7^e *Conférence Francophone de MOdélisation et SIMulation - MOSIM'08 - du 31 mars au 2 avril* 2008 *- Paris - France*
≪Modélisation, Optimisation et Simulation des Systèmes : Communication, Coopération et Coordination≫

A LINEAR RELAXATION-BASED HEURISTIC FOR LOGISTICS NETWORK DESIGN

Phuong Nga THANH, Olivier PÉTON

École des Mines de Nantes, IRCCyN
4 rue Alfred Kastler
B.P. 20722, 44307 Nantes Cedex 3
{phuong-nga.thanh, olivier.peton}@emn.fr

Nathalie BOSTEL

IUT de Saint-Nazaire, IRCCyN
58 rue Michel Ange
B.P. 420, 44606 Saint-Nazaire Cedex
Nathalie.Bostel@univ-nantes.fr

ABSTRACT: *We address the problem of designing and planning a multi-period, multi-echelon, multi-commodity logistics network with deterministic demands. This consists of making strategic and tactical decisions: opening, closing or expansion of facilities, supplier selection and definition of the product flows. We use a heuristic approach based on the linear relaxation of the original mixed integer linear problem (MILP). The main idea is to fix as many binary variables as possible with the linear relaxation coupled with several rounding procedures for key variables. The number of binary variables in the resulting MILP is small enough to enable it to be solved with a commercial solver. We compare the computational time and the quality of the solution obtained by the heuristic and a commercial solver.*

KEYWORDS: *logistics network design, MILP, heuristic, linear relaxation, rounding procedures*

INTRODUCTION

Logistics network design is concerned with strategic and tactical decisions such as the location of warehouses and production plants, allocation of warehouses to production plants, allocation of customer demand points to warehouses, selection of suppliers and definition of the product flows between entities. An optimal configuration must be able to deliver the products to the customers at the lowest cost while satisfying the service level requirements (Cheong Lee Fong (2005)).

From the manager's point of view, modelling and optimising a strategic logistics network is useful for evaluating different scenarios. Mathematical models may be used as simulation tools with what-if scenarios. Should we locate the new warehouse at point A instead of point B? Should we expand the existing plant or build a brand new one at another location? Mathematical models may be considered as decision support systems only if they are able to propose very good solutions within an acceptable amount of time. Trying to get reduced computational times when solving a strategic and tactical problem may seem too demanding, but necessary. The algorithms can be used repeatedly with different configurations, so that it is important to limit the computational times and favour the approximation methods.

Another factor supports the use of heuristic approaches instead of exact methods. When it is impossible to have a precise estimation of the costs involved in the model, calculating optimal solutions is not necessary. Good robust solutions are often sufficient.

We consider the problem of designing and planning logistics network over a strategic horizon. This can be described as a facility location problem, with multiple facilities, periods and commodities and additional constraints. The mathematical formulation and some preliminary numerical results obtained with a commercial solver are presented in Thanh et al. (2007). The aims of the present papers is to propose a new heuristic based on the LP-rounding rules, which gives very good approximations of the optimal solutions while keeping the computational time below a limit of three hours. We also report solutions obtained with a commercial solver as a benchmark for evaluating the heuristic.

The paper is organised as follows. In section 1, we recall the main concepts of supply chain design and planning and introduce the optimisation problem. In section 2, we present the linear relaxation-based heuristic. The computational experiments are reported in section 3.

1 A DYNAMIC MODEL FOR LOGISTICS NETWORK DESIGN

1.1 A brief review of the literature

The multi-period planning of a supply chain is an NP-hard problem that has been addressed by many authors in the recent and abundant literature. There is such a large variety of enterprise logistics networks that it makes no sense to build a completely generic model that would fit any company. The most recent models include many additional features, with the idea of reflecting some real cases or of focusing on some particular aspects of the location problems. However, no published paper has simultaneously taken into account all the possible features. Among the most widespread characteristics in the recent models, we can list:

- supply chain with multiple echelons and multiple products or families of products,
- dynamic models where the data and variables may change at every period,
- complex product flow: exchange of products between plants or warehouses, direct deliveries to some customers, reverse logistics, re-manufacturing, etc.
- variety of constraints: competition or budget constraints, etc.
- complex structure of the costs: fixed and variable parts, linear or nonlinear costs,
- hybrid strategic/tactical models with inventories: average, safety or cyclic inventories.

Dias et al. (2006) worked on the re-engineering of a network composed of facilities and customers. The authors suppose that facilities can be open, closed and reopen more than once during the planning horizon. The model is solved by primal-dual heuristics. Melo et al. (2006) aimed at relocating the network with expansion/reduction capacity scenarios. Capacity can be exchanged between an existing facility and a new one, or between two existing facilities under some conditions. Vila et al. (2006) proposed a dynamic model in a much more specialised context. They considered an application in the lumber industry, but their model can be applied to other sectors.

The methods for solving logistics network planning and design use the classical tools of operations research. The exact methods include branch-and-bound, Benders decomposition or the use of a commercial solver (see Canel et al. (2001), Hamer-Lavoie and Cordeau (2004), Melo et al. (2006), Martel (2005)).

Most of the metaheuristic approaches are proposed for basic models like static models for simple network planning. Filho and Galvão (2001) used tabu search to solve a concentrator location problem. Syam (2002) mixed simulated annealing and Lagrangean relaxation to solve a 3-echelon problem. Other authors have used genetic algorithms (Gong et al. (1997), Jaramillo et al. (2002)).

Surprisingly enough, very few heuristic methods are reported in the specialised literature, although they seem to be efficient for the most complex problems. The most common heuristics use some "add" and "drop" procedures (Saldanha da Gama and Captivo (1998)), or rely on primal-dual methods (Hinojosa et al. (2000), Dias et al. (2006)) or Lagrangean relaxation (Pirkul and Jayaraman (1998)). Heuristic methods based on neighbourhoods and local improvements are rarely used because it is often difficult to define efficient neighbourhoods in complex models.

LP-rounding heuristics belong to the more general class of approximation algorithms for combinatorial optimisation problems. Extensive information about approximation heuristics can be found in the monograph of Vazirani (2001). Despite their poor formal guarantee of performance, LP-rounding heuristics are known to yield good lower bounds for some assignment or location problems (see Benders and van Nunen (1982). They have been applied to the general assignment problem (see Trick (1992) or French and Wilson (2007)) or lot-sizing problems (see Hardin et al. (2007)). In the field of facility location or logistics network planning, linear relaxation-based heuristics have been used for the single facility location and more complex models. Among the recent references, we can cite Levi et al. (2004) for example. Velásquez et al. (2007) propose a heuristic based on the same ideas as the present work. Due to some budget constraints, their heuristic does not always yield feasible solutions.

1.2 Description of the problem under consideration

The logistics network considered is composed of four layers, like the one depicted in Figure 1. The first layer consists of a set of potential suppliers that provide the company with raw materials. Production steps composed of plants make up the second layer. Product storage and distribution steps, performed by warehouses, are the main constituents of the third layer. Finally, the fourth layer is composed of customers. We adopt a flexible network structure: products can be transferred between plants or delivered directly from plants to important customers.

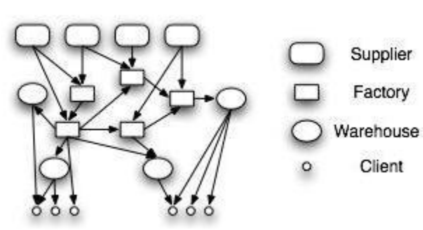

Figure 1: *Example of the general network*

Every product has its own bill of materials. For every plant or warehouse, we have classical capacity constraints and add lower and upper limits to the rate of utilisation (e.g. running a plant at 1% or at 100% of its capacity is not authorised). We also model the possibility of increasing the capacity of an existing facility by building some physical extensions, called capacity options.

Our purpose is to make strategic decisions over a multi-period planning horizon. These decisions concern the selection of suppliers, opening or closing of facilities, capacity planning for open facilities and production and distribution management. Production management consists of allocating manufactured products to open plants and of managing subcontracted products (we suppose that the company can subcontract a part of its production). Distribution management consists of allocating finished products and customers to open or hired warehouses. The allocation of products and customers is decided by the quantity of transferred products between entities. Our model does not take into consideration the constraints of single-sourcing in order to avoid introducing new binary variables and to focus on the location aspect. Inventory management is also included as it influences strategic decisions such as capacity planning even though it is a tactical decision. However, we suppose that the inventories are held only in warehouses, not in plants.

For purposes of simplicity, we do not present the extensive mathematical model in this paper. The interested reader should refer to Thanh et al. (2007).

The mathematical model handles four types of binary decision variables that represent:

- The status of each entity, at each period (named x),
- The status of each capacity option, at each facility and at each period,
- The supplier selection for each raw material, at each period,
- The discount according to each supplier, at each period.

Depending on the context, the same variables x stand for suppliers, plants and warehouses. A supplier is said to be active if it delivers at least one raw material and a plant/warehouse is active if it is open in the corresponding period.

The MILP also includes continuous decision variables that model the product flows along the network:

- Quantity of transferred products between two entities in the network, at each period,
- Quantity of sub-contracted products for each plant, at each period,
- Quantity of manufactured products in each plant, at each period,
- Quantity of stored products in each warehouse at the beginning of each period.

The cost structure consists of two parts: fixed costs and variable costs. All these costs are linear and time-independent. The objective function is the sum of nine components: supplier selection fixed costs, opening facility fixed costs, closing facility fixed costs, capacity option costs, operating facility fixed costs, production variable costs, storage variable costs, distribution variable costs and transportation variable costs.

The set of linear constraints can be divided into four categories:

- *Demand satisfaction constraints:* these constraints ensure that the demands at each level of the network must be satisfied. We notice that for warehouses, the seasonal inventory is taken into consideration. The subcontracted products are also counted for each plant.
- *Capacity limitation constraints:* these constraints require that plants and warehouses cannot function under its minimum rate of utilization and cannot exceed the maximum rate of utilisation of its capacity.
- *Logical constraints:* these constraints ensure the coherence of the model. For example, suppliers can deliver a raw material if and only if they are selected for this raw material; their delivery cannot exceed their capacity; a plant/warehouse can only deliver products that it can manufacture/store ...
- *Integrity and non-negativity constraints:* these constraints focus on the nature of variables.

The mathematical formulation is an MILP. Most of the binary variables are related to the location of facilities. As shown in Thanh et al. (2007), variables x associated with plants and warehouses play a critical role in the model. These location variables are the core of the problem and have a direct influence on a large part of the cost function.

2 LINEAR RELAXATION AND ROUNDING HEURISTIC

The principle of the heuristic arises from the well known fact about facility location problems: the resolution of the MILP linear relaxation yields good lower bound and sets most of the binary variables to 0 or 1 within a short period of time. Robert et al. (2006) solved a close strategic supply chain planning problem and observed that more than 60% of the binary variables were directly set to 0 or 1 by the linear relaxation.

The main idea of the heuristic is to fix as many binary variables as possible with the linear relaxation so that solving the resulting problem with an exact method becomes easier.

In the formulation, x variables play a key role, especially those related to the plant and warehouse location. Hence, the level of fixed variables x makes an important contribution to the success of the heuristic.

There are three ways for assigning the value 0 or 1 to a binary variable:

- the value is set by the LP relaxation,
- the value is set by an exact method,
- the value is set by rounding rules.

The heuristic uses these three approaches and aims to find a trade-off between the optimality gap and the computational time.

2.1 Description of the heuristic

We can summarise the heuristic by the three following steps:

- ***Step 1:*** Solve the linear relaxation of the original problem, denoted **LP**, and fix all decision variables whose relaxed values have a strong tendency towards 0 or 1 (determined by two thresholds) (section 2.2.1). A new **LP** is then created and solved at the following iteration. Continue until no more binary variables can be set to 0 or 1. The resulting LP may be infeasible. In this case, a correction procedure is applied (section 2.2.2).
- ***Step 2:*** This step aims to increase the number of fixed binary variables x in order to reach a target level s. The value of s makes the equilibrium between the execution time and the solution quality. If s is large, the step 3 is faster but usually we obtain a worse solution. For fixing more location variables, we adopt distinct procedures for warehouse and plant location. The warehouse variables are determined by a rounding rule with two new thresholds. When no new binary variable is fixed, the largest fractional variable is rounded up to 1. The plant variables are determined through a greedy heuristic. These procedures are detailed in sections 2.2.3 and 2.2.4.
- ***Step 3:*** This step consists of solving the resulting problem with an exact method. If the resulting problem is infeasible, a correction procedure is then needed.

2.2 Some key procedures

2.2.1 Fixing binary variables (Step 1 and 2)

We associate every binary variable with a lower bound (initially set to 0) and an upper bound (initially set to 1). Fixing binary variables is performed by modifying these bounds (for example setting an upper bound to 0 forces the corresponding variable to equal 0). On the other hand, we can relax fixed binary variables by setting their bounds back to their initial values.

The strong tendency towards 0 or 1 is detected by using four thresholds ϵ_0 and ϵ_1 in Step 1, γ_0 and γ_1 in Step 2. When a relaxed variable is lower than ϵ_0 (γ_0), the variable is set to 0. When a relaxed variable is larger than ϵ_1 (γ_1), the variable is set to 1. Decreasing one variable to 0 may provoke infeasibility, thus it is recommended to set ϵ_0 (γ_0) closer to 0 than ϵ_1 (γ_1) to 1.

2.2.2 Correction procedure (Steps 1 and 3)

The correction procedure is divided into two phases. As a precaution, we reduce ϵ_0 and increase ϵ_1 in order to avoid getting the same solution in the next iterations and to decrease the probability of obtaining an infeasible solution. As explained previously, decreasing one variable to 0 may provoke infeasibility, hence ϵ_0 is submitted to a stronger modification than ϵ_1. We multiply ϵ_0 by 0.5 while ϵ_1 is multiplied by 0.9.

Then we detect the constraint violation that caused the infeasibility and relax the corresponding variables. If the infeasibility is caused by the lack of capacity at suppliers, all fixed binary variables concerning the selection of suppliers are relaxed. In the same way, if the cause is the lack of capacity at plants or warehouses, the corresponding fixed binary variables are relaxed. Regarding the plant and warehouse variables, we relax them in two stages: first the capacity option variables are relaxed and, whenever needed, we relax the location variables (variables x). This strategy minimizes the number of location binary variables relaxed.

From the theoretical point of view, infeasibility may also be caused by a violated logical constraint but we have never observed this in practice.

2.2.3 Setting binary variables for warehouse location (Step 2)

Reaching the desired level s of fixed location variables before Step 3 is a key point of the heuristic because it enables Step 3 to run within a reasonable computational time. If this level has not been reached after Step 1, it is necessary to increase it by fixing a larger number of variables.

We use two rules for setting binary variables for warehouse location:

- rounding the variables that have a tendency towards 0 or 1, with new parameters γ_0 and γ_1 such that $0 \leq \epsilon_0 \leq \gamma_0$ and $\gamma_1 \leq \epsilon_1 \leq 1$

- rounding the largest fractional value to 1.

These rules are depicted in Algorithm 1.

Algorithm 1 Set binary variables for warehouse location

Arguments:
γ_0: pre-defined threshold to set a binary variable to 0
γ_1: pre-defined threshold to set a binary variable to 1
r: level of fixed warehouse location variables
s: desired level of fixed warehouse location variables

```
while r < s do
  for every warehouse location variable x_i do
    Call the fixing procedure with γ0 and γ1 as
    arguments
  end for

  if no new variable is set to 0 or 1 then
    Select the largest fractional value x̄_i
    x̄_i ← 1
    x_j ← 0    ∀j ≠ i
    Solve LP
    if LP is infeasible then
      Relax all x_j that were fixed to 0
      Solve LP
    end if
  end if
end while
```

As we want to fix more binary variables than in step 1, γ_0 and γ_1 must be farther from 0 and 1 than ϵ_0 and ϵ_1. However, accepting too loose thresholds γ_0 and γ_1 is likely to yield very bad solutions. Finding a good tradeoff between a high level of fixed values and an acceptable solution is discussed in section 3.2.2.

The second part of Algorithm 1 gives priority to the variables that have a tendency towards 1. The variable with the largest fractional value is selected and set to 1 whereas the others are set to 0. If the obtained solution is infeasible due to the lack of warehouse capacity, we relax all binary variables that were fixed to 0 at the current iteration. The advantage of this rule is that, in the best scenario, we can fix all warehouse location variables in one iteration. However, in practice, this scenario very seldom occurs.

2.2.4 Setting binary variables for plant location (used in Step 2)

We cannot apply Algorithm 1 to the plant variables because the model includes the possibility of subcontracting a part of the production. Moreover, the capacity of subcontracted plants is supposed to be unlimited. Hence, Algorithm 1 can always find a feasible solution for plant capacities with no guarantee of minimizing the objective function.

We resort to a greedy heuristic where a feasible solution is built iteratively by rounding the greatest fractional value up to 1. The method is detailed in Algorithm 2.

Algorithm 2 Set binary variables for plant location

Parameters:
O_P: objective function value of the previous iteration
O_I: objective function value of the current iteration
δ_O: difference in the objective function values

```
for every plant location variable x_i do
  x_i ← 0
end for
Solve LP and Save O_I
Relax all variables x_i that were fixed to 0
repeat
  O_P = O_I
  Select the largest fractional value x_i among
  variables for plant location
  x_i ← 1
  x_j ← 0    ∀j ≠ i
  Solve LP and Save O_I
  δ_O = O_I − O_P

  if δ_O ≤ 0 then
    Relax all variables x_j that were fixed to 0
    Solve LP
  end if
until δ_O ≥ 0
```

The advantage of this procedure is that it sets all plant location variables to 0 or 1. On the other hand, it takes the value of the objective function into account and does not only try to get a feasible solution.

3 COMPUTATIONAL TESTS

We evaluate the heuristic by solving the test instances with the same characteristics as those presented in Thanh et al. (2007). In that paper, the numerical experiments highlighted the large variability of the computational times for different instances with the same characteristics. For instances of limited size (about 1000 binary decision variables), it was always possible to obtain optimal solutions in less than three hours on a personal computer. Unfortunately, for larger instances, obtaining the optimal solution generally required hours of calculation. This makes the problem intractable in practice.

3.1 Data description

We consider ten families of instances, which are divided into two sets: five families of small instances (denoted in order of increasing size from S1 to S5) and five families of medium instances (denoted in order of increasing size from M1 to M5). In the small instances, the number of entities (suppliers, plants, warehouses, customers) varies from 65 to 120 and there are from 10 to 18 products. For medium instances, the number of entities varies in the interval $[144, 205]$ and there are from 18 to 26 products. The planning horizon is fixed to five periods. For each family, we generate ten instances. This limits the impact of particular instances. The detailed characteristics of the data sets are described in Table 1 and Table 2.

Family	**S1**	**S2**	**S3**	**S4**	**S5**
Suppliers	5	10	10	12	12
Plants	5	9	9	9	9
Warehouses	5	9	9	9	9
Customers	50	70	70	90	90
Products	10	10	14	14	18
Binary variables	375	710	810	900	1020

Table 1: *Parameters for small instances*

Family	**M1**	**M2**	**M3**	**M4**	**M5**
Suppliers	12	12	14	14	15
Plants	11	11	11	11	15
Warehouses	11	11	11	11	15
Customers	110	110	130	130	160
Products	18	22	22	26	26
Binary variables	1100	1220	1350	1490	1725

Table 2: *Parameters for medium instances*

We report the total number of binary variables, but only half of them are location variables. The others concern the selection of suppliers.

3.2 Numerical results

The heuristic is implemented in C++ while the linear relaxations and the resulting problem in Step 3 are solved with the MILP solver of Xpress-MP (release 2005). In addition, to evaluate the quality of the heuristic, each instance is solved separately with Xpress-MP. The experiments are performed on a Pentium IV, 3.2 GHz processor with 1 GHz of RAM.

3.2.1 Numerical results with the MILP solver

The numerical results obtained with the MILP solver are presented in Table 3 and Table 4. We use them as a benchmark for the heuristic. Best and Worst represent the shortest and the longest time (in seconds) among the ten instances of each family.

Family	**S1**	**S2**	**S3**	**S4**	**S5**
Best	5	38	237	274	748
Average	24	144	1099	4500	6840
Worst	53	638	1954	10920	11100

Table 3: *Computational time (in seconds) with the MILP solver for small instances*

We observe that the longest CPU time for these instances is about three hours. Thus, we choose this computational times as a limit for the medium instances. The obtained optimality gap is calculated as follows:

$$Optimality\ gap = \frac{Best\ solution\ -\ Lower\ bound}{Lower\ bound} \times 100.$$

Family	**M1**	**M2**	**M3**	**M4**	**M5**
Best	0	0	0	0.6	1.1
Average	0.5	0.7	0.8	1	2.96
Worst	1.4	2.1	2	2.8	5

Table 4: *Optimality gap (%) after 3h of computation with the MILP solver for medium instances*

We decide to report the optimality gap rather than the time needed to get the optimal solution because this is too long for instances of the last five families.

As described in Thanh et al. (2007), we observe a strong variability in the results for instances of the same family. This can be explained by the nature of our data sets. Three criteria influence the difficulty of the problem: the size of the logistics network, the complexity of the supply chain structure and the complexity of the bill of materials. For instances of the same family, we have the same size of network but we generate instances with different levels of difficulty for the other two criteria.

3.2.2 Tuning the heuristic parameter values

As stated previously, the heuristic is influenced by five parameters:

- ϵ_0, γ_0: thresholds to fix binary variables to 0,
- ϵ_1, γ_1: thresholds to fix binary variables to 1,
- s: desired level of fixed location variables.

These parameters can lie in the interval $[0, 1]$ with $\epsilon_0 < \epsilon_1$ and $\gamma_0 < \gamma_1$. With the aim of tuning these parameter values, we carried out some experiments with two instances of each family. The following set seems to achieve a good trade-off between the computational time and the quality of the solution:

- $\epsilon_0 = 0.05$, $\gamma_0 = 0.1$,
- $\epsilon_1 = 0.9$, $\gamma_1 = 0.8$,
- $s = 0.6$ for the families S1 to S5 and $s = 0.8$ for the families M1 to M5

The numerical results reported in sections 3.2.3 and 3.2.4 are obtained with these parameter values.

We found out that among these five parameters, s has the strongest influence on the performance of the heuristic because the other four parameters can be adjusted when we obtain an infeasible solution. In order to examine the stability of the heuristic with respect to s, we kept fixed values for ϵ_0, ϵ_1, γ_0, γ_1 and tested two other scenarios for s:

- First scenario: $s = 0.5$ for the families S1 to S5 and $s = 0.65$ for the families M1 to M5
- Second scenario: $s = 0.75$ for the families S1 to S5 and $s = 0.85$ for the families M1 to M5

Fortunately, we observed only a slight difference between the three scenarios. The maximal difference is 0.8% for the objective function value and 10% for the execution time. From our experiments, the heuristic seems to be robust with respect to reasonable changes in parameters.

3.2.3 Heuristic numerical results

We report the results of the heuristic in Table 5 and Table 6. The first part of these tables represents the CPU times (in seconds) of each family. For better describing how this time is used, we report also the ratio (in percentage) of the running time of first two steps with respect to the total time of the heuristic. The last showed information is the number of iterations and the number of infeasible solutions occurring in the first step. These quantities are the average of ten instances in each family.

Family	**S1**	**S2**	**S3**	**S4**	**S5**
CPU times (in seconds)					
Best	5	14	31	60	99
Average	**7**	**19**	**61**	**147**	**159**
Worst	13	25	81	324	356
Ratio (%) of Step 1 and Step 2 execution times					
Average	64	60	59	63	57
Infeasible solutions in Step 1 (on average)					
Iterations	3.6	3.7	4.8	7.7	8.5
Infeasible	0.4	0	0.4	2.6	2.8

Table 5: *Small instances - Results with the heuristic*

Family	**M1**	**M2**	**M3**	**M4**	**M5**
CPU times (in seconds)					
Best	238	254	346	486	1671
Average	**368**	**646**	**1774**	**1470**	**3675**
Worst	658	1331	11111	2028	6330
Ratio (%) of Step 1 and Step 2 execution times					
Average	76	70	59	74	64
Infeasible solutions in Step 1 (on average)					
Iterations	7.1	7.6	7.5	8.5	8.9
Infeasible	2.2	2.2	2.3	2	2

Table 6: *Medium instances - Results with the heuristic*

We observe a strong increase in the CPU times for the last families. Nevertheless, the heuristic requires on average about one hour for the largest instances. On the other hand, as with the MILP solver, we obtain a high variability within the same family. The largest difference is recorded for the family M3.

We suppose that the heuristic needs less than three hours so that this time is taken as the limit for its running time. In practice, the heuristic stopped after less than one hour in more than 90% of the instances and after less than ten minutes in more than 60% of them.

The execution time tends to get longer when the instance size grows, but the running time is larger for family M3 than for family M4. This abnormal result is due to one difficult instance of M3 that requires more than three hours.

The largest part of the computational time is dedicated to Steps 1 and 2. On average, these steps represent 60% of the total time of the heuristic.

We report only the number of infeasible solutions obtained in Step 1 because none is recorded in Step 3. We note that, for the last seven families, the ratio of infeasible solutions is higher, around 30% of the total number of iterations. Finding feasible solutions for the first three families seems to be much easier. The more complex the problem, the more difficult it is to find a feasible solution.

3.2.4 Comparison between the MILP solver and the heuristic solutions

In this section, we present the comparison between the MILP solver and the heuristic solutions with respect to the execution time and the value of the objective function (Table 7 and Table 8).

The relative gap in computational time Δ_T and the relative deviation of the value of the objective function Δ_O are defined as follows:

$$\Delta_T = \frac{T_H - T_S}{T_S} \times 100, \qquad \Delta_O = \frac{O_H - O_S}{O_S} \times 100.$$

where
T_H: computational time of the heuristic,
T_S: computational time of the MILP solver,
O_H: objective function value of the heuristic,
O_S: objective function value of the MILP.

Family	S1	S2	S3	S4	S5
Δ_T *(%)*					
Best	-86	-97	-96	-99	-99
Average	**-43**	**-75**	**-91**	**-93**	**-95**
Worst	30	-56	-74	-78	-80
Δ_O *(%)*					
Best	0	0.7	0.2	0.04	0.04
Average	**3.32**	**3.67**	**1.7**	**2.7**	**1.3**
Worst	10.6	10.5	8.3	7.7	3.7

Table 7: *Small instances - Comparison of computational time and value of the objective function*

Family	M1	M2	M3	M4	M5
Δ_T *(%)*					
Best	-97	-97	-96	-95	-85
Average	**-96**	**-94**	**-83**	**-86**	**-68**
Worst	-94	-87	0.4	-76	-55
Δ_O *(%)*					
Best	-0.01	0.03	0.09	-0.9	-5
Average	**2.8**	**0.6**	**1.09**	**0.5**	**-2.5**
Worst	6.7	1.8	3.4	2.2	0.98

Table 8: *Medium instances - Comparison of computational time and value of the objective function*

Table 7 and Table 8 report the best, the average and the worst value of Δ_T and Δ_O for each family. The average gap in execution time varies between -43% and -96% and the average of the relative deviation of the objective function value varies from 3.67% to -2.5%. We observe a large variability between the ten families.

In addition, the results are different from one family to another. For the first two families (S1 and S2), Δ_T is lower than the others. Unfortunately these families also have the highest relative deviation of value of the objective function. This confirms that the heuristic is more efficient for large instances than for small ones. Nevertheless, we suppose that the resolution of small instances with the heuristic is not so useful because they can be solved with the commercial solver within a reasonable time. We turn our attention to the results of larger instances. From S3 to M2, the gain in time is very high (more than 90%) while we lose around 2% to 3% on Δ_O. For the last two families, Δ_T is lower than for the preceding families but Δ_O is also lower.

For all test instances, the heuristic runs faster (this is represented by the negative value of Δ_T) except for two instances (one instance of S1 and another of M3). Concerning the instance of the first family, since its execution time is negligible, this overtaking is not important. As regards the instance of M3, it seems that the desired level s is not large enough to help the heuristic to finish within three hours.

Note that the instances of the family M5 are better solved by the heuristic than by the solver (stopped after three hours). This confirms the efficiency of the heuristic for large instances.

In order to evaluate the performance of the heuristic, we measure the running time T^* of each instance. The MILP solver is run for exactly the same duration T^*. The values of the objective function of the heuristic and the MILP solver are then compared. Table 9 and Table 10 show the relative deviation ρ_O of the heuristic solution with respect to the MILP solution. ρ_O is defined as follows:

$$\rho_O = \frac{O_H - O_S}{O_S} \times 100.$$

where
O_H: objective function value with the heuristic,
O_S: objective function value with the MILP solver.

Family	S1	S2	S3	S4	S5
ρ_O *(%)*					
Best	-24	-41.11	-47.6	-50.67	-57.34
Average	**-8.7**	**-26**	**-28.55**	**-21.78**	**-30.77**
Worst	8.38	-4.21	-7.2	8.49	0.805
Instances for which the heuristic outperforms the solver (/10)					
Number	5	10	10	7	9

Table 9: *Small instances - Comparison of the objective function value with the same allocated computational time*

Family	M1	M2	M3	M4	M5
ρ_O *(%)*					
Best	-51.32	-59	-57	-60	-59.35
Average	**-29.2**	**-44.5**	**-34.11**	**-32.45**	**-37.84**
Worst	2.8	0.08	-0.02	0.08	-7.55
Instances for which the heuristic outperforms the solver (/10)					
Number	9	9	10	9	10

Table 10: *Medium instances - Comparison of the objective function value with the same allocated computational time*

On average the heuristic outperforms the MILP solver. The average relative deviation varies from -8.7% to -44.5%. Nevertheless, for six families we have at least one instance where the MILP solver is more efficient than the heuristic. This is represented by the positive worst value of ρ_O and by the number of instances for which the heuristic outperforms the MILP solver (less than 10).

The worst results are recorded for families S1 and S4. For these two families the worst relative deviation is around 8% and at least two instances of each family are better solved by the MILP solver. On the other hand, the results for medium families are more encouraging. In each family, we have at least nine smaller values of the objective function with the heuristic. Even when the MILP solution is better the relative deviation is negligible.

4 CONCLUSION AND FURTHER RESEARCH

We have proposed a heuristic based on LP-rounding rules for the design and strategic planning of complex logistics networks. The method relies on successive relaxations of the original mixed integer linear program. At each iteration, some binary variables are set to 0 or 1, either directly by the linear relaxation or by some rounding procedures. The last step of the heuristic consists of solving exactly the resulting problem by a solver.

The proposed LP-rounding heuristic has some new performance points. First of all, it ensures a feasible solution for every instance thanks to the correction procedure. We try to develop an efficient procedure by relaxing as few fixed variables as possible. On the other hand, the heuristic selects a subset of critical binary variables corresponding to location decisions and neglects less important decisions. We present two additional fixing rules for important variables: largest fractional value and a greedy rule. These procedures are based on the fractional value of binary variables and do not take probabilistic elements into consideration. The use of an exact method in the last step of the heuristic is also a new point. Indeed, after the first two steps, most of the binary variables are fixed. The problem size is reduced so that the execution time of the exact method may also be reduced.

The numerical results confirm the efficiency of the heuristic, especially for medium-sized instances. The observed average reduction of computational time is more than 80% while we lose only 1.5% on average of the value of the objective function. The gain is maximal for medium-sized instances. We observe that for the largest instances (family M5), the resulting MILP at Step 3 is still large, and the time reduction of the heuristic is only 68% on average. However, the objective function of the exact method is improved by 2.5%.

There are several ways of improving the heuristic. Although we observe some robustness of the result with respect to the parameters mentioned in section 3, we believe that having a good comprehension of their individual contribution would benefit the whole algorithm. The present heuristic can be embedded into a more complex method or be used for other types of MILP problem. One important issue is to identify the decision variables that play a critical role in the quality of solutions.

REFERENCES

Benders J.F. and van Nunen J.A.E.E. (1982). A property of assignment type mixed integer linear programming problems. O.R. Letters, **2**, 47-52.

Canel C., Khumawala B.M., Law J. and Loh A. (2001). An algorithm for the capacitated, multi-commodity multi-period facility location problem. Computers & Operations Research, **28**, 411-427.

Cheong Lee Fong M. (2005). New Models in Logistics Network Design and Implications for 3PL Companies, PhD dissertation, Nanyang Technological University, Singapore-MIT Alliance.

Cordeau J-F., Pasin F. and Solomon M. M. (2006). An Integrated Model for Logistics Network Design. Annals of Operations Research, **144(1)**, 59-82.

Dias J., Captivo M.E. and Climaco J. (2006). Capacitated Dynamic Location Problems with Opening, Closure and Reopening of Facilities. IMA Journal of Management Mathematics: Models and Applications in Location Analysis, **17(4)**, 317-348.

French A.P. and Wilson J.M. (2007). An LP-based heuristic procedure for the generalized assignment problem with special ordered sets. Computers & Operations Research, **34(8)**, 2359-2369.

Filho V.J.M.F. and Galvão R.D. (1998). A tabu search heuristic for the concentrator location problem. Location Science, **6**, 189-209.

Gong D., Gen M., Yamazaki G. and Xu W. (1997). Hybrid evolutionary method for Capacitated Location-Allocation problem. Computers and Industrial Engineering, **33(3)**, 577-580.

Hardin J.R., Nemhauser G.L., Savelsbergh M.W.P. (2007). Analysis of bounds for a capacitated single-item lot-sizing problem. Computers & Operations Research, **34(6)**, 1721-1743.

Hamer-Lavoie G. and Cordeau J.F. (2006). Un modèle pour la conception d'un réseau de distribution avec localisation, affectation et stocks. INFOR, **44**, 1-18.

Hinojosa Y., Puerto J. and Fernández F.R. (2000). A multiperiod two-echelon multicommodity capacitated plant location problem. European Journal of Operational Research, **123**, 271-291.

Jaramillo J.H., Bhadury J., Batta R. (2002). On the use of genetic algorithms to solve location problems. Computers & Operations Research, **29(6)**, 761-779.

Levi R., Shmoys D.B. and Swamy C. (2004). LP-based Approximation Algorithms for Capacitated Facility Location. Proceedings of the 10th MPS Conference on Integer Programming and Combinatorial Optimization, 206-218.

Martel A. (2005). The design of production-distribution networks: A mathematical programming approach. Supply Chain Optimization, Eds: J. Geunes and P.M. Pardalos, Springer, 265-306.

Melo M.T., Nickel S. and Saldanha da Gama F. (2006). Dynamic multi-commodity capacitated facility location: a mathematical modelling framework for strategic supply chain planning. Computers & Operations Research, **33**, 181-208.

Pirkul H. and Jayaraman V. (1998). A multi-commodity, multi-plant, capacitated facility location problem: formulation and efficient heuristic solution. Computers & Operations Research, **25(10)**, 869-878.

Robert A., Dejax P. and Péton O. (2006). A dynamic model for the strategic design of complex supply chains. Proceedings of the ILS06 conference, Lyon, France.

Saldanha da Gama F. and Captivo M.E. (1998). A heuristic approach for the discrete dynamic location problem. Location Science, **6(1)**, 211-223.

Syam, S.S. (2002). A model and methodologies for the location problem with logistical components. Computers & Operations Research, **29 (9)**, 1173-1193.

Thanh P.N., Bostel N. and Péton O. (2006). A dynamic model for the facility location in supply chain design. Rapport de Recherche 06/3/AUTO, École des Mines de Nantes. To appear in International Journal of Production Economics.

Trick M.A. (1992). A Linear Relaxation Heuristic for the Generalized Assignment Problem. Naval Research Logistics, **39**, 137-151.

Vazirani V.V. (2001). Approximation algorithms. Berlin-Heidelberg, Springer-Verlag.

Velásquez R., Melo T. and Nickel S. (2005). An LP-based Heuristic Approach for Strategic Supply Chain Design. In Operations Research Proceedings, Volume 2005, Berlin-Heidelberg, Springer-Verlag, 167-172.

Vila D., Martel A. and Beauregard R. (2006). Designing logistics networks in divergent process industries: A methodology and its application to the lumber industry. International Journal of Production Economics, **102(2)**, 358-378.