The Stochastic Multi-Period Location-Transportation Problem

Walid Klibi
Francis Lasalle
Alain Martel
Soumia Ichoua

June 2008

CIRRELT-2008-19
The Stochastic Multi-Period Location-Transportation Problem

Walid Klibi\textsuperscript{1,2}, Francis Lasalle\textsuperscript{1,2}, Alain Martel\textsuperscript{1,2,*}, Soumia Ichoua\textsuperscript{1,3}

\textsuperscript{1} Interuniversity Research Centre on Enterprise Networks, Logistics and Transportation (CIRRELT)
\textsuperscript{2} Département des opérations et systèmes de décision, Pavillon Palasis-Prince, 2325, rue de la Terrasse, Université Laval, Québec, Canada G1V 0A6
\textsuperscript{3} Department of Computer Science and Engineering, Johnson C. Smith University, Charlotte, NC 28216, USA

Abstract. This paper studies a Stochastic Multi-Period Location-Transportation Problem (SMLTP) characterized by multiple transportation options, multiple demand periods and a stochastic stationary demand. We consider the determination of the number and location of the depots required to satisfy a set of customer’s demands, and 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 Clark and Wright procedure. Three neighbourhood exploration strategies are proposed and compared with extensive experiments based on realistic problems.

Keywords. Location problem, transportation problem, stochastic customer order process, stochastic programming, Monte Carlo scenarios, tabu search.

Acknowledgements. This research was supported in part by Natural Sciences and Engineering Research Council of Canada (NSERC) grant no DNDPJ 335078-05 and by Modellium Inc.

Results and views expressed in this publication are the sole responsibility of the authors and do not necessarily reflect those of CIRRELT.

Les résultats et opinions contenus dans cette publication ne reflètent pas nécessairement la position du CIRRELT et n’engagent pas sa responsabilité.

* Corresponding author: Alain.Martel@cirrelt.ca

Dépôt légal – Bibliothèque nationale du Québec,
Bibliothèque nationale du Canada, 2008

© Copyright Klibi, Lasalle, Martel, Ichoua and CIRRELT, 2008
Introduction

Depots 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 a set of customer’s demands, and 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 et al., 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 multi-drop truckload routes (MTL), or via less-than-truckload (LTL) transportation, and the profits of the distribution network depends heavily on the efficiency of the transportation decisions made. The vehicle routing problems (VRP) encountered for the MTL case were studied extensively in the literature (see Laporte et al. (1995) for a review). In spite of the exact solution methods developed for deterministic routing problems (Toth and Vigo, 1998), its complexity lead most researchers in this field to develop 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 models integrating location, routing and inventory decisions. 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; Tuzin et al., 1999; Barreto, 2007; Prins et al., 2007) present heuristic methods to solve deterministic LRPs. A multi-echelon version with inventory is addressed in Ambrosino and Scutella (2005). An uncapacitated LRP with distance constraints is studied in Berger et al. (2007); they propose a set partitioning formulation and a branch and price solution approach. The hierarchical
structure of the problem is stressed in Salhi and Nagy (1996a, b) who develop a 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). A stochastic LRP is studied in Laporte and Louveaux (1989) and in Albareda-Sambola et al. (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 LRP’s is also presented in Salhi and Nagy (2007). Comprehensive reviews of location-routing models and of their applications are provided in Min et al. (1998) and Nagy and Salhi (2007).

The LRP models found in the literature have three 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 pure VRP problems. When common or contract-carriers are used, more options are available, namely: single-customer full (FTL) or partial (STL) truckload, multi-drop truckload (MTL) and/or less-than-truckload (LTL) transportation. This leads to Location-Transportation Problems (LTP) instead of location-routing problems.

- Second, most LRP models implicitly assume that the location decisions and the 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 the depots must be fixed for the planning horizon considered, but transportation decisions must be made on a daily basis in reaction to the customer orders received. This gives rise to what we call Multi-period Location-Transportation Problems (MLTP). Salhi and Nagy (1999) define a similar multi-period location-routing problem but they address it only implicitly.

- Finally, most LRP models do not consider the customer order arrival process explicitly. In our context, a random number of customers order a random amount of products on a daily basis, and these orders must be delivered on the next day. This gives rise to what we call Stochastic Multi-period Location-Transportation Problems (SMLTP).

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 a heuristic method to solve it.

The SMLTP is a NP-hard stochastic combinatorial optimization problem as it reduces to the open LRP (Sariklis and Powell, 2000) when a single time period is considered, when customer demands are known for that period and are less than a full truckload, and when no LTL transportation is available. A proof that the LRP is a NP-hard problem is provided in Laporte (1988).
Hence, while 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 it formulates the problem as a stochastic program with recourse. The following section shows how this stochastic program can be solved with a sample average approximation (SAA) 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, and computational results are 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 it must be shipped to a large number of ship-to-points. In order to provide a good service, the company cannot satisfy customer orders directly from the supply sources and it must operate a number of uncapacitated distribution centers (DCs, also referred to as depots). The 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 DC, when all the orders are in, the company plans its transportation for the next day, and it requests from its carriers the trucks required to deliver products to ship-to-points through full truckloads (FTL), single customer partial truckloads (STL), multi-drop TL routes (MTL) or LTL transportation. 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 which 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 which 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-points demands, service requirements and vehicle characteristics; and let \( P_k \subset P \) be the ordered set of ship-to-points in route \( k \in K_{lt}^{TL} \). The TL-tariff \( w_k \) charged by the carriers to assign a specific truck to route \( k \) is based on the following formula:

\[
w_k = \max(r_i m_k; TL_r) + a_i (|P_k| - 1)
\]
where

\[
m_k = \text{total mileage of route } k
\]
\[
r_k = \text{transportation cost rate per mile for the vehicle associated to route } k
\]
\[
TL_l = \text{minimum transportation charge for any TL route starting at depot } l
\]
\[
a_l = \text{drop charge for any additional stop on a route starting at depot } l
\]

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 \(k \in K^\text{TL}_{l}\) 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 \(K^\text{TL}_{l}\) contain only non-dominated TL routes.

If the vehicle on road \(k\) has a load close to its capacity, then multi-drop 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/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 \(k \in K^\text{TL}_{l}\) would not be used if \(w_k > \sum_{p \in P_k} LTL(l, p; d_{pt})\), where \(d_{pt}\) is the load to be dropped at ship-to-point \(p \in P_k\) on day \(t\), and \(LTL(l, p; d)\) 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 \(a \text{ priori}\). We assume, from now on, that all the routes in the set \(K^\text{TL}_{l}\) are also non-dominated by their alternative LTL routes. Conversely, on a day \(t\), a 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 \(LTL(l, p; d_{pt}) < \max(r_{(l,p)}, m_{(l,p)}; TL_l)\). This implies that, in the daily transportation planning process at depot \(l\), in addition to the non-dominated feasible TL-routes \(k \in K^\text{TL}_{l}\), all the economic depot to ship-to-point LTL routes \(k \in K^\text{LTL}_{l}\) must be considered, i.e. the set of routes to consider is \(K_{l} = K^\text{TL}_{l} \cup K^\text{LTL}_{l}\). The LTL-tariff paid for a single destination route \(k \in K^\text{LTL}_{l}\) is given by \(w_k = LTL(l, p_k; d_{p_k})\), 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 \(L \subset L\) to operate during the planning horizon \(T\) considered and the assignment of ship-to-points \(P^* \subset P\), \(l \in L^*\), to these depots, in order to maximize total expected profits. Note that, in what follows, the notation \(L\) is used to represent a subset of open depots in \(L\), and \(\overline{L} = L \setminus L\) its complement. Similarly, \(P \subset P\) is used to represent the subset of ship-to-points assigned to depot \(l \in L\). An important aspect of the problem is that the mission of the selected depots, defined by their customer sets, \(P^*, l \in L^*\).
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 products shipped from that depot is \( v_i \). The value \( v_i \) takes into account the product production/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 multi-period location-transportation network examined is illustrated in Figure 1.

### The Distribution Network User Problem

As previously explained, for a given distribution network with depots set \( L \subset L \) and mission \( P, l \in L \), on a daily basis, the depots \( l \in L \) receive orders from their customers \( p \in P \) 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 inter-arrival time \( q_p \) and a random order size \( o_p \). The cumulative distribution functions of inter-arrival times and order sizes are denoted respectively by \( F_p^q(.) \) and \( F_p^o(.) \). A possible realization of these compound stochastic processes over planning horizon \( T \) is illustrated in Figure 2, for exponential inter-arrival times and Normal order sizes. Such realizations constitute demand scenarios and the set of all demand scenarios associated to the compound demand processes considered is denoted by \( \Omega \). 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 \( l \in L \) are defined by the loads \( d_{pl}(\omega), p \in P_t(\omega) \), where \( P_t(\omega) = P_l \cap P_t(\omega) \) is the set of depot \( l \) ship-to-points which order products on day \( t \).
Given the loads \( d_{pt}(\omega), p \in P_t(\omega), l \in L \), 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 \( K_{plt}^{FTL}(\omega) \) 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:

\[
\bar{d}_{pt}(\omega) = d_{pt}(\omega) - \sum_{k \in K_{plt}^{FTL}(\omega)} b_k y_{klt}^{FTL}(\omega), \quad p \in P_t(\omega)
\]

where \( y_{klt}^{FTL}(\omega) \) 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.

\[\text{Figure 2- Ship-to-Points Stochastic Demand Process}\]

Next, the best delivery routes must be constructed. Let,

\( K_{lt}(\omega) \): Set of non-dominated feasible delivery routes (i.e. such that \( P_k \subset P_{lt}(\omega) \) and \( \sum_{p \in P_k} \bar{d}_{pt}(\omega) \leq b_k, \quad k \in K_{lt}(\omega) \)) from depot \( l \), on day \( t \), under scenario \( \omega \);

\( \delta_{kp} \) : Binary coefficient taking the value 1 if ship-to point \( p \) is covered by route \( k \) (i.e. if \( p \in P_k \)), and 0 otherwise;

\( y_{klt}(\omega) \) : Binary decision variable equal to 1 if route \( k \) is used from depot \( l \) on day \( t \) under scenario \( \omega \), and to 0 otherwise.

For demand scenario \( \omega \), the best routes are obtained at depot \( l \) on day \( t \) by solving the following transportation sub-problem:
subject to

\[
\sum_{k \in K_k(\omega)} \delta_{kp} y_{ktl}(\omega) = 1 \quad p \in P_l(\omega) \\
y_{ktl}(\omega) \in \{0,1\} \quad k \in K_k(\omega)
\]

where \( y \) denotes the vector of all the routing decisions, and \( C_{lt}(\omega) \) is the cost of the optimal shipments made by depot \( l \) on day \( t \) under scenario \( \omega \). 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 depots production/procurement, warehousing, inventory holding and customer shipment costs, the net revenues \( R^u(\omega) \) generated at the distribution network user level for demand scenario \( \omega \) are given by:

\[
R^u(\omega) = \sum_{t \in T} \sum_{l \in L} \left[ \sum_{p \in P_l(\omega)} \left( u_p - v_l \right) c_{ptl}(\omega) - \sum_{p \in P_l(\omega)} \sum_{k \in K_k(\omega)} w_k y_{ktl}^{FTL}(\omega) - C_{lt}(\omega) \right]
\]  

These net revenues are an important element to take into account in the distribution network design model.

**The Distribution Network Design Problem**

The SMLTP is as 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 \( L' \subset L \) to use during the planning horizon \( T \) considered, and the mission \( P'_l, l \in L' \), 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 taking daily transportation decisions and, conversely, adequate network design decisions can’t 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. Since the ship-to-points demands for the horizon \( T \) are not known when the design decisions are made, this information take the form of the set of potential demand scenarios \( \Omega \) 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 loca-
tion and mission decisions, and the second stage with daily transportation decisions. The follow-

The following first stage decision variables are required to formulate the model:

\[ x_l : \text{ Binary variable equal to 1 if DC } l \text{ is opened, and to 0 otherwise; } \]

\[ x_{lp} : \text{ Binary variable equal to 1 if ship-to point } p \text{ is assigned to DC } l, \text{ and to 0 otherwise; } \]

and the notation \( \mathbf{x} \) is used to denote the vector of all these decision variables, and \( \mathbf{x}_l \) the vector of depot \( l \) ship-to-point assignment variables. Note that a given binary vector \( \mathbf{x} \) specifies unique design sets \( L \) and \( P_l \), namely that: \( L = \{ i | x_i = 1 \} \), \( \bar{L} = \{ i | x_i = 0 \} \) and \( P_l = \{ p | x_{lp} = 1 \} \), \( l \in L \).

The stochastic programming model to solve is the following:

\[
R = \max_{\mathbf{x}} \sum_{\omega \in \Omega} \pi(\omega) R^{du}_{\mathbf{x}}(\mathbf{x}, \omega) - \sum_{l \in L} A_l x_l
\]

subject to

\[
\sum_{l \in L_p} x_{lp} = 1 \quad \text{ for } p \in P
\]

\[
x_{lp} \leq x_l \quad l \in L, \ p \in P_l
\]

\[
x_l, x_{lp} \in \{0, 1\} \quad l \in L, \ p \in P_l
\]

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

\[
R^{du}_{\mathbf{x}}(\mathbf{x}, \omega) = \sum_{l \in L} \sum_{i \in I_l} \left\{ \sum_{\omega \in \Omega} \left[ (u_{pi} - v_i)d_{pi}(\omega) - \sum_{k \in K_{pi}^l(\omega)} w_{kp} y^TTL_{kp}(\omega) \right] x_{lp} - C^{du}_{il}(\mathbf{x}, \omega) \right\}
\]

with

\[
C^{du}_{il}(\mathbf{x}, \omega) = \min_{y} \sum_{k \in K_{il}(\omega)} w_{kp} y_{kl}(\omega)
\]

subject to

\[
\sum_{k \in K_{il}(\omega)} \delta_{lp} y_{kl}(\omega) = x_{lp} \quad p \in P_l(\omega)
\]

\[
y_{kl}(\omega) \in \{0, 1\} \quad k \in K_{il}(\omega)
\]

In the first term of objective function (6), expected net revenues are calculated and, in the second term, depots 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 $\Omega$ is usually extremely large. In fact, when the inter-arrival times and order sizes distribution functions $F^u_p(.)$ and $F^o_p(.)$ are continuous, there is an infinite number of possible demand scenarios. This is the case, for example, when inter-arrival 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 et al. (2007), and to supply chain network design problems in Santoso et al. (2005) and Vila et al. (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 first to generate an independent sample of $n$ equiprobable scenarios $\{\omega^1, \ldots, \omega^n\} = \Omega^s \subset \Omega$ from the initial probability distributions of order inter-arrival 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:

$$\bar{R}_n = \max_{x,y} \frac{1}{n} \sum_{\omega \in \Omega^s} \sum_{t \in T} \sum_{l \in L} \left[ \sum_{p \in F_l(\omega)} (u_p - v_j) d_{pt}(\omega) - \sum_{k \in K_p(\omega)} w_k y_{kl}(\omega) \right] x_{lp} - \sum_{k \in K_p(\omega)} w_k y_{kl}(\omega) - \sum_{l \in L} A_l x_l$$

subject to

$$\sum_{l \in L} x_{lp} = 1 \quad p \in P$$

$$x_{lp} \leq x_l \quad l \in L, \; p \in P_i$$

$$\sum_{k \in K_p(\omega)} \delta_{kp} y_{kl}(\omega) = x_{lp} \quad \omega \in \Omega^s, \; l \in L, \; t \in T, \; p \in P_i(\omega)$$

$$x_l, y_{kl}(\omega) \in \{0,1\} \quad \omega \in \Omega^s, \; l \in L, \; p \in P, \; t \in T, \; k \in K_p(\omega)$$

The scenarios in sample $\Omega^s \subset \Omega$ used in the model are generated directly from the cumulative distribution functions of inter-arrival times and order sizes $F^u_p(.)$ and $F^o_p(.)$, $p \in P$. 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, $F^{-1}_p(u_q)$ and $F^{-1}_o(u_o)$, of the distributions of inter-arrival 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 $\omega$ is presented in Figure 3. In this procedure, the continuous variable $\tau$ is used to denote order-arrival times. Order arrivals are gener-
ated in the interval \([0,T]\) and mapped onto the corresponding planning periods \(t \in T\). Note that 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 \(\Omega^*\).

\[
\text{MonteCarlo}\left((F_p^{\alpha}(\cdot),F_p^{\alpha}(\cdot),p \in P),T; d_{pt}(\omega), p \in P, t \in T\right)
\]

For all \(p \in P\), do:
\[
\tau = 0; \quad d_{pt}(\omega) = 0, \quad t \in T
\]

While \(\tau \leq |T|\), do:
\[
\quad\text{Generate the Uniform [0,1] random numbers } u_q \text{ and } u_o
\]
\[
\quad\text{Compute the next order arrival time } \tau = \tau + F_p^{q^{-1}}(u_q) \text{ and } t = \left\lfloor \tau \right\rfloor
\]
\[
\quad\text{Compute the planning period } t \text{ demand } d_{pt}(\omega) = d_{pt}(\omega) + F_p^{\alpha^{-1}}(u_o)
\]

End While

End Do

**Figure 3- Procedure MonteCarlo for the Generation of Scenario \(\omega\)**

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 *et al.*, 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. Pre-established set of non-dominated 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 \(|T|^*|\Omega^*|\) is large, however, this optimal approach can be used only for relatively small problems. Our aim in the next section is to propose a heuristic method to solve realistic problem instances.

**Solution approach**

**The General Scheme**

The SMLTP is as 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 sub-problems. 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 bi-level 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; it 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 savings heuristic is proposed. This user heuristics 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 scenarios sample $\Omega^n$ 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**

Let’s consider the user problem for a given distribution network with depots set $L \subset L$ and mission $P_l$, $l \in L$. At the user level, under scenario $\omega$, for all the ship-to-point orders $P_{pl}(\omega)$ received by depot $l$ on day $t$, the objective of depot $l$ is to select the FTL, STL, MTL and LTL shipments minimizing its transportation costs. As explained previously, this is achieved with a two steps procedure. The first step determines the number of full truckload shipments to make and residual ship-to-point loads. The number of full trucks of each type $k \in K_{pl}^{FTL}(\omega)$ to ship on day $t$ from depot $l$ is found by solving the following simple integer programs by inspection:

$$y_{lkt}(\omega) = \arg \max \left\{ \sum_{k \in K_{pl}^{FTL}(\omega)} b_k y_k \right\} \text{subject to} \sum_{k \in K_{pl}^{FTL}(\omega)} b_k y_k \leq d_{pt}(\omega), y_k = 0,1, ... , p \in P_{pl}(\omega)$$

Then the residual loads $\vec{d}_{pt}(\omega), p \in P_{pl}(\omega)$, 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) savings and 2-opt improvements, developed by Girard *et al.*, (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 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:

$$w_{l(p)} = \min \left[ LTL(l, p; \vec{d}_{pt}(\omega)); \max (r_{l(p)}, m_{l(p)}; TL_l) \right], p \in P_{pl}(\omega)$$

**The Stochastic Multi-Period Location-Transportation Problem**
Then, as illustrated in Figure 4, for two ship-to-points $p$ and $p'$, the savings associated to 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:

$$e_{pp'} = w_{(l,p)} + w_{(l,p')} - w_{(l,p,p')}, \quad p \in P_{lt}(\omega), \quad p' \in P_{lt}(\omega) \setminus \{p\}$$

where $w_{(l,p,p')} = \max(\ell_{(l,p,p')} m_{(l,p,p')}, TL_l) + a_i$. The perturbed CW heuristic works as the original CW algorithm but uses the following modified savings formula:

$$e_{pp'} = w_{(l,p)} + w_{(l,p')} - \lambda_{pp'} w_{(l,p,p')}, \quad p \in P_{ln}(\omega), \quad p' \in P_{ln}(\omega) \setminus \{p\}$$

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

$$\hat{R}_{lu}^{du} \left( P_{lu}(\omega) \right) = \sum_{p \in P_{ln}(\omega)} \left[ (u_p - v_i) d_{pt}(\omega) - \sum_{k \in K_{pt}^{ln}(\omega)} w_k y_{lt}^{TL}(\omega) \right] - \hat{C}_{lu}^{du} \left( P_{lu}(\omega) \right)$$

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

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 $x$ under any given scenario $\omega \in \Omega$. The total net revenues generated by design $x$, for the scenario $\omega$ considered, is obtained simply by summing net revenues over all depots and days, i.e. by calculating $\hat{R}_u(x,\omega) = \sum_{l \in L} \sum_{t \in T} \hat{R}_{lu}^{du} \left( P_{lu}(\omega) \right)$. When a sample $\Omega^*$ of $n$ Monte Carlo scenarios is used, an estimate $\hat{R}_u(x)$ of the expected value of the design considered is thus given by:

$$\hat{R}_u(x) = \frac{1}{n} \sum_{\omega \in \Omega^*} \sum_{l \in L} \sum_{t \in T} \hat{R}_{lu}^{du} \left( P_{lu}(\omega) \right) - \sum_{l \in L} A_l$$

Note that all the **Procedures** presented in the paper use the following syntax:

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

CIRRELT-2008-19 12
User \( \left( b_k \neq k \in K_{pt}^{FTL}(\omega) \right) \), \( d_{pt}(\omega) \), \( p \in P \Theta(\omega) \), \( \gamma \), \( \lambda \), \( \lambda^+ \); \( \hat{R}_U \left( P_X(\omega) \right) \)

S1:  

a) Solve (19) by inspection to obtain the FTL shipments 
\( y_{pt}^{FTL}(\omega) \), \( k \in K_{pt}^{FTL}(\omega) \), \( p \in P \Theta(\omega) \)  

b) Compute the residual loads \( d_{pt}(\omega) \), \( p \in P \Theta(\omega) \), with (1)

S2:  

a) Select the best direct delivery transportation modes (LTL vs STL), and compute their costs \( w_{i,p} \), \( p \in P \Theta(\omega) \), with (20)  

b) Solve the resulting open routing problem \( \gamma \) times with the modified Clark and Wright algorithm, using the savings \( e_{pp}^r \), \( p \in P \Theta(\omega) \), \( p' \in P \Theta(\omega) \) \( \setminus \{ p \} \) computed with (22), and retain the best transportation solution.

S3:  

Compute the net revenue \( \hat{R}_U \left( P_X(\omega) \right) \) with (23)

**Figure 5- Procedure User for Depot / in Period \( t \) under Scenario \( \omega \)**

**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. Since non-improving moves are allowed, solutions previously encountered during the search may be re-visited. To avoid this cycling phenomenon, a short term memory, called 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 \( x^c \) available is improved. Potential, non-Tabu, solutions in the neighborhood of \( x^c \) 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 \( x^r \) found to date. The following paragraphs describe the main features of the Tabu search heuristic proposed, as well as three alternative neighbourhood exploration strategies.

**Neighborhood Structure**

At each iteration, a new design \( x^m \) is generated from the current design \( x^c \) using one of the following three moves: (i) Drop: close an opened distribution center and re-assign its ship-to-points; (ii) Add: open a closed distribution center and assign some ship-to-points to it; (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 \( x^c \) at an iteration of the algorithm. Then, using the operator \( \oplus \) to denote a move, the **General Neighbourhood** of design
The feasible solution set $GN(x^c) = \{x^m | x^m = x^c \oplus m, m \in M^c\}$. Unfortunately, the size of $GN(x^c)$ 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 $GN(x^c)$. 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 $N(l)$ be the set of neighbours of depot $l$. Then, for each open depot $l \in L^c$ specified by the current design $x^c$, we define a region $R(l)$ including depot $l$, the depots in $N(l)$, and the ship-to points $P^r, l' \in [l] \cup (N(l) \cap L^c)$, 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 $RN(x^c) \subset GN(x^c)$ defined by limited move subsets $M(l) \subset M^c, l \in L^c$. For a given region $R(l)$, the feasible moves considered are:

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

ii. **Add moves** ($M_{\text{add}}(l)$): If $N(l) \cap \overline{L}^c \neq \emptyset$, for all $l'' \in N(l) \cap \overline{L}^c$, open depot $l''$ and assign each ship-to points $p \in R(l) \cap P$ to depot $l_p = \arg \max_{l'' \in R(l) \cap L^c \cup \{l''\}} (u_p - v_{l''}) - w_{l''(p)}$.

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

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

$$M_{\text{drop}} = \{m_{\text{drop}}(l)\}_{l \in L^c}, M_{\text{add}} = \bigcup_{l \in L^c} M_{\text{add}}(l), M_{\text{shift}} = \bigcup_{l \in L^c} M_{\text{shift}}(l)$$

$$M = M_{\text{drop}} \cup M_{\text{add}} \cup M_{\text{shift}}, M(l) = \{m_{\text{drop}}(l)\} \cup M_{\text{add}}(l) \cup M_{\text{shift}}(l)$$

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 DCs to permit next day delivery, then it is not considered. Note also that during the neighbourhood exploration the moves in $M$ are evaluated only if they are not Tabu.
Neighborhood Evaluation

The evaluation of a design \( x^m \in RN(x^c) \) involves the estimation of its expected value for the sample \( \Omega^n \) of Monte Carlo scenarios generated. This could be done by applying the User heuristic to all \((i, \omega, t) \in L^n \times \Omega^n \times T\), and then by computing \( R_n(x^m) \) 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 \( \tilde{C}_{li}^du \left( P^m_{li} (\omega) \right) \) of depot \( l \) on day \( t \) under scenario \( \omega \), as a function of the set of ship-to-points \( P^m_{li} (\omega) \) visited from depot \( l \) on that day under design \( x^m \). The function is obtained with a multiple regression model based on three explanatory variables: i) a linehaul variable \( \left( \xi_i (P^m_{li} (\omega)) / NS_i \right) \), where \( \xi_i (P) \) is the sum of the distances from depot \( l \) to all \( p \in P \), and \( NS \), the average number of stops on the routes from depot \( l \); ii) a detour variable \( \left( \xi_i (P^m_{li} (\omega)) / \sqrt{P^m_{li} (\omega)} \right) \); and iii) a LTL variable \( \left( \rho_i^{LTL} \varphi_i (P^m_{li} (\omega)) \right) \), where \( \varphi_i (P) \) is the sum of the load-distances from depot \( l \) to all \( p \in P \), and \( \rho_i^{LTL} \) the proportion of load-distances from depot \( l \) on LTL routes. This leads to the following cost approximation formula:

\[
\tilde{C}_{li}^du \left( P^m_{li} (\omega) \right) \approx \hat{\beta}_1 \left( \frac{\xi_i (P^m_{li} (\omega))}{NS_i} \right) + \hat{\beta}_2 \left( \frac{\xi_i (P^m_{li} (\omega))}{\sqrt{P^m_{li} (\omega)}} \right) + \hat{\beta}_3 \left( \rho_i^{LTL} \varphi_i (P^m_{li} (\omega)) \right) \tag{25}
\]

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 \( NS_i \) and \( \rho_i^{LTL} \) 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 \( R_n (x^m) \) is then obtained by replacing \( \tilde{C}_{li}^du \left( P^m_{li} (\omega) \right) \) in (23) by \( \tilde{C}_{li}^du \left( P^m_{li} (\omega) \right) \), and by substituting in (24), to get:

\[
\tilde{R}_{li} \left( P^m_{li} (\omega) \right) = \sum_{p \in P} \left[ u_{p-v_i} \right] d_{pt} (\omega) - \sum_{k \in K^LTL_{li}} w_{k} \varphi_{LTL} (\omega) \right] - \tilde{C}_{li}^du \left( P^m_{li} (\omega) \right) \tag{26}
\]

\[
\tilde{R}_n (x^m) = \frac{1}{n} \sum_{\omega \in \Omega^n} \sum_{l \in L^n} \sum_{t \in T} \tilde{R}_{li} \left( P^m_{li} (\omega) \right) - \sum_{l \in L^n} A_l \tag{27}
\]
Restricted Neighborhood Exploration

Given the restricted neighbourhood structure previously defined, three different strategies are considered to explore the neighbourhood \( RN(x^c) \) of the current solution \( x^c \). 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:** This strategy is inspired from the Tabu search approach proposed by Salhi and Nagy (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 \( M \) are considered to generate a new solution.

**Strategy 2:** 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 \( M_{\text{add}} \) only, the second phase considers drop moves \( M_{\text{drop}} \) exclusively, and the third phase concentrates on shift moves \( M_{\text{shift}} \). In our implementation, only non-Tabu shift moves are considered.

**Strategy 3:** This strategy starts with the initial solution obtained by assigning each ship-to-point to the feasible depot yielding the maximum marginal net revenue. Clearly, in this solution, all the potentially interesting depots are opened. Then, at each iteration, a drop move in \( M_{\text{drop}} \) is performed, followed by shift moves in \( M_{\text{shift}} \). This process continues until the specified maximum number of iteration 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 non-improving moves are allowed.

Note that the moves \( m(l) \in M \) 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 \( x^c \). To do this, we elaborated a reassignment procedure for borderline points, inspired from a heuristic proposed by Zainuddin and Salhi (2007) to solve the capacitated Weber problem. A point \( p \) assigned to depot \( l \in L^m \) (i.e. in \( P^m \)), is considered borderline if its distance from depot \( l \) exceeds a predefined target distance \( m_{\text{max}} \), i.e. if \( m_{(l,p)} > m_{\text{max}} \). Let \( B \) be the set of borderline ship-to-points. A ship-to-point \( p \) is reassigned from depot \( l \) to depot \( l_p \in L^m \) if its distance ratio \( \rho_p = m_{(l_p,p)}/m_{(l,p)} \) is the lowest among the eligible depots, provided that it does not exceed a predefined value \( \rho_{\text{max}} \). When this is done for all the borderline points \( p \in B \) asso-
associated to \(x^n\), a new solution vector \(x'\) results. Several alternative solution vectors \(x'\) can be generated by considering a set \(\Upsilon_{\text{max}}\) of \(\rho_{\text{max}}\) 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 in a drop/add list \(T_1\). On the other hand, when the new current solution is obtained by shifting two depots \(l\) and \(l'\), the pair \((l,l')\) is inserted in a shift list \(T_2\). In both cases, the oldest element of the list is removed. Note that when a depot \(l\) is inserted in \(T_1\), it becomes also Tabu until its Tabu status is revoked.

At each iteration, the length of both lists is randomly generated in the interval \([\alpha_1|L|/2, |L|/2]\) for \(T_1\) and \([\alpha_2|L|/2, |L|]\) for \(T_2\) where \(\alpha_1, \alpha_2 \in [0,1]\) are predefined parameters. Note that these intervals bound 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 first involves 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:

\[
MI = \text{maximum number of iterations},
\]
\[
MNI = \text{the maximum number of iterations without improvement}.
\]

Next, a sample of \(n\) demand scenarios \(d(\omega)=[d_{pl}(\omega)]_{p \in P, t \in T}, \omega \in \Omega^n\), must be generated using procedure MonteCarlo \(n\) times, and the neighbour sets \(N(l), l \in L\), must be created. Initial
solutions must then be constructed for the neighbourhood exploration strategy considered. For strategy 2), an initial solution $x^0$ is obtained by sequentially opening the closed depot $l \in L$ maximizing the marginal net revenues of the ship-to-points $P$ not yet allocated. For strategy 1) and 3), $L^0$ 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 $l_p \in L_p$ maximizing marginal net revenues. The resulting design $x_0$ is then evaluated with the User heuristic and the expected value function (24). Figure 7 presents the initialization procedure thus obtained.

| Initialize (Strategy; $n, \gamma, \lambda^-, \lambda^+, \alpha_1, \alpha_2, m_{\text{max}}, Y_{\text{max}}, Ml, MNI, (d(\omega), \omega \in \Omega^l), (N(l), l \in L), x^0, \bar{R}_n(x^0))$
| Set the heuristic parameters $(n, \gamma, \lambda^-, \lambda^+, \alpha_1, \alpha_2, m_{\text{max}}, Y_{\text{max}}, Ml, MNI)$
| For all $\omega \in \Omega^l$, do MonteCarlo $(F_{p}^s(\cdot), F_{p}^o(\cdot), p \in P), T; d_{pt}(\omega), p \in P, t \in T$
| Obtain the neighbour sets $N(l), l \in L$
| If Strategy = 2, then
| Set $L^0 = \emptyset$, $P = P$
| While $P \neq \emptyset$, do
| $l' = \arg \max_{l \in L^0} \sum_{p \in P \cap l} (u_p - v_l) - w_{(l,p)}$
| Set $L^0 = L^0 \cup \{l'\}$, $P_p = P \cap l'$ and $P = P \setminus P_p$
| End While
| Else Set $L^0 = L$
| Construct an initial design $x^0$ by assigning each ship-to-point $p \in P$ to the depot $l_p = \arg \max_{l \in L_p} (u_p - v_l) - w_{(l,p)}$
| For all $(l, \omega, t) \in L^0 \times \Omega^l \times T$, do
| User $(b_k, k \in R_{\text{pl}}^s(\omega)), d_{pt}(\omega), p \in P^o_n(\omega); \gamma, \lambda^-, \lambda^+; \bar{R}_n^{\text{pl}}(P^o_n(\omega))$
| Compute the expected value $\bar{R}_n(x^o)$ with (24)

**Figure 7- Initialization Procedure**

**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. Step S2 examines different moves in the neighbourhood of the current solution $x^c$. Step S3 applies the reassignment procedure to the current solution $x^c$ when $m(l) \in M_{\text{shift}}$ in strategy 3 and $m(l) \in M$ in strategies 1 and 2. Step S4 evaluates the best move and updates the parameters of the transportation costs estimation function. Step S5 manages the Tabu lists. Step S6 performs a shift-move intensification when the parameter Improve=1. Step S7 checks if the last iteration improved the best solution to date $x^*$. Finally, steps
S8 and S9 control the iterations of the algorithm. If either of the iteration limits is not reached, the exploration continues from S2 with the current solution $x^c$. Else, if $MNI$ iterations were made without improvement, the exploration continue from S1 with the best solution to date $x^*$. Otherwise, the algorithm stops. Note that, to simplify the exposition, we used the parameters $MI$ and $MNI$ when calling Tabu in S6 as in the main procedure. In the implementation, however, the iteration limits parameters used in step 6 are not the same as in the main procedure.

<table>
<thead>
<tr>
<th>Step</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>S0:</td>
<td>Set $x^* = x^o$ and $\tilde{R}_n(x^o) = \tilde{R}_n(x^*)$ and initialize the Tabu lists ($T_1$ and $T_2$)</td>
</tr>
<tr>
<td>S1:</td>
<td>Set $x^c = x^<em>$ and $\tilde{R}_n(x^c) = \tilde{R}_n(x^</em>)$</td>
</tr>
<tr>
<td>S2:</td>
<td>$\tilde{R}_n(x') = 0$ For all $l \in \mathbb{L}^c$, do Construct the region $R(l)$ For all $m(l) \in M(l)$ and $m(l)$ not Tabu, do Set $x^m = x^c \oplus m(l)$ and compute $\tilde{R}_n(x^m)$ with (27) If ($\tilde{R}_n(x^m) &gt; \tilde{R}_n(x')$), then $x' = x^m$ and $\tilde{R}_n(x') = \tilde{R}_n(x^m)$ End do End do $x^c = x'$</td>
</tr>
<tr>
<td>S3:</td>
<td>Reassign $\left(x^c; m_{\text{max}}, Y_{\text{max}}; x', \tilde{R}_n(x') \right)$ and set $x^c = x'$</td>
</tr>
<tr>
<td>S4:</td>
<td>For all $(l, \omega, t) \in \mathbb{L} \times \Omega \times T$, do User $\left(b_k, k \in K^{FTL}(\omega), d_{pr}(\omega), p \in \mathcal{P}_p(\omega); \gamma, \lambda^-, \lambda^+; \tilde{R}<em>n \left(P^c</em>{\text{d}}(\omega) \right) \right)$ Compute the expected value $\tilde{R}_n(x^c)$ with (24) Update the means $NS_l$ and $\rho_l^{FTL}$, $l \in \mathbb{L}$, using the augmented set of generated routes</td>
</tr>
<tr>
<td>S5:</td>
<td>Update the Tabu lists ($T_1$ and $T_2$)</td>
</tr>
<tr>
<td>S6:</td>
<td>If (Improve=1), then $\tilde{R}<em>n(x^c), M</em>{\text{shift}} ; MI, MNI, 0; x', \tilde{R}_n(x') \right)$ $x^c = x'$ and $\tilde{R}_n(x^c) = \tilde{R}_n(x')$</td>
</tr>
<tr>
<td>S7:</td>
<td>If ($\tilde{R}_n(x^c) &gt; \tilde{R}<em>n(x^<em>)$), then $x^</em> = x^c$, $\tilde{R}<em>n(x^*) = \tilde{R}<em>n(x^c)$ and $\text{iter}</em>{ni} = 0$ Else $\text{iter}</em>{ni} = \text{iter}</em>{ni} + 1$</td>
</tr>
<tr>
<td>S8:</td>
<td>$\text{iter} = \text{iter} + 1$</td>
</tr>
<tr>
<td>S9:</td>
<td>If ($\text{iter} &lt; MI$) and ($\text{iter}_{ni} &lt; MNI$), then go to S2 Else If ($\text{iter} &lt; MI$), then go to S1 Else stop</td>
</tr>
</tbody>
</table>

**Figure 8- Tabu Procedure**
Using the **Initialize** and the **Tabu** procedures described previously, the neighbourhood exploration strategies considered are implemented as follows:

**Strategy 1 ($S_1$):**

\[
\text{Initialize}(1; n, \gamma, \lambda^-, \lambda^+, \alpha_1, \alpha_2, m_{\text{max}}, Y_{\text{max}}, MI,MNI, (d(\omega), \omega \in \Omega^o),(N (l), l \in L), x^o, R_n(x^o))
\]

\[
\text{Tabu}(x^o, R_n(x^o), M; MI,MNI, 0; x^*, R_n(x^*))
\]

**Strategy 2 ($S_2$):**

\[
\text{Initialize}(2; n, \gamma, \lambda^-, \lambda^+, \alpha_1, \alpha_2, m_{\text{max}}, Y_{\text{max}}, MI,MNI, (d(\omega), \omega \in \Omega^o),(N (l), l \in L), x^o, R_n(x^o))
\]

\[
\text{Tabu}(x^o, R_n(x^o), M_{\text{add}}; MI,MNI, 0; x^{\text{add}}, R_n(x^{\text{add}}))
\]

\[
\text{Tabu}(x^{\text{add}}, R_n(x^{\text{add}}), M_{\text{drop}}; MI,MNI, 0; x^{\text{drop}}, R_n(x^{\text{drop}}))
\]

\[
\text{Tabu}(x^{\text{drop}}, R_n(x^{\text{drop}}), M_{\text{shift}}; MI,MNI, 0; x^*, R_n(x^*))
\]

**Strategy 3 ($S_3$):**

\[
\text{Initialize}(3; n, \gamma, \lambda^-, \lambda^+, \alpha_1, \alpha_2, m_{\text{max}}, Y_{\text{max}}, MI,MNI, (d(\omega), \omega \in \Omega^o),(N (l), l \in L), x^o, R_n(x^o))
\]

\[
\text{Tabu}(x^o, R_n(x^o), M_{\text{drop}}; MI,MNI, 1; x^*, R_n(x^*))
\]

The next section evaluates and compares these strategies.

**Computational Results**

**Plan of Experiments**

In order to test the heuristic approach proposed to solve the SMLTP, several problem instances were generated based on the following four dimensions: the problem size, the cost structure, the demand process and the 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 US regions and all distances were calculated with PC*MILER (www.alk.com), for the current US road network. For all cases, it was assumed that the order inter-arrival times are exponentially distributed with a mean inter-arrival time $\lambda$, and that order sizes are log-Normal with a mean $\mu$ and a standard-deviation $\sigma$.

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 DCs and ship-to-points considered, and in terms of the geographical region in the US 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 DCs, and the later was fixed at about $3\% |P|$. In order to capture different cost structures, two levels of fixed and variable costs were considered, as specified in Table 2. The fixed operating costs, $A_i$, and the unit value of products, $v_i$, were selected randomly in the interval specified in the table. The products price on the market, $u_p$, was fixed equal to the value in the table for all ship-to-points.

<table>
<thead>
<tr>
<th>Problem Instance</th>
<th>Geographical Area</th>
<th>Number of warehouses</th>
<th>Number of Ship-to-points</th>
</tr>
</thead>
<tbody>
<tr>
<td>$P_1$</td>
<td>Central North Eastern US States</td>
<td>7</td>
<td>206</td>
</tr>
<tr>
<td>$P_2$</td>
<td>North Eastern &amp; Midwest US States</td>
<td>15</td>
<td>706</td>
</tr>
<tr>
<td>$P_3$</td>
<td>North Eastern &amp; Midwest US States</td>
<td>28</td>
<td>1206</td>
</tr>
</tbody>
</table>

**Table 1- Test Problems Size**

<table>
<thead>
<tr>
<th>$[A_i]$; $[v_i]$; $u_p$</th>
<th>High product value/price</th>
<th>Low product value/price</th>
</tr>
</thead>
<tbody>
<tr>
<td>High fixed costs</td>
<td>$(a): [230K, 250K]; [19, 21]; 23$</td>
<td>$(b): [230K, 250K]; [9, 11]; 13$</td>
</tr>
<tr>
<td>Low fixed costs</td>
<td>$(c): [130K, 150K]; [19, 21]; 23$</td>
<td>$(d): [130K, 150K]; [9, 11]; 13$</td>
</tr>
</tbody>
</table>

**Table 2- Test Problems Cost Structure**

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) networks composed mainly of large and medium size customers ($LN$), and 2) networks including mainly small and medium size customers ($SN$). Table 3 provides the proportion of each type of customers in $LN$ and $LS$ networks, as well as the $(\lambda, \mu, \sigma)$ parameter values range used to generate specific instances. Finally, two random replications ($DS^1, DS^2$) are generated for each network structure considered.

<table>
<thead>
<tr>
<th>Ship-to-point size and characteristics</th>
<th>Large</th>
<th>Medium</th>
<th>Small</th>
</tr>
</thead>
<tbody>
<tr>
<td>Larger Ship-to-points Network ($LN$)</td>
<td>15%</td>
<td>65%</td>
<td>20%</td>
</tr>
<tr>
<td>Smaller Ship-to-points Network ($SN$)</td>
<td>10%</td>
<td>30%</td>
<td>60%</td>
</tr>
<tr>
<td>$\mu$ (cwt)</td>
<td>[480, 580]</td>
<td>[300, 400]</td>
<td>[120, 220]</td>
</tr>
<tr>
<td>$\sigma$ ($% \mu$)</td>
<td>7%</td>
<td>10%</td>
<td>16%</td>
</tr>
<tr>
<td>$\lambda$ (days)</td>
<td>[2.5, 4.5]</td>
<td>[5.5, 15.5]</td>
<td>[20.5, 35.5]</td>
</tr>
</tbody>
</table>

**Table 3- Ship-to-Point Demand Structure**

The combination of these four dimensions yields 48 problem instances. Each instance is denoted as follows:
The three neighbourhood exploration strategies proposed in the previous section were tested for all 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 three 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 non-dominated routes.

**Procedures Calibration**

As mentioned, the solution approach uses several procedures which are based on a set of parameters calibrated a priori. Using several $P_1$ and $P_2$ instances, the three strategies have been executed alternatively in order to fix the heuristic parameters specified in the initialization procedure. *Table 4* presents, for each parameter, the range of values tested and provides the value selected. Note that these values are fixed for the three strategies and present the best trade-off in terms of algorithm search speed and solution accuracy.

<table>
<thead>
<tr>
<th>Procedure</th>
<th>User</th>
<th>Reassign</th>
<th>Tabu</th>
</tr>
</thead>
<tbody>
<tr>
<td>Parameter</td>
<td>γ</td>
<td>λ⁻</td>
<td>λ⁺</td>
</tr>
<tr>
<td>Values range</td>
<td>[1,50]</td>
<td>[50,400]</td>
<td>[0,1.5]</td>
</tr>
<tr>
<td>Selected value</td>
<td>10</td>
<td>1</td>
<td>1.2</td>
</tr>
</tbody>
</table>

*Table 4- Heuristic Parameters Values*

Once the **User**, **Reassign** and **Tabu** procedures parameters were fixed, they were used to investigate the number $n$ of scenarios to generate (with the **MonteCarlo** procedure) to obtain good solutions. Several sample replications with different sizes ($n=1, 2, 4, 6, 8, 10$) were generated for $P_2$ instances. For each scenario sample, the design obtained with strategy 1 was evalu-
ated, and the average design value for a given sample size was calculated. These values are plotted in Figure 9 where the 100% line is the average value for all the replications. The diamonds and squares represent the design values and the average design value by sample size, respectively. As seen in this figure, for samples of size 1, 2 and 4 the spread of the design values is wider, but it stabilizes for larger sample sizes. In addition, the average value is approximately the same for sample sizes 6, 8 and 10. However, the solution time increases considerably with the sample size. For these reasons, a sample size of \( n = 6 \) scenarios has been selected as a best trade-off to use in the experiments.

![Figure 9- Scenarios Analysis Performed with Strategy 1](image)

**Analysis of Results**

This section discusses the quality of the supply network designs obtained with the three neighbourhood exploration strategies proposed, as well as their respective solution times. First, in order to determine how close to the optimum the solutions obtained are, the SAA model (14-18) was solved to optimality with CPLEX-11, with a sample size \( n = 4 \), for a small number of P1 instances. These are the largest problems we were able to solve to optimality. For these problems, the optimum was always reached by at least one of our three exploration strategies, which leads us to believe that our heuristic provide close-to-optimal solutions. Next, our three exploration strategies were compared for all problem instances. The mean network design values obtained are presented in Table 5 for problem instances with different cost structures, demand processes and sizes. Note that, for a given problem instance, to ensure that solution strategies are compared on the same basis, the sample of scenarios used in the heuristic was the same for the three exploration strategies.

The first observations that come out are that none of the three proposed strategies is dominated, and that, for a given problem instance, the alternative solutions obtained differ by less than 3% in terms of design value. In most cases, the same location decisions are obtained, and the difference in value comes from different ship-to-points allocation decisions. Note that in
some cases, two strategies produce the same design but small variations are observed in the transportation costs due to the random perturbations performed in the modified Clark and Wright algorithm used to solve the user model.

<table>
<thead>
<tr>
<th>P_1</th>
<th>a</th>
<th>b</th>
<th>c</th>
<th>d</th>
<th>LN</th>
<th>SN</th>
<th>All</th>
</tr>
</thead>
<tbody>
<tr>
<td>S_1</td>
<td>2 120 523</td>
<td>1 958 710</td>
<td>3 366 561</td>
<td>3 318 466</td>
<td>3 474 400</td>
<td>1 907 729</td>
<td>2 691 065</td>
</tr>
<tr>
<td>S_2</td>
<td>2 120 262</td>
<td>1 935 323</td>
<td>3 366 173</td>
<td>3 317 982</td>
<td>3 465 285</td>
<td>1 904 584</td>
<td>2 684 935</td>
</tr>
<tr>
<td>S_3</td>
<td>2 088 331</td>
<td>1 945 334</td>
<td>3 365 429</td>
<td>3 317 647</td>
<td>3 450 787</td>
<td>1 907 583</td>
<td>2 679 185</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>P_2</th>
<th>a</th>
<th>b</th>
<th>c</th>
<th>d</th>
<th>LN</th>
<th>SN</th>
<th>All</th>
</tr>
</thead>
<tbody>
<tr>
<td>S_1</td>
<td>9 550 162</td>
<td>8 822 303</td>
<td>4 914 403</td>
<td>9 608 100</td>
<td>10 692 208</td>
<td>5 755 276</td>
<td>8 223 742</td>
</tr>
<tr>
<td>S_2</td>
<td>9 496 354</td>
<td>8 812 047</td>
<td>4 916 613</td>
<td>9 610 701</td>
<td>10 659 665</td>
<td>5 758 193</td>
<td>8 208 929</td>
</tr>
<tr>
<td>S_3</td>
<td>9 535 461</td>
<td>8 785 367</td>
<td>4 899 761</td>
<td>9 594 118</td>
<td>10 671 089</td>
<td>5 736 264</td>
<td>8 203 677</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>P_3</th>
<th>a</th>
<th>b</th>
<th>c</th>
<th>d</th>
<th>LN</th>
<th>SN</th>
<th>All</th>
</tr>
</thead>
<tbody>
<tr>
<td>S_1</td>
<td>20 680 572</td>
<td>15 973 496</td>
<td>20 087 444</td>
<td>17 508 878</td>
<td>23 509 110</td>
<td>13 616 284</td>
<td>18 562 697</td>
</tr>
<tr>
<td>S_2</td>
<td>20 585 319</td>
<td>16 008 181</td>
<td>20 109 451</td>
<td>17 504 457</td>
<td>23 461 568</td>
<td>13 642 136</td>
<td>18 551 852</td>
</tr>
<tr>
<td>S_3</td>
<td>20 587 619</td>
<td>15 906 228</td>
<td>20 070 302</td>
<td>17 484 097</td>
<td>23 455 746</td>
<td>13 568 377</td>
<td>18 512 061</td>
</tr>
</tbody>
</table>

Table 5- Mean Design Values for all Problem Types

Figure 10- Exploration Strategies Design Value and Solution Time Comparisons
Comparisons of the three strategies’ design value and solution time are provided in Figure 10. 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 computational times (in seconds) for $P_1$, $P_2$ and $P_3$.

These results show that Strategy 1 performs better in general: it provides the best solution for 54% of the problem instances solved, and it is within 0.5% of the best solution found for 98% 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 gives excellent results for all the instances with smaller ship-to-points (SN). It is also the fastest strategy. Strategy 3 gives the best design only for four problem instances. It gives better results for networks with smaller ship-to points (SN) and with high depot costs and low product value/price ($b$). Due to the intensification phase added in Strategy 3, the number of shift moves grows rapidly and, for large problems, the solution time is significantly larger than for the two other strategies.

Figure 11 illustrates the network structure obtained with Strategy 1 for problem instance $P_2$-$c$-$LN$-$DS^1$. It can be seen that the model proposed provides a well-balanced hub-and-spoke network structure. It shows also that the next-day-delivery constraints imposed by limiting customer assignments to depots in a 400 miles radius are respected.
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 DCs in order to save on transportation costs. Conversely, when ship-to-points are smaller, the design obtained includes less DCs. Our results also show that the route-length estimation formula (27) is very accurate. Its use 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 Multi-period 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 showed how a sample of multi-period demand scenarios can be generated from the stochastic demand processes of customers, and used to solve the problem. Since, the resulting sample average approximation MIP is extreme large, it cannot be solved to optimality for realistic problems, and an efficient 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. In addition, three different strategies, based on drop, add and shift moves, were proposed to explore the neighbourhood of a solution.

In order 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 Strategy 1 gives the best results. The superior performance of the nested solution approach proposed is principally due to the good quality of the anticipation of transportation costs in the design problem, and to the efficient combination of several procedures and heuristics. The three search strategies examined could certainly be refined and fine-tuned, and other strategies could be elaborated, but we leave this for further research. The approach as it stands is sufficiently evolved to solve most practical cases efficiently and effectively.
References


