Shapiro. 2003. The sample average approximation method applied to stochastic routing problems: A computational study. *Computational Optim. Appl.* **24** 289–333.

Vila, D., A. Martel, R. Beauregard. 2007. Taking market forces into account in the design of production-distribution networks: A positioning by anticipation approach. *J. Indust. Management Optim.* **3**(1) 29–50.

Zainuddin, Z. M., S. Salhi. 2007. A perturbation-based heuristic for the capacitated multisource Weber problem. *Eur. J. Oper. Res.* **179** 1194–1207.