CIRRELT

Centre interuniversitaire de recherche sur les réseaux d'entreprise, la logistique et le transport

Interuniversity Research Centre on Enterprise Networks, Logistics and Transportation

A Variable Neighborhood Search Heuristic for the Design of Multicommodity Production-Distribution Networks with Alternative Facility Configuration

Houssam Amrani
Alain Martel
Nicolas Zufferey
Polina Makeeva

August 2008

CIRRELT-2008-35

Bureaux de Montréal :
Université de Montréal
C.P. 6128, succ. Centre-ville
Montréal (Québec)
Canada H3C 3J7
Téléphone : 514 343-7575
Télécopie : 514 343-7121

Bureaux de Québec :
Université Laval
Pavillon Palasis-Prince, local 2642
Québec (Québec)
Canada G1K 7P4
Téléphone : 418 656-2073
Télécopie : 418 656-2624

www.cirrelt.ca

Université Laval · UQÀM · HEC Montréal · École Polytechnique Montréal · Université de Montréal

A Variable Neighborhood Search Heuristic for the Design of Multicommodity Production-Distribution Networks with Alternative Facility Configurations

Houssam Amrani[1,2], Alain Martel[1,2,*], Nicolas Zufferey[1,3], Polina Makeeva[4]

[1] Interuniversity Research Centre on Enterprise Networks, Logistics and Transportation (CIRRELT)

[2] Département des opérations et systèmes de décision, Pavillon Palasis-Prince, 2325, rue de la Terrasse, Université Laval, Québec, Canada G1V 0A6

[3] HEC, University of Geneva, Faculté des Sciences Économiques et Sociales, Boulevard du Pont-D'Arve 40, 1211 Geneva 4, Switzerland

[4] School of Computer and Communication Sciences, Swiss Federal Institute of Technology, 1015 Lausanne, Switzerland

Abstract. This paper proposes efficient heuristics to solve large-scale production-distribution network design models. The problem studied is an extension of the two-echelon multicommodity CFLPSS (Capacitated Facility Location Problem with Single Sourcing) considering direct shipments from manufacturing facilities, alternative facility configurations, and concave inventory holding costs. After reviewing the relevant literature, a detailed description of the production-distribution network design problem studied is provided, and the problem if formulated as a mixed-integer program. The heuristic solution approach proposed is then presented. It is a variable neighbourhood search (VNS) method integrating a tabu procedure. Experiments are designed to calibrate the heuristics developed, and to compare their performance with the CPEX solver, for problems with different realistic characteristics. Computational results are presented and discussed.

Keywords. Production-distribution networks, location-allocation problem, facilities configuration, tabu search, variable neighborhood search.

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

---

* Corresponding author: Alain.Martel@cirrelt.ca

Dépôt légal – Bibliothèque et Archives nationales du Québec,
Bibliothèque et Archives Canada, 2008

Introduction

Under the pressure of accrued global competition, several industrial sectors have witnessed important mergers and acquisitions over recent years. These industrial re-organizations lead to the reengineering of the companies production-distribution networks, and the resulting supply chains seem to be continually growing in size. The reengineering of a supply chain network involves critical strategic decisions related to opening and/or closing production and distribution centers, to the reconfiguration of some of these centers, and to the specification of their mission in terms of the products they should produce and/or stock and in terms of the customers they should supply. When getting involved in such reorganizations, companies aim to improve their competitive position, by simultaneously making large economies of scale and providing better service to their customers (Cooke, 2007). In order to reach the best strategic trade-off between total network costs and service levels, it is possible to rely on the solution of large mixed-integer programming models incorporating concave cost functions to capture economies of scale. However, despite the increasing power of commercial solvers, given the huge size of the problems involved, it may be extremely difficult to solve these models to optimality. The aim of this paper is to propose an efficient heuristic to solve a typical large-scale production-distribution network optimization model.

Production-distribution networks can be represented by multi-level directed graphs (digraphs). The digraph nodes are associated with external entities such as product family sources and demand zones (set of similar customers in a geographical region), and to internal entities such as potential production-distribution center locations and configurations. The arcs, and the paths, in the digraph are associated with product flows. Fixed location/configuration costs and variable production, handling, storage and inventory holding costs are associated with the internal nodes, and variable procurement and transportation costs with the arcs. The variable costs, however, are not necessarily linear, and only a subset of the potential network arcs may be considered to ensure that predetermined service levels are satisfied. Given a forecasted demand over a fixed planning period, the objective is to find a feasible sub-digraph minimizing the total system costs. The nodes of an optimal sub-digraph specify the location and configuration of the facilities to use, and its arcs specify their mission in terms of the products to produce and stock, and of the downstream nodes in the network to supply. The complexity of the resulting mixed-integer programming model depends on the number of product families (commodities) and digraph levels (echelons) involved, on whether capacity is constrained and alternative facility configurations are considered, on the linearity or non-linearity of costs, and on whether or not the customers in a demand zone must be supplied by a single source.

The simplest discrete facility location models deal with uncapacitated or capacitated facility location problems (UFLP/CFLP). They consider a single product and a single production/distribution echelon. Their original formulation goes back to Balinski (1961) and, despite their simplicity, they are still the subject of numerous publications (ReVelle, 2008). In the CFLP, demand can be supplied from more than one source. When it is required that each demand zone is supplied from a single source, the resulting CFLP with single sourcing (CFLPSS) is much more difficult to solve. In fact, the generalized assignment sub-problem obtained for a given set of facilities is NP-hard (Fisher, 1986). Kaufman (1977) studied an extended version of the UFLP incorporating a production echelon and a distribution echelon. Several authors also studied multi-product extensions of the one or two echelon CFLP and CFLPSS. Geoffrion and Graves (1974) proposed a Benders decomposition approach to solve a path-based formulation of a multicommodity CFLPSS, with fixed production facilities and location-allocation decisions for the distribution echelon. Hindi and Basta (1994) solve an arc-based formulation of a similar problem with a Branch and Bound algorithm. Hindi et al. (1998), Klose (2000) and Pirkul and Jayaraman (1996, 1998) present Lagrangian relaxation procedures to solve two-echelon CFLPSS's and CFLP's. Klose and Drexl (2005) present a recent review of the large literature available on these problems.

The models reviewed in the previous paragraph assume that the facilities capacity is predetermined, and that the fundamental tradeoffs are between facilities fixed operating costs and variable linear production, warehousing and transportation costs. In real life, however, capacity options are usually available in the facilities, and economies of scale in production, inventory holding and/or transportation costs are often present. The importance of capacity as a decision variable in location problems was recognized early (Elson, 1972), but location-allocation models incorporating capacity decisions explicitly were studied only more recently. Some models consider capacity expansion as a continuous variable (Verter and Dincer, 1995) but, more realistically, others consider discrete facility capacity options (Paquet *et al*., 2004; Amiri, 2006) or alternative facility configurations (Eppen *et al.*, 1989). Soland (1974) and Kelly (1982) studied a CFLP with concave variable facility costs, and Fleischmann (1993) considers a multi-echelon location problem with transport economies of scale. The importance, in production-distribution network design, of modeling inventory holding costs through concave inventory-throughput curves is stressed in Ballou (1992). Martel and Vankatadri (1999) propose a model with capacity expansion projects under economies of scale. A production-distribution network design framework integrating all these extensions is presented in Martel (2005). The problem studied in this paper is an extension of the multicommodity CFLPSS with two production-distribution echelons considering direct shipments from production-distribution facilities to demand zones, alternative facility configurations, and concave inventory holding costs. This problem is clearly NP-hard. A good practical illustration of the type of network design problems considered is the Usemore Soap Company Case presented in Ballou (1992).

The first heuristics proposed to solve UFLPs and CFLPs were interchange procedures based on facility ADD, DROP or SHIFT moves (Kuenh and Hamberger, 1963; Jacobsen, 1983; Bornstein *et al*., 2004). A procedure to solve the CFLP based on multiple simultaneous interchanges was also developed by Zhang *et al*. (2005). Tabu search procedures to solve the UFLP were published by Al-Sultan *et al*. (1999) and Michel *et al*. (2004). Sörensen (2002) developed a tabu heuristic for the CFLP. Kratica *et al*. (2001) proposed a genetic algorithm for the UFLP, and Barahona *et al*. (2005) developed a randomized rounding procedure to solve the UFLP/CFLP. Beasley (1993), Delmaire *et al*. (1999) and Ahuja *et al*. (2004) solved the CFLPSS, with a Lagrangean heuristic procedure, a tabu search procedure and a very large-scale neighborhood (VLSN) search procedure, respectively. Crainic *et al.* (1993) and Gendron *et al.* (2003) proposed tabu search procedures for a multicommodity UFLP with balancing requirements. Finally, Klose (1999) present a LP-based heuristic to solve a two-echelon CFLPSS. The heuristics proposed in this paper to solve the extended multicommodity two-echelons CFLPSS considered are Variable Neighbourhood Search (VNS) methods integrating tabu procedures.

The paper is organized as follows. The next section provides a detailed description of the production-distribution network design problem studied and it models it as a mixed-integer program. The following section presents the heuristic solution approach proposed. Finally, in the last section, experiments are designed to calibrate the VNS heuristics developed, and to compare their performance with the CPLEX solver, for problems with different realistic characteristics. Computational results are presented and discussed.

Problem Description and Formulation

Consider a potential production-distribution network such as the one illustrated in Figure 1. This network is composed of sites for potential production-distribution centers $u \in U$ and distribution centers $w \in W$, corresponding to existing facilities or to locations where a facility could be opened. The plants manufacture a set of finished products $p \in P$. A product, in this type of strategic planning problem, corresponds to a family of items requiring the same type of production capacity. It is possible however that some production-distribution sites $u \in U$ are considered for focussed factories producing only a subset of products $P_u \subseteq P$, i.e. it may be possible to manufacture a product p only on a subset of sites $U_p \subseteq U$. The facilities must supply a set of external demand zones (group of ship-to-points located in a given geographical area) $d \in D$. A demand zone d, however, may require only a subset $P_d \subseteq P$ of products. The production-distribution centers can stock finished products, and their mission is to supply the distribution centers as well as some demand zones (direct shipments). However, each demand zone must be supplied by a single source, i.e. by a single production-distribution or distribution center. Also, in order to respect some predetermined service criteria (e.g. next day delivery), only a subset $S_d \subseteq S = U \cup W$ of the sites are positioned to supply a given demand zone $d \in D$, or conversely a potential facility $s \in S$ may supply only a subset of demand zones $D_s \subseteq D$.

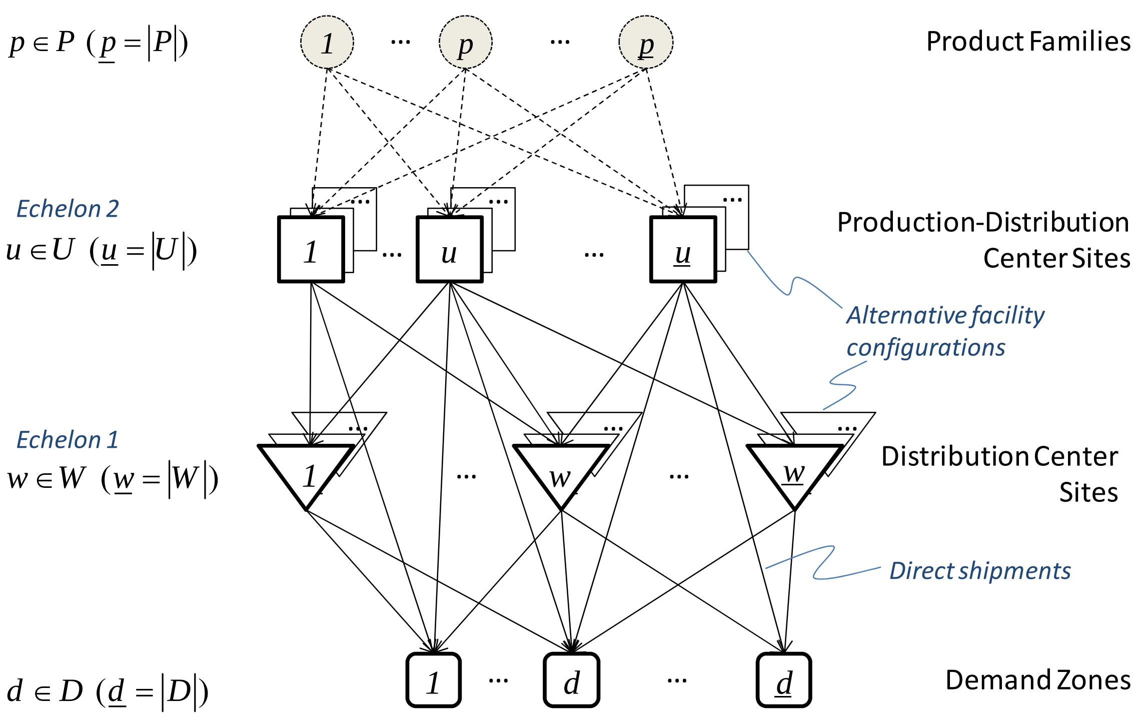

Figure 1: Potential Production-Distribution Network

The facilities already in place are characterized by a configuration specifying their production and storage capacity, as well as their fixed and variable costs. Alternative configurations can however be considered for each site. These alternative configurations may correspond to a reengineering of current layouts or equipments, to the addition of new space and/or equipment to expand capacity, or to different facility specifications for new sites. Alternative configurations may be associated with different equipment size to capture economies of scale. For each site $s \in S$ in the potential network, a set J_s of possible configurations could thus be implemented. For the planning period considered, each configuration $j \in J_s$ is characterized by:

- A dedicated production capacity b_{jps}, $s \in U$, expressed in terms of an upper bound on the quantity of product $p \in P_s$ that can be produced.
- A flexible storage capacity b_{js}, $s \in S$, expressed as a maximum throughput in terms of a standard pack (e.g. a pallet). A size parameter q_p is used to convert the flow of product p in this standard pack.
- A fixed exploitation cost A_{js}, $s \in S$, for the planning period considered. This fixed cost takes into account the state of the site (empty site, owned facility on site, leased facility on site, public warehouse…) at the beginning of the planning period.
- A variable throughput cost c_{jps}, $s \in S$, covering relevant procurement, reception, production, handling and shipping expenses.

In the heuristic proposed, we assume that the configurations considered for node $s \in S$ can be numbered so that $b_{1ps} \leq b_{2ps} \leq ... \leq b_{\underline{j}_s ps}$, $p \in P_s$; $b_{1s} \leq b_{2s} \leq ... \leq b_{\underline{j}_s s}$ $(\underline{j}_s = |J_s|)$. We also assume that management would not operate a center on a site $s \in S$ unless the quantity of products manufactured is at least $\underline{X}_{ps}$, for all $p \in P_s$, if s is a production-distribution site, and unless the throughput (in storage packs) in the storage facility is at least $\underline{X}_s$.

The level of order cycle stocks and of safety stocks in the facilities depends on the operations management policies of the company and on the ordering behavior of customers. It can be shown (Martel, 2003) that, when sound inventory management and forecasting methods are used, the relationship between the throughput X_{ps} of product p in storage facility s and the average cycle and safety stock $I_p(X_{ps})$ required to support this throughput takes the form of the following power function: $I_p(X_{ps}) = \alpha_p (X_{ps})^{\beta_p}$, with $\beta_p < 1$ to reflect economies of scale. The parameters α_p and β_p of this function are obtained by regression, from historical or simulation data (Ballou, 1992). We assume here that the throughput X_{ps} used as an argument in this function is the sum of all product p shipments from center s to demand zones. For production-distribution centers, if the products manufactured for distribution centers are not shipped immediately after production, it could be more appropriate to also include flows to DCs in the throughput variables X_{pu}, $u \in U$. It could also be more appropriate to use different inventory-throughput functions for production-distribution and for distribution centers. The model proposed can be easily modified to accommodate such problem variants.

Most supply chain network design models proposed in the literature do not take the risk pooling effects captured by the function $I_p(X_{ps})$ into account: they assume either explicitly or implicitly that the relationship between inventory levels and throughput is linear. If the historical annual throughput level and average inventory level observed for product *p*, in facility *s*, are X_{ps}^o and $I_p(X_{ps}^o)$, respectively, then the ratio $X_{ps}^o / I_p(X_{ps}^o)$ is the familiar inventory turnover ratio, and its inverse $\rho_{ps} = I_p(X_{ps}^o)/X_{ps}^o$ is the number of years of inventory kept in stock. Assuming that the relationship between inventory level and throughput is linear boils down to approximating $I_p(X_{ps})$ by $\rho_{ps} X_{ps}$. Since the facilities' throughputs are not known before the network design model is solved, and since they can be far from historical values (mainly if a new facility is opened or an existing one is closed), calculating inventory levels with historical inventory turnover ratios can be completely inadequate. An effort is therefore made in this paper to take risk pooling effects into account explicitly. Starting from the inventory-throughput function just defined, and taking into account the average unit inventory holding cost r_{ps} of the products of family *p* in facility *s*, the following facilities inventory holding cost functions are obtained:

$$H_{ps}(X_{ps}) = r_{ps} I_p(X_{ps}) = r_{ps} \alpha_p (X_{ps})^{\beta_p}, \quad s \in S,\ p \in P_s \tag{1}$$

To formulate the model, the following additional data is required:

x_{pd} : Demand for product $p \in P_d$ in demand zone *d* during the planning period considered

f_{psn} : Unit transportation cost for product *p* between origin $s \in S$ and destination $n \in S \cup D$

In order to find the optimal design, the following decision variables are required:

Y_{js} : Binary variable equal to 1 if configuration $j \in J_s$ is selected for facility *s*, and to 0 otherwise

Z_{sd} : Binary variable equal to 1 if demand zone *d* is supplied by facility $s \in S_d$, and to 0 otherwise

F_{puw} : Number of units of product *p* transported from production-distribution center $u \in U_p$ to distribution center $w \in W$ during the planning period

X_{jps}: Number of units of product p flowing out of facility s during the planning period when configuration $j \in J_s$ is implemented

X_{ps}: Number of units of product p supplied to demand zones from facility s during the planning period

Using this notation, the concave multicommodity two-echelons CFLPSS with alternative facility configurations considered can be formulated as follows:

$$
\begin{aligned}
\text{Min } TC = & \sum_{s \in S} \sum_{j \in J_s} A_{js} Y_{js} && (\textit{Facilities fixed costs}) \quad (2)\\
& + \sum_{p \in P} \sum_{s \in S} \sum_{j \in J_s} c_{jps} X_{jps} && (\textit{Facilities variable costs})\\
& + \sum_{d \in D} \sum_{s \in S_d} \Big(\sum_{p \in P_d} f_{psd} x_{pd} \Big) Z_{sd} + \sum_{p \in P} \sum_{u \in U_p} \sum_{w \in W} f_{puw} F_{puw} && (\textit{Transportation costs})\\
& + \sum_{p \in P} \sum_{s \in S} H_{ps}(X_{ps}) && (\textit{Inventory holding costs})
\end{aligned}
$$

Subject to:

- Demand constraints:

$$\sum_{s \in S_d} Z_{sd} = 1 \qquad d \in D \tag{3}$$

- Facility configuration selection constraints:

$$\sum_{j \in J_s} Y_{js} \le 1 \qquad s \in S \tag{4}$$

- Storage facility throughput definition constraints:

$$X_{ps} - \sum_{d \in D_s} x_{pd} Z_{sd} = 0 \qquad p \in P,\ s \in S \tag{5}$$

- Flow equilibrium constraints:

$$\sum_{j \in J_u} X_{jpu} - X_{pu} - \sum_{w \in W} F_{puw} = 0 \qquad p \in P,\ u \in U_p \tag{6}$$

$$\sum_{j \in J_w} X_{jpw} - \sum_{u \in U_p} F_{puw} = 0 \qquad p \in P,\ w \in W \tag{7}$$

$$X_{pw} - \sum_{j \in J_w} X_{jpw} = 0 \qquad p \in P,\ w \in W \tag{8}$$

- Capacity constraints:

$$\underline{X}_{pu} Y_{ju} \le X_{jpu} \le b_{jpu} Y_{ju} \qquad p \in P,\ u \in U_p,\ j \in J_u \tag{9}$$

$$\underline{X}_s \sum_{j \in J_s} Y_{js} \le \sum_{p \in P} q_p X_{ps} \le \sum_{j \in J_s} b_{js} Y_{js} \qquad s \in S \tag{10}$$

- Non-negativity and integrality constraints:

$$X_{ps} \ge 0 \quad p \in P,\ s \in S;\ X_{jps} \ge 0 \quad p \in P,\ s \in S,\ j \in J_s \tag{11}$$

$$F_{puw} \ge 0 \quad p \in P,\ u \in U_p,\ w \in W \tag{12}$$

$$Y_{js} \text{ binary} \quad s \in S,\ j \in J_s\ ;\ Z_{sd} \text{ binary} \quad s \in S,\ d \in D_s \tag{13}$$

An important particular case, which we shall examine, is the linear inventory holding cost case. The last term in the objective function is then replaced by $\Sigma_{p\in P}\Sigma_{s\in S}h_{ps}X_{ps}$, where h_{ps} is the inventory cost per unit of product *p* flowing in storage facility *s*. According to our previous discussion, this cost would typically be set equal to $\rho_{ps}r_{ps}$. For this particular case, (5) can be used to substitute $\Sigma_{d\in D_s}x_{pd}Z_{sd}$ for X_{ps} in the model, which simplifies it to some extent. The next section presents a variable neighborhood search heuristic to solve these problems.

Heuristic Solution Approach

In this section, we first describe the general elements of two well-known heuristics, namely tabu search and Variable Neighborhood Search (VNS). Both heuristics will then be adapted to the context of our problem.

Description of Tabu Search and Variable Neighborhood Search

Tabu search is a well-known local search heuristic originally proposed by Glover (1986) and by Hansen (1986). Its crucial component is to prevent the search to cycle by forbidding some moves during a certain number of iterations (also referred to as the duration of the tabu status). Many tabu search variants are found for example in Glover & Laguna (1997).

Let $N = \{N^{(1)},..., N^{(t)},..., N^{(t_{\max})}\}$ denote a finite set of neighborhoods, where $N^{(t)}(v)$ is the set of solutions in the t^{th} neighborhood of solution *v*. Most local search methods use only one type of neighborhood, i.e. $t_{\max} = 1$. The basic VNS, as proposed by Mladenovic and Hansen (1997), tries to avoid being trapped in local minima by using more than one neighborhood. Its basic steps are given in Figure 2.

The stopping condition of a VNS may be, for example, an upper bound on the CPU time, a maximum number of iterations, or a maximum number of iterations between two improvements of the incumbent. In step (2a), the neighbor solution can be generated at random, or can be the best one among a sample or all the solutions in $N^{(t)}(v)$. The local search method used in step (2b) can be a simple descent method, or a more powerful technique, such as tabu search or simulated annealing. This basic VNS can be viewed as a descent algorithm since the incumbent *v* is modified only if the local optimum *v*'' obtained in step (2b) is better than *v*. Without much additional effort, it is possible to transform this basic VNS into a descent-ascent method: in step (2c) it is also possible to set *v* = *v*'' with some probability even if *v*'' is worse than the incumbent.

1) Initialization
   a) Determine an initial solution *v*;
   b) Set $t = 1$;
2) Repeat the following until a stopping condition is met:
   a) *Shaking*. Generate a solution *v'* in $N^{(t)}(v)$;
   b) *Local search.* Apply some local search method with *v'* as initial solution; let *v''* be the local optimum so obtained;
   c) *Move or not.* If *v''* is better than the incumbent *v*, move there (i.e. set *v* = *v''*), and continue the search with $N^{(1)}$ (i.e. set $t = 1$); otherwise set $t = (t \bmod t_{\max}) + 1$;

Figure 2: A Basic VNS Algorithm

Adaptation of VNS to the Problem Considered

In the algorithm proposed, except when building an initial solution, we always generate feasible solutions. More precisely, we only consider options or moves for which: (1) all demands from the zones $d \in D$ are supplied by open centers $s \in S_d$ satisfying the service criteria, (2) the minimum and maximum capacity constraints of the centers configuration used are respected, and (3) if the demand from a zone *d* can be delivered from a single center (i.e. if $|S_d| = 1$), it is always kept open during the search. In addition, if the demand x_{pd}, $p \in P_d$, from a zone $d \in D_w$ is reassigned to a distribution center $w \in W$, for each $p \in P_d$, it then induces a new requirement (equal to x_{pd}) to the opened production centers that can deliver product *p* to distribution center *w*. Such requirements may also be induced when an existing production-distribution center is closed. In both situations, the requirements induced at the first echelon are assigned to the second echelon center *u* with the lowest unit production and transportation cost ($f_{puw} + c_{jpu}$), provided that a configuration *j* with sufficient capacity can be used. If capacity is not sufficient, then the outstanding requirement is assigned to the next best plant. This implies that the distribution centers can be supplied by more than one plant, as specified by the flow variables F_{puw} previously defined.

The key components of a solution of the problem can be represented as follows. For any solution *v*, there is a subset $W(v) \subseteq W$ of opened distribution centers, as well as a subset $U(v) \subseteq U$ of opened production-distribution centers. Thus, a solution *v* is partially characterized by the subsets $(W(v),\ U(v))$. In order to adapt the VNS algorithm to our problem, we mainly have: (1) to design some neighborhood structures, (2) to choose a way to generate a neighbor solution in step (2a), (3) to choose a local search for step (2b), and (4), to determine a stopping condition. The way to generate an initial solution must also to be discussed. Based on preliminary experiments, we decided to always choose the best neighbor solution in step (2a), and to stop the VNS algorithm when a predetermined time limit *MaxTime* is reached.

Our VNS method is based on five neighborhood structures, denoted $N^{(1)}$ to $N^{(5)}$. A solution v' in $N^{(1)}(v)$ is obtained by replacing an element of $U(v)$ with an element of $U \backslash U(v)$, i.e. we close a production-distribution center but we open another one (this is called a *U-shift* move). A solution v' in $N^{(2)}(v)$ is obtained from v by adding an element of $U \backslash U(v)$ to $U(v)$, i.e. we open an additional production-distribution center (this is called a *U-add* move). A solution v' in $N^{(3)}(v)$ is obtained from v by removing an element from $U(v)$, i.e. we close a production-distribution center (this is called a *U-drop* move). A solution v' in $N^{(4)}(v)$ is obtained from v by adding an element of $W \backslash W(v)$ to $W(v)$, i.e. we open an additional distribution center (this is called a *W-add* move). A solution v' in $N^{(5)}(v)$ is obtained from v by removing an element from $W(v)$, i.e. we close a distribution center (this is called a *W-drop* move). As the number of potential distribution centers is generally much larger than the number of production-distribution centers, it is relevant to test lots of different possibilities for $W(v)$. Consequently, the neighborhood consisting of replacing an element of $W(v)$ with an element of $W \backslash W(v)$ (this is called a *W-shift* move) will be used within the local search of step (2b).

Two key issues related to these neighborhood structures must be discussed: (1) which demand zones should be assigned to a center to be opened, following a shift or add move, and (2) to which centers should the demands of a center to be closed be reassigned, following a drop or shift move? In both cases, the costs affected are the transportation costs, the configuration costs, the production costs and the inventory holding costs. We tackle issue (1) as follows. Suppose that we would like to open a center s'. It is then relevant to assign demand zone $d \in D_{s'}$ to s' instead of its current supplier s if the total costs are reduced, and if the minimum capacity constraint of s is still respected. However, we observed that such a reassignment often leads to solutions not satisfying the minimum capacity constraint of s'. To avoid such violations, we give more clients to s' by proceeding as follows. While the minimum capacity constraint of s' is not respected, we randomly choose a demand zone $d \in D_{s'}$ not already supplied by s'. If there exists an assignment (s'', d), with s'' already open, that leads to a solution with larger costs than the assignment (s', d), then we assign d to s'. If such an assignment does not exist, we cannot open s'. In order to tackle issue (2), for any demand zone d associated with the center that we would like to close, we simply reassign d to the best (in term of costs) possible open center s^*.

In both cases, we then select the tightest center configuration, i.e. the feasible capacity configuration with the smallest fixed cost.

The five neighborhoods described previously are used as follows. Let v^* be the best solution encountered so far during the search. The process starts with $N = \{N^{(1)}, N^{(2)}, N^{(3)}, N^{(4)}, N^{(5)}\}$. Instead of initially setting $t = 1$, in step (2a) of the VNS we randomly choose t in {1, 2, …, 5}. In step 2b), the local search is performed to obtain $v"$. Then, in step (2c), if $v"$ is better than v, we set $N = \{N^{(1)}, N^{(2)}, N^{(3)}, N^{(4)}, N^{(5)}\}$ and we update v (i.e. we set $v = v"$), otherwise: if $|N| > 1$, and a predetermined condition Δ is satisfied, we remove $N^{(t)}$ from N, and if $|N| = 1$, we set $N = \{N^{(1)}, N^{(2)}, N^{(3)}, N^{(4)}, N^{(5)}\}$. The next selected neighborhood structure used in step (2a) is then randomly chosen in N. We test two possible conditions Δ, which are: (1) $v"$ is not better than v, and (2), $v"$ is not better than v^*. Thus, we have two VNS variants, respectively denoted VNS(1) and VNS(2). The pseudo-code of the proposed VNS is presented in Figure 4 below.

We propose a constructive method to build an initial solution in step (1a) of the VNS, and we use a tabu search in step (2b) of the VNS. These two methods are presented below.

Generation of an initial solution for the VNS algorithm

In order to generate an initial solution v, we use a 3-phase procedure denoted INIT. In the first phase, at each step, we randomly select a demand zone d not already covered, and we assign it to the center $s \in S_d$ which can supply it at minimum cost, while respecting the storage capacity of its largest possible configuration ($b_{\underline{j}_s s}$). The costs considered are not the same for the concave and linear cases, and in the calculations, the costs of the largest configuration $\underline{j}_s$ of each site s considered are used. For the linear case, we consider the transportation costs, the production cost, the inventory holding costs, and the configuration cost of s. For the concave costs case, the inventory holding costs are neglected. The different cost functions used to select the supplier s^* of demand zone d, and the supplier u^* of warehouse w, are summarized in Table 1 below.

Which center s^* should supply zone d?	**Linear inventory holding costs**	**Concave inventory holding costs**
Supply Echelon (s^* can be in S_d)	$s^* = \arg\min_{s \in S_d \mid P_d \subseteq P_s} [\sum_{p \in P_d} (f_{psd} + h_{ps}) + (A_{\underline{j}_s s} / b_{\underline{j}_s s})]$	$s^* = \arg\min_{s \in S_d \mid P_d \subseteq P_s} [\sum_{p \in P_d} f_{psd} + (A_{\underline{j}_s s} / b_{\underline{j}_s s})]$
Production Echelon (for $s^* \in W$)	We must supply products to warehouse s^* from a production center $u^* \in U$: $u^* = \arg\min_u \sum_{p \in P} (f_{pus^*} + c_{\underline{j}_u pu})$	

Table 1: Description of the Costs Considered Depending on the Context

Such a procedure implies that many centers will be opened, and sometimes, the minimum capacity constraints will be violated. In addition, the configuration of some centers will be unnecessarily large. Thus, in the second phase of the procedure, we close a center s if it does not satisfy the minimum capacity constraint, we greedily reassign its customers (considered in a random order) to minimize costs, and we adjust the configuration of all the centers in order to reduce the associated fixed costs. At this stage, the solution obtained is feasible.

In the third phase of the procedure, at each step, we randomly choose a demand zone d, and we try to reduce the total costs by supplying it from another open center (we test every possibility and keep the best option). In this phase, any demand zone d can be reassigned a single time. The costs involved in the comparisons are the transportation costs (which are easy to update), the inventory holding costs and the configuration costs. More precisely, if demand zone d is supplied from center s' instead of center s, this may: (1) reduce the configuration cost of s, (2) increase the configuration cost of s', and (3), modify the inventory holding costs of s and s' (because such costs depend on the quantity of product which transit through s and s'). Note that we do not consider options for which the capacity constraints or the service criteria are violated. Of course, we do not assign d to another center if the costs are not reduced.

Tabu search with a fixed number of open distribution centers

The initial solution v' provided to tabu search is given by step (2a) of the VNS. Given a current solutions v, the neighborhood structure involves W-shift moves, i.e. the replacement of an open center $w \in W(v)$ with a closed one $w' \in W \backslash W(v)$. Note that only W-shift moves (w; w') such that w and w' could supply common demand zones (i.e. such that $D_w \cap D_{w'} \neq \emptyset$) are considered. Let $UNIF(e_1;e_2)$ be a function providing a random integer between the parameters e_1 and e_2 (e_1 and e_2 included). When a W-shift move $(w;w')$ is performed, it is then tabu to close w', and to open w, for $UNIF(e_1;e_2)$ iterations. Note that more sophisticated tabu tenures were tested (e.g. dynamic tabu lists), but they did not lead to better results. Thus, we prefer to keep our algorithm as simple as possible. We stop the proposed tabu search procedure when a maximum number θ of iterations without improving the best solution v'' encountered so far is reached, where θ is a parameter. A key issue in tabu search is the selection of a neighbor solution at each iteration. Because the objective function TC is very cumbersome, it would be too time consuming to test every possible neighbor solution at each iteration. In order to reduce the computational effort, an approach is to use an auxiliary objective function TC' which can be computed quickly. Given a current solution v, the role of TC' is to quickly evaluate which distribution centers are interesting to be closed (by interesting, we mean that the total costs would probably be reduced if we replace it by another distribution center). In other words, TC' is used to determine a subset $W^{\kappa}(v)$ of $W(v)$ which contains the κ worst distribution centers, where κ is a parameter. Let $D_w(v)$ be the demand zones supplied from *w* in solution *v*, and $X_{pw}(v)$ be the throughput of product *p* in center *w* for solution *v*. We propose to use the function $TC'(w)$, for all $w \in W(v)$, to evaluate the cost improvement which could result if center *w* was replaced by another center. This function is different for the linear and concave inventory holding cost cases and, letting $\rho_{pw}(v) = I_p(X_{pw}(v))/X_{pw}(v)$, it is defined as follows:

Linear inventory holding costs	$TC'(w) = \min_{w' \in W \setminus W(v)} [\sum_{d \in D_w(v)} (\sum_{p \in P_d} (f_{pw'd} - f_{pwd})) + \sum_{p \in P_d} (h_{pw'} - h_{pw})]$
Concave inventory holding costs	$TC'(w) = \min_{w' \in W \setminus W(v)} [\sum_{d \in D_w(v)} (\sum_{p \in P_d} (f_{pw'd} - f_{pwd})) + \sum_{p \in P_d} (r_{pw'} - r_{pw}) \rho_{pw}(v)]$

If $TC'(w) < 0$, it means that at least another distribution center w' can supply the demand zones associated with *w* at a lower cost. The subset $W^{\kappa}(v)$ is obtained by sorting the centers $w \in W(v)$ in increasing order of $TC'(w)$ and by taking the first κ elements in the list. Then, we compute the value of each move $(w;\ w')$ with the true objective function *TC*, for $w \in W^{\kappa}(v)$ and for $w' \in W \setminus W(v)$. We now have all the key ingredients of our tabu search for a fixed number of distribution centers. The algorithm is summarized in Figure 3. The pseudo code for VNS(1) and VNS(2) is given in Figure 4.

1) Initialization
   a) Set v = solution provided by step (2a) of the VNS;
   b) Set *iter* = 0, $v'' = v$ and $TC'' = +\infty$;
2) While *iter* < θ, do:
   a) Set *iter* = *iter* + 1;
   b) Compute the set $W^{\kappa}(v)$ of the κ distribution centers $w \in W(v)$ with the lowest values of $TC'(w)$;
   c) Set v^* as the best solution (according to *TC*) among the ones obtained from *v* by performing a move $(w;\ w')$, where $w \in W^{\kappa}(v)$ and $w' \in W \setminus W(v)$;
   d) If $TC(v^*) < TC''$, set $v'' = v^*$, $TC'' = TC(v^*)$ and *iter* = 0;
   e) Update the tabu status: it is forbidden to open *w* or close w' for $UNIF(e_1;e_2)$ iterations;

Figure 3: Tabu Search with a Fixed Number of Distribution Centers

Experimental Evaluation

Test Problems and Calibration

In order to test the heuristic approach proposed, problem instances of different size and cost structure were generated. 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). The problems were defined over various United-States regions and all distances were calculated with PC*MILER (www.alk.com), for the current US road network. The demand for the various demand zones considered was generated using a Uniform distribution, with lower and upper bounds based on the total production capacity of the network. In all the problem instances created, it was assumed that demand zones had to be supplied from facilities located at a distance not exceeding 530 miles from its centroid.

Problems of different size were obtained by changing the number of product families (3 or 20), demand zones (500 or 1000), potential distribution centers (60 or 100) and potential production-distribution centers (4 or 6). For the DCs, 2 different configurations were considered, and 4 configurations were taken into account for the production-distribution centers. For the linear inventory holding costs case, to obtain different cost structures, 2 unit inventory holding cost values (r_{ps}), 2 unit transportation cost (f_{psd} and f_{puw}) levels, and 2 fixed exploitation cost (A_{js}) levels were used. The same configuration variable throughput costs (c_{jps}), and the same inventory-duration parameters, ρ_{ps}, were used in all problems. For the concave inventory holding cost case, a single cost structure, generated randomly, was used in all problems with the same inventory-throughput function parameters (α_p, β_p). The test problems generated are described in Table 2, where the problem instances are defined using the following notation: $|P|\mathrm{P}-|D|\mathrm{D}-|W|\mathrm{W}-|U|\mathrm{U}$. The X-values in the Cost Structures columns represent the multiplication factor used to generate the problem costs from a base case.

1) Initialization

   a) Determine an initial solution *v* with the greedy 3-phase procedure;

   b) Set $TC^* = TC(v)$, $v^* = v$ and $N = \{N^{(1)}, N^{(2)}, N^{(3)}, N^{(4)}, N^{(5)}\}$;

2) While *MaxTime* of CPU time is not reached, do:

   a) *Shaking.* Choose at random a neighborhood structure $N^{(t)}$ in *N*. Generate the best possible solution *v*' in $N^{(t)}(v)$;

   b) *Local search.* Apply tabu search (based on shift moves of distribution centers) with *v*' as initial solution; let *v*'' be the local optimum so obtained.

      If $TC(v'') < TC^*$, set $TC^* = TC(v'')$ and $v^* = v''$.

   c) *Move or not.* If $TC(v'') < TC(v)$, set $v = v''$ and $N = \{N^{(1)}, N^{(2)}, N^{(3)}, N^{(4)}, N^{(5)}\}$.

      Otherwise: if $|N| > 1$ and condition Δ is satisfied, remove $N^{(t)}$ from *N*;

      if $|N| = 1$, set $N = \{N^{(1)}, N^{(2)}, N^{(3)}, N^{(4)}, N^{(5)}\}$;

      Condition Δ: option VNS(1): $TC(v'') \geq TC(v)$;

      option VNS(2): $TC(v'') \geq TC(v^*)$.

Figure 4: Two VNS Versions for the Problem Considered

N°	Problem Instances	Cost Structures			
		r_{ps}	f_{psd}, f_{puw}	A_{js}	$H_{ps}(.)$
P1	3P-500D-60W-4U	X 1	X 1	X 1	Linear
P2	3P-500D -60W-4U	X 2	X 2	X 0.5	Linear
P3	3P -500D-60 W-4U	X 2	X 2	X 1	Linear
P4	3P-1000D-60W-4U	X 1	X 1	X 1	Linear
P5	3P-1000D -60W-4U	X 2	X 2	X 0.5	Linear
P6	3P-1000D- 60W-4U	X 2	X 2	X 1	Linear
P7	3P-500D-100W-6U	X 1	X 1	X 1	Linear
P8	3P-500D -100W-6U	X 2	X 2	X 0.5	Linear
P9	3P-500D -100W - 6U	X 2	X 2	X 1	Linear
P10	3P-1000D-100W-6U	X 1	X 1	X 1	Linear
P11	3P-1000D-100W-6U	X 2	X 2	X 0.5	Linear
P12	3P-1000D -100W-6U	X 2	X 2	X 1	Linear
P13	20P-500D -60W-4U	X 1	X 1	X 1	Linear
P14	20P-500D -60W- 4U	X 2	X 2	X 0.5	Linear
P15	20P-500D-60W-4U	X 2	X 2	X 1	Linear
P16	20P-1000D-60W-4U	X 1	X 1	X 1	Linear
P17	20P-1000D-60W-4U	X 2	X 2	X 0.5	Linear
P18	20P-1000D-60W-4U	X 2	X 2	X 1	Linear
P19	20P-500D-100W-6U	X 1	X 1	X 1	Linear
P20	20P-500D-100W-6U	X 2	X 2	X 0.5	Linear
P21	20P-500D-100-6U	X 2	X 2	X 1	Linear
P22	20P-1000D-100W-6U	X 1	X 1	X 1	Linear
P23	20P-1000D-100W-6U	X 2	X 2	X 0.5	Linear
P24	20P-1000D-100W-6U	X 2	X 2	X 1	Linear
P25	3P-500D-60W-4U	—	—	—	Concave
P26	3P-1000D-60W-4U	—	—	—	Concave
P27	3P-500D-100W-6U	—	—	—	Concave
P28	3P-1000D-100W-6U	—	—	—	Concave
P29	20P-500D-60W-4U	—	—	—	Concave
P30	20P-1000D-60W-4U	—	—	—	Concave
P31	20P-500D-100W-6U	—	—	—	Concave
P32	20P-1000D-100W-6U	—	—	—	Concave

Table 2: Test Problems Characteristics

As mentioned previously, our first task was the calibration of the several procedures used in the heuristics. Preliminary tests on various problem instances were performed to fix the algorithm parameters. These experiments showed that, for the realistic problems in our test sample, $(e_1; e_2; \theta; \kappa) = (5; 15; 35; 0.15|W(v)|)$ is a reasonable choice of parameters.

In addition, for each problem, the performance of the heuristics was compared to the optimal solution of program (2-13), obtained with OPL 5.1 and CPLEX-11. For the concave holding cost case, the problems were solved using the separable programming capabilities of OPL. The concave inventory holding cost functions $H_{ps}(X_{ps})$, $s \in S$, $p \in P_s$, were approximated with 3-segments piece-wise linear functions defined over the grid points $(0, \bar{X}_p/6, \bar{X}_p/2, \bar{X}_p)$, where $\bar{X}_p = \Sigma_d x_{pd}$ is the total network demand for product *p*. This is the more precise approximation we were able to solve with CPLEX.

The heuristics proposed were implemented in VB.Net 2005. The experiments were performed on a 32-bit 2 GHz Dual Core computer with 1 GB of RAM. However, we were not able to solve the concave holding cost instances with CPLEX on this machine. For this reason, a 64-bit 2 GHz Dual Core computer with 3 GB of RAM was used to solve the concave cost instances with CPLEX.

Evaluation Results

In Table 3, we present the results for the linear holding costs instances P1 to P24. For each instance, we give information associated with: the optimal solution (based on OPL-CPLEX), the INIT procedure, and the two proposed VNS heuristics, namely VNS(1) and VNS(2). All the CPU times are quoted in minutes. For CPLEX, we indicate the value *TC(opt)* of the optimal solution and the time needed to obtain it. For the INIT procedure, we indicate the percentage gap between *TC(opt)* and the value *TC(INIT)* of the best solution encountered among three runs of INIT, i.e. 100 *x* [*TC(INIT)* – *TC(opt)*] / *TC(opt)*. We also provide the time needed to perform three runs of INIT. For the VNS(1) heuristic, we indicate the percentage gap between *TC(opt)* and the value *TC(VNS(1))* of the best solution encountered within one hour. The same kind of information is given for VNS(2). Average values are given on the last line. For each VNS heuristic and each instance, we also indicate the time needed to obtain the best encountered solution within one hour.

On the average, for the linear case, we can observe that: (1) the exact method needs 375 minutes to find the optimal solution; (2) the INIT procedure is able to find a good solution (the gap is 1.84%) in much less than one minute; (3) the VNS(1) heuristic needs about 13 minutes to find a solution with a gap of 0.84%; the VNS(2) heuristic needs about 17 minutes to find a solution with a gap of 0.93%. Therefore, we conclude that all the proposed heuristics are relevant for the problem considered. Even if VNS(1) is on average better than VNS(2) according to the gap and the time needed to find its best solution within one hour, we note that VNS(2) obtains better solutions than VNS(1) on six instances (namely P5, P9, P13, P14, P21, P22). Thus, we cannot conclude that VNS(1) performs strictly better than VNS(2): both heuristics are relevant.

Instance	OPL/CPLEX		INIT		VNS(1)		VNS(2)	
	TC(opt)	Time	Gap(%)	Time	Gap(%)	Time	Gap(%)	Time
P1	16,685,600	280	3.55	0.0003	1.97	10	2.44	20
P2	16,240,400	104	3.01	0.0003	1.53	10	1.56	10
P3	19,301,500	196	6.84	0.0003	1.66	10	1.81	50
P4	33,123,400	140	2.60	0.0003	1.87	10	1.87	10
P5	32,013,900	90	2.39	0.0003	1.43	10	1.38	10
P6	38,022,500	150	2.41	0.05	1.85	10	2.55	10
P7	16,502,900	313	3.65	0.0003	1.59	10	1.76	10
P8	16,136,100	168	2.68	0.2667	0.90	10	0.95	10
P9	19,079,800	410	4.00	0.0003	1.83	20	1.34	10
P10	32,276,500	181	2.48	0.0003	1.11	20	1.16	20
P11	31,987,100	270	0.79	0.5	0.40	20	0.41	20
P12	37,349,839	313	1.79	0.0833	0.96	20	1.33	20
P13	119,597,000	402	0.796	0.0003	0.361	10	0.183	10
P14	119,215,000	382	0.374	0.0003	0.172	10	0.140	10
P15	138,570,000	390	0.691	0.0005	0.095	10	0.615	10
P16	229,708,000	465	0.687	0.05	0.333	10	0.385	10
P17	221,949,100	440	0.574	0.0003	0.241	10	0.294	10
P18	260,737,860	525	0.718	0.05	0.324	10	0.363	10
P19	111,625,000	488	0.917	0.0003	0.158	20	0.217	30
P20	110,081,000	480	0.389	0.6667	0.250	10	0.270	20
P21	127,453,000	486	0.648	0.8167	0.189	20	0.171	20
P22	222,640,000	628	0.826	0.0005	0.363	10	0.330	20
P23	219,182,000	794	0.541	0.0005	0.310	20	0.363	40
P24	254,052,000	914	0.898	0.0005	0.313	20	0.368	30
Average	**101,813,729.1**	**375.375**	**1.84**	**0.1037**	**0.84**	**13.3333**	**0.93**	**17.5**

Table 3: Results of the Linear Inventory Holding Costs Instances

In Table 4, based on Table 3, we present average results depending on the characteristics of the instances. We focus on four components: the number $|P|$ of products (3 or 20), the number $|D|$ of demand zones (500 or 1000), the number of centers ($|U|$ = 4 or 6 production-distribution centers, $|W|$ = 60 or 100 distribution centers), and the cost structures (see Table 1, where three types of cost structures, denoted by “X 1 - X 1 - X 1”, “X 2 - X 2 - X 0.5” and “X 2 - X 2 - X 1”, are presented). When one component is fixed, the three other components can vary, for example, in the line labeled “$|P|$ = 3”, we consider instances P1 to P12, which cover all values of $|D|$, $|U|$ and $|W|$, as well as all cost structures.

Characteristics	Exact method		INIT		VNS(1)		VNS(2)	
	TC(opt)	Time	Gap(%)	Time	Gap(%)	Time	Gap(%)	Time
$\vert P \vert = 3$	25,726,628.25	217.92	3.02	0.08	1.43	13.33	1.55	16.67
$\vert P \vert = 20$	177,900,830	532.83	0.67	0.13	0.26	13.33	0.31	18.33
$\vert D \vert = 500$	69,207,275	341.58	2.30	0.15	0.89	12.5	0.95	17.5
$\vert D \vert = 1000$	134,420,183.3	409.17	1.39	0.06	0.79	14.17	0.90	17.5
$(\vert U \vert;\vert W \vert) = (4; 60)$	103,763,688.3	297	2.05	0.01	0.99	10	1.13	14.17
$(\vert U \vert;\vert W \vert) = (6; 100)$	99,863,769.92	453.75	1.63	0.19	0.70	16.67	0.72	20.83
Cost structure: X 1 - X 1 - X 1	97,769,800	362.13	1.94%	0.01	0.97%	12.5	1.04%	16.25
Cost structure: X 2 - X 2 - X 0.5	95,850,575	341	1.34%	0.18	0.65%	12.5	0.67%	16.25
Cost structure: X 2 - X 2 - X 1	111,820,812.4	423	2.25%	0.13	0.90%	15	1.07%	20

Table 4: Average Results by Problem Type for the Linear Inventory Cost Instances

We can remark that: (1) the gap associated with VNS(1) is always smaller than the gap associated with VNS(2); (2) the VNS heuristics seem to be very efficient with a large number of products, because both gaps are close to 0.30 %; (3) the heuristics perform better with $|P| = 20$ rather than with $|P| = 3$, with $|D| = 1000$ rather than with $|D| = 500$, and with $(|U|;|W|) = (4;60)$ rather than with $(|U|;|W|) = (6;100)$; in other words, the heuristics perform better if there are more decision variables, which is very encouraging. When considering the cost structures, we can observe that our heuristics perform better when the variable costs are relatively high and the fixed costs relatively low.

The results obtained for the concave inventory holding cost instances are given in Table 5. We were not able to solve the three largest test problems with OPL-CPLEX, but all the problems were solved in reasonable time with our heuristics. VSN(1) performs particularly well with concave inventory holding costs. It outperforms the piece-wise linear approximation model for all the problems we were able to solve with CPLEX, and gives solutions that are 1.01% better on average. Again, the INIT procedure is able to find a good solution (the average gap with the best solution found is 2.8%) in much less than one minute. VNS(2) is not better than CPLEX for all problem instances, but for problems P25 and P29 it gives the best solution.

Instance	OPL/CPLEX		INIT		VNS(1)		VNS(2)	
	TC(CPLEX)	Time	Gap(%)	Time	Gap(%)	Time	Gap(%)	Time
P25	15,478,254	150	1.65	0.0003	-0.13	10	-0.20	10
P26	29,990,900	530	1.07	0.0005	-0.31	10	0.44	10
P27	15,317,097	93	2.23	0.3167	-0.01	30	0.04	10
P28	29,663,164	180	1.83	0.0003	-3.55	40	-0.10	30
P29	106,136,721	565	1.50	0.0003	-1.06	10	-1.43	10
P30	N/A	N/A	N/A	0.0003	N/A	30	N/A	20
P31	N/A	N/A	N/A	0.0003	N/A	60	N/A	60
P32	N/A	N/A	N/A	0.0003	N/A	60	N/A	50
Average	**39,317,227**	**303.6**	**1.66**	**0.0398**	**-1.01**	**31.25**	**-0.25**	**25**

Table 5: Results of the Concave Inventory Holding Costs Instances

Conclusion

This paper proposed efficient heuristics to solve an extension of the two-echelon multi-commodity CFLPSS considering direct shipments from manufacturing facilities and alternative facility configurations. This problem is not treated explicitly in the literature, even if it is encountered frequently in practice. Problem variants with linear and concave inventory holding costs were considered. Our computational results demonstrate the efficiency of the variable neighbourhood search (VNS) heuristics proposed. For linear inventory holding cost problems, near optimal solutions are obtained in a fraction of the time taken by CPLEX to reach the optimal solution. For concave inventory holding cost problems, the solutions provided by one of our heuristics (VNS-1) is always better than the one obtained by solving a piece-wise linear approximation of the model with OPL-CPLEX. Moreover, the largest problems considered could not be solved with CPLEX, but they were solved in reasonable time with our heuristics. All this indicates that variable neighbourhood search is a promising approach to solve more complex supply chain network design problems.

References

[1] Ahuja RK, Orlin JB, Pallattino S, Scaparra MP, Scutellà MG (2004) A Multi-Exchange Heuristic for the Single-Source Capacitated Facility Location Problem. Management Science **50**(6): 749–760

[2] Al-Sultan KS, Al-Fawzan MA (1999) A tabu search approach to uncapacitated facility location problem. Annals of Operations Research **86**: 91-103

[3] Amiri A (2006) Designing a distribution network in a supply chain system: Formulation and efficient solution procedure. European Journal of Operational Research **171**: 567–576

[4] Ballou RH (1992) Business Logistics Management, 3rd ed., Prentice Hall

[5] Balinski M (1961) Fixed-cost Transportation Problems. Naval Research Logistics Quarterly **8**: 41-54

[6] Barahona F, Chudak FA (2005) Near-optimal solutions to large-scale facility location problems. Discrete Optimization **2**: 35-50

[7] Beasley JE (1993) Lagrangean heuristics for location problems. European Journal of Operational Research **65**: 383-399

[8] Bornstein CT, Campêlo M (2004) An ADD/DROP Procedure for the Capacitated Plant Location Problem. Pesquisa Operacional **24**(1): 151-162

[9] Cooke J (2007) Weaving 2 supply chains together, CSCMP's Supply Chain Quarterly **3**: 34-38

[10] Crainic TG, Gendreau M, Soriano P, Toulouse M (1993) A tabu search procedure for multicommodity location/allocation with balancing requirements. Annals of Operations Research **41**: 359-383

[11] Delmaire H, Diaz JA, Fernandez E, Ortega M (1999) Reactive GRASP and Tabu Search Based Heuristics for the Single Source Capacitated Plant Location Problem. INFOR **37**(3): 194-225

[12] Elson DG (1972) Site Location via Mixed-Integer Programming, Operational Research Quarterly **23**(1): 31-43

[13] Fisher ML, Jaikumar R, Van Wassenhove L (1986) A Multiplier Adjustment Method for the Generalized Assignment Problem. Management Science **32**: 1095-1103

[14] Fleischmann B (1993) Designing Distribution Systems with Transport Economies of Scale. European Journal of Operational Research **70**: 31-42

[15] Gendron B, Potvin JY, Soriano P (2003) A Tabu Search with Slope Scaling for the Multicommodity Capacitated Location Problem with Balancing Requierments. Annals of Operations Research **122**: 193-217

[16] Geoffrion A, Graves G (1974) Multicommodity Distribution System Design by Benders Decomposition. Management Science **20**: 822-844

[17] Glover F (1986) Future paths for integer programming and linkage to artificial intelligence. Computers & Operations Research **13**: 533-549

[18] Glover F, Laguna M (1997) Tabu search. Kluwer Academic Publishers, Boston, MA

[19] Hansen P (1986) The steepest ascent mildest descent heuristic for combinatorial programming. In: Congress on Numerical Methods in Combinatorial Optimization, Capri, Italy

[20] Hindi KS, Basta T (1994) Computationally Efficient Solution of a Multiproduct, Two-Stage Distribution-Location Problem. The Journal of the Operational Research Society **45**(11): 1316-1323

[21] Hindi KS, Basta T, Pienakosz K (1998) Efficient Solution of a Multi-commodity, Two-stage Distribution Problem with Constraints on Assignment of Customers to Distribution Centers. International Transactions in Operational Research **5**(6): 519-527

[22] Jacobsen SK (1983) Heuristics for the capacitated plant location model. European Journal of Operational Research **12**: 253-261

[23] Kaufman L, Vanden Eede M, Hansen P (1997) A Plant and Warehouse Location Problem. Operational Research Quarterly **28**(3): 547-554

[24] Kelly DL, Khumawala BM (1982) Capacitated warehouse location with concave costs. Journal of the Operational Research Society **33**: 817-826

[25] Klose A (1999) An LP-Based Heuristic for Two-Stage Capacitated Facility Location Problems. The Journal of the Operational Research Society **50**(2): 157-166

[26] Klose A (2000) A Lagrangean relax-and-cut approach for two-stage capacitated facility location problem. European Journal of Operational Research **126**: 408-421

[27] Klose A, Drexl A (2005) Facility location models for distribution system design. European Journal of Operational Research **162**: 4-29

[28] Kratica J, Tosic D, Filipovic V, Ljubic I (2001) Solving the Simple Plant Location Problem by Genetic Algorithm. RAIRO Operations Research **85**: 127-142

[29] Kuenh, AA, Hamberger MJ (1963) A Heuristic Program for Locating Warehouses. Management Science **9**(4): 643-666

[30] Manne A (1964) Plant Location under Economies-of-scale: Decentralization and Computation. Management Science **11**: 213-235

[31] Martel A, Vankatadri U (1999) Optimizing Supply Network Structures Under Economies of Scale. IEPM Conference Proceedings, Glasgow, Book 1, pp 56-65

[32] Martel A (2003) Généralisations du modèle d’optimisation d’un réseau de ravitaillement. Document de formation DF-2.4.2, CIRRELT, Université Laval, Canada

[33] Martel A (2005) The Design of Production-Distribution Networks: a Mathematical Programming Approach, In: Geunes J, Pardalos PM (eds) Supply Chain Optimization, pp 265-306, Springer

[34] Michel L, Van Hentenryck P (2004) A simple tabu search for warehouse location. European Journal of Operational Research **157**: 576–591

[35] Mladenovic N, Hansen P (1997) Variable Neighborhood Search. Computers & Operations Research **24**: 1097-1100

[36] Paquet M, Martel A, Desaulniers G (2004) Including Technology Selection Decisions in Manufacturing Network Design Models. International Journal of Computer Integrated Manufacturing **17**: 117-125

[37] Pirkul H, Jayaraman V (1996) Production, Transportation, and Distribution Planning in a Multi-Commodity Tri-Echelon System. Transportation Science **30**(4): 291-302

[38] Pirkul H, Jayaraman V (1998) Multi-Commodity, Multi-Plant, Capacitated Facility Location Problem: Formulation and Efficient Heuristic Solution. Computers and Operations Research **25**(10): 869-878

[39] ReVelle C, Eiselt H, Daskin M (2008) A bibliography for some fundamental problem categories in discrete location science. European Journal of Operational Research **184**: 817-848

[40] Soland RM (1974) Optimal plant location with concave costs. Operations Research **22**: 373-382

[41] Sörensen K (2002) A Framework for Robust and Flexible Optimisation using Metaheuristics with Applications in Supply Chain Design, Ph.D. Thesis, Faculteit Toegepaste Economische Wentenschappen, Universiteit Antwerpen, Belgium

[42] Talbi EG (2002) A Taxonomy of Hybrid Metaheuristics. Journal of Heuristic **8**: 541-564

[43] Zhang J, Chen B, Ye Y (2005) A multi-exchange local search algorithm for the capacitated facility location problem. Mathematics of Operations Research **30**(2): 389-403