Hybrid Heuristic for the Inventory Location-Routing Problem

W.J. Guerrero[12], C. Prodhon[1]
[1] ICD-LOSI, UMR CNRS STMR, Troyes Université de Technologie de Troyes
12 Rue Marie Curie, BP 2060, 10000 Troyes, France
e-mail {guerrerw, caroline.prodhon}@utt.fr

N. Velasco[2], C.A. Amaya[2]
[2] PyLo, School of Engineering, Universidad de los Andes, Colombia
Cra 1 Este No 19A - 40 Bogotá (Colombia)
e-mail {nvelasco, ca.amaya}@uniandes.edu.co

RESUMEN

Resolver el problema de localización y ruteo de inventarios optimiza simultáneamente el diseño de la cadena de suministro y sus costos operacionales. Los supuestos del modelo incluyen el hecho que los vehículos pueden visitar más de un cliente por ruta y se incluyen las decisiones de inventario para un sistema con múltiples depósitos, con múltiples puntos de venta para un horizonte de planeación discreto. El problema es determinar el conjunto de depósitosporabrir, las cantidades por enviar desde los proveedores a los depósitos y de los depósitos a los puntos de venta, y la secuencia en que los puntos de venta serán visitados por una flota de vehículos homogénea. Un modelo de programación entera-mixta se propone para describir el problema. Un método hibrido que involucra un enfoque exacto dentro de un esquema heurístico es presentado. Su desempeño se prueba con instancias del problema de localizacion y ruteo, y de ruteo de inventarios.

PALABRAS CLAVE. Localización y Ruteo, Ruteo de Inventarios, Heurística Hibrida, OptimizaciónCombinatoria.

Área principal (Vehicle Routing, Supply chain Logistics, Hybrid Methods)

ABSTRACT

Solving the Inventory Location-Routing Problem can been seen as an approach to optimize both a supply chain design and its operations costs. Assumptions consider that vehicles might visit more than one retailer per route and that inventory management decisions are included for a multi-depot, multi-retailer system with storage capacity over a discrete time planning horizon. The problem is to determine the set of candidate depots to open, the quantities to ship from suppliers to depots and from depots to retailers per period, and the sequence in which retailers are replenished by an homogeneous capacitated fleet of vehicles. A mixed-integer linear programming model is proposed to describe the problem. Since the model is not able to solve exactly the targeted instances within a reasonable computation time, a hybrid method, embedding an exact approach within a heuristic scheme, is presented. Its performance is tested over instances for the inventory location routing, location-routing and inventory-routing problems.

KEY WORDS. Location-Routing Problem, Inventory-Routing Problem, Hybrid Heuristic, Combinatorial Optimization.

Main area (Vehicle Routing, Supply chain Logistics, Hybrid Methods)

1. Introduction

The design of a supply chain is considered as being a strategic level decision. It consists in identifying the optimal number of plants and their locations such that logistics costs are minimal. Instead, the management of a supply chain is usually considered to tackle tactical and/or operational decisions [16]. It concerns the cooperation between facilities in order to obtain, store and distribute materials, which also induces logistical costs. Balancing strategic with operational objectives over a planning horizon is then the challenge.

Most of the models dealing with this problem consider distribution to be performed by dedicated routes, e.g. one vehicle visits one client at the most. However, in the case where orders are smaller than vehicle capacity, this assumption is no longer true. The effects of ignoring routing decisions when locating depots is studied by [20]. When vehicles are not performing single-visit tours, locating depots, such that the sum of the distances between depots and retailers is minimized, is not optimal. Hence, Location-Routing problems propose to simultaneously optimize location and routing decisions for a single period. Examples are [5, 17, 18].

In addition, inventory and routing decisions are strongly interdependent for two reasons: First, the set of minimal cost routes to visit retailers is built as a function of the quantities to deliver which are determined by the inventory policies; and second, the ordering costs required to design such inventory policies include the transportation costs resulting from the choice of the routes. The optimal trade-off between inventory holding and distribution costs is studied by the Inventory-Routing Problem (IRP) as in [2, 3, 6]. Meanwhile, the trade-off between production set-up costs, inventory holding costs at the depot and distribution costs is optimized by the Integrated Production-Distribution Problem (IPDP) as in [4, 7].

Few journal papers refer to the issue of combining the three problems: Depot location-allocation, vehicle routing and optimizing inventory control policies. Considering deterministic demand, [1] propose a linear model. For the stochastic demand setting, [15] propose an iterative sequential optimization approach where the problem is never tackled in a global perspective.

Besides, two different characterizations of this problem exist. First, some research papers tackle a location-routing problem with the objective of minimizing the expected inventory management cost at retailer resulting in a non-linear objective function (e.g. [13]). Differing, the second approach treats the problem of location-routing problem managing stock at depots only (e.g. [12]).

Most of the research addressing the issue of locating depots simultaneously with inventory and routing decisions considers an EOQ-like formula supposing a regular demand (Wilson model). The first papers decomposed the problem into two independent sub problems; the solution for the inventory management problem fixes the quantities to deliver for the LRP. Later papers seek to solve a LRP with the expected cost of ordering plus holding stock on the non-linear objective function.

Most methods present single-period routing decisions as an approximation to the operational distribution costs. In reality, frequencies of delivery for each retailer might be different in order to optimize both routing and inventory management costs. Therefore, single-period routing decisions are not an appropriate approach when integrating inventory and distribution decisions. The presented approach explores distribution activities over a multi-period planning horizon to have a better insight of the tactical and operational costs when locating depots. In addition, we extend the approach to allow non-constant inventory holding costs and time dependent demand. This paper presents an extension of a previous work from the same authors [10] by including inventory decisions at both echelons and improves the work presented by [9]. In the remainder, the problem is detailed in section 2. A Hybrid Heuristic is described in section 3 and a computational study is presented in section 4. Conclusions are given in section 5.

2. Problem Definition

This paper tackles the design of a two-echelon supply chain comprising the location of depots serving the deterministic demand of retailers, and the assignment of the latter to a depot. The costs include the opening costs, the delivery costs (dedicated routes to depots, non-dedicated to retailers) and the inventory costs, including an obsolescence penalty costs. Interest arises mainly from two contexts: i) When temporary location is required (humanitarian missions); ii) When long-term objectives require a robust supply chain design such as in the large retail sector.

Formally, let J be a set of retailers facing a deterministic non-constant demand d_{jt}, $\forall j \in J$, $\forall t \in H$, with t a period and $H=\{1,...,p\}$ a discretized planning horizon. Also, a set of candidate depots I is available to replenish retailers and to stock up product. The ILRP is defined on a weighted and directed graph $G=\{V,A,C\}$.$V=\{J\cup I\}$ is the set of nodes in the graph. C is the cost matrix c_{ij} associated to the traveling cost from node i to node j in the set of arcs A. A set K of r identical vehicles is available. Each node $i \in I$ is associated to a storage capacity W_i. Each depot $j \in I$ is associated to an opening cost O_j and ordering cost s_j (dedicated route from the factory). Every vehicle has a capacity of Q units of product and the cost of using a vehicle at least once in the planning horizon is F. Let B_j be the initial inventory at facility $j\in V$. H_o = $\{0\}\cup H$ and $H'=H\cup\{p+1\}$ are included to model initial and final conditions. The holding plus obsolescence penalty cost for one unit of product kept at node $j \in V$ from period $t \in H_o$ until period $l \in H'$ is q_{jtl} (backlogging not allowed).

Let be the decision variables y_i=1 iff depot i is opened. f_{ij}=1 iff retailer $j \in J$ is assigned to depot $i \in I$, x_{ijkt}=1 iff arc (i,j) is crossed from i to j by vehicle k on period $t \in H$, T_i be the maximum number of vehicles used from depot i over H. Inventory decisions at echelon e are denoted by the variable w^e . The quantity replenished from depot i to retailer j on period t to satisfy the demand on period l using the vehicle k is denoted by w^2_{ijtlk} . The quantity of product used from initial stock at retailer j to satisfy demand in period t is denoted by w^2_{j0t}. At the first echelon, z_{ti}=1 iff depot i is replenished in period t, 0 otherwise. The quantity to replenish in depot i that is delivered in period t to satisfy the demand in period l is w^1_{itl}.

Then, the ILRP can be stated as follows:

$$\min \sum_{i\in I}\left(O_i y_i + FT_i + \sum_{l\in H} s_i z_{li}\right) + \sum_{i\in I}\sum_{t\in H_0}\sum_{l=t|l>0}^{p+1} q_{itl} w^1_{itl} + \sum_{t\in H'} q_{j0t} w^2_{j0t} + \sum_{i\in I}\sum_{j\in J}\sum_{t\in H}\sum_{l=t}^{p+1}\sum_{k\in K} q_{jtl} w^2_{ijtlk} + \sum_{i\in V}\sum_{j\in V}\sum_{k\in K}\sum_{t\in H} c_{ij} x_{ijkt} \tag{2.1}$$

Subject to:

$$\sum_{i\in I}\sum_{k\in K}\sum_{l=1}^{t} w^2_{ijltk} + w^2_{j0t} = d_{jt} \quad \forall j\in J, \forall t \in H \tag{2.2}$$

$$\sum_{l=0}^{t} w^1_{ilt} = \sum_{k\in K}\sum_{l=t}^{P+1}\sum_{j\in J} w^2_{ijtlk} \quad \forall i \in I, \forall t\in H \tag{2.3}$$

$$\sum_{t\in H'} w^2_{j0t} = B_j, \quad \forall j \in J \tag{2.4}$$

$$\sum_{t\in H'} w^1_{i0t} \models B_i \cdot y_i, \ \ \forall i \in I \tag{2.5}$$

$$\sum_{r=0}^{t}\sum_{l=t}^{P+1} w^1_{irl} \leq W_i \cdot y_i \ \ \forall i \in I, \ \ \forall t \in H \tag{2.6}$$

$$\sum_{l=t}^{P+1}\left(w^2_{j0l} + \sum_{r=1}^{t}\sum_{k\in K} w^2_{ijrlk}\right) \leq W_j \ \ ,\forall i \in I, \ \ \forall j \in J, \ \ \forall t \in H \tag{2.7}$$

$$\sum_{l=t}^{P+1} w^1_{itl} \leq W_i \cdot z_{ti} \ \ \forall i \in I, \ \ \forall t \in H \tag{2.8}$$

$$\sum_{i\in I} f_{ij} = 1, \ \ \forall j \in J \tag{2.9}$$

$$f_{ij} \leq y_i, \ \ \forall j \in J, \ \ \forall i \in I \tag{2.10}$$

$$\min(Q, W_j) \cdot \sum_{u\in J} x_{iukt} \geq \sum_{l=t}^{p+1} w^2_{ijtlk} \ \ \forall i \in I, \ \forall j \in J, \forall t \in H, \forall k \in K \tag{2.11}$$

$$\min(Q, W_j) \cdot \sum_{u\in J\cup\{i\}} x_{ujkt} \geq \sum_{l=t}^{p+1} w^2_{ijtlk} \ \ \forall i \in I, \ \forall j \in J, \forall t \in H, \forall k \in K \tag{2.12}$$

$$\sum_{j\in V} x_{ijkt} - \sum_{j\in V} x_{jikt} = 0, \ \ \forall t \in H, \ \ \forall i \in V, \forall k \in K \tag{2.13}$$

$$\sum_{i\in V}\sum_{k\in K} x_{ijkt} \leq 1 \ \ \forall \ t \in H, \ \ \forall j \in V \tag{2.14}$$

$$\sum_{i\in V}\sum_{k\in K} x_{jikt} \leq 1 \ \ \forall \ t \in H, \ \ \forall j \in V \tag{2.15}$$

$$\sum_{i\in I}\sum_{j\in J} x_{ijkt} \leq 1 \ \ \forall \ t \in H, \ \ \forall k \in K \tag{2.16}$$

$$\sum_{k\in K}\sum_{j\in J} x_{ijkt} \leq T_i \ \ \forall t \in H, \ \ \forall i \in I \tag{2.17}$$

$$\sum_{i\in I}\sum_{l=t}^{p+1}\sum_{j\in J} w^2_{ijtlk} \leq Q \ \ \forall k \in K, \ \ \forall t \in H \tag{2.18}$$

$$\sum_{u\in J} x_{iukt} + \sum_{u\in V\setminus\{j\}} x_{ujkt} \leq 1 + f_{ij} \ \ \forall i \in I, \ \ \forall j \in J, \ \ \forall t \in H, \ \ \forall k \in K \tag{2.19}$$

$$\sum_{i\in S}\sum_{j\in S} x_{ijkt} \leq |S| - 1 \ \ \forall t \in H, \ \ \forall k \in K, \ \ \forall S \subseteq J \tag{2.20}$$

$$x_{ijkt} \in \{0, 1\} \ \ \forall i \in I, \ \ \forall j \in J, \ \ \forall t \in H, \ \ \forall k \in K \tag{2.21}$$

$$y_i \in \{0, 1\} \ \ \forall i \in I \tag{2.22}$$

$$z_{it} \in \{0,1\} \ \ \forall i \in I, \ \ \forall t \in H \quad (2.23)$$

$$f_{ij} \in \{0,1\} \ \ \forall i \in I, \ \ \forall j \in J \quad (2.24)$$

$$T_i \in \mathbb{N} \ \forall i \in I \quad (2.25)$$

$$w^2_{ijtlk} \in \mathbb{R}^+ \ \forall i \in I, \ \forall j \in J, \ \forall t \in H, \ \forall l \in H'|l \geq t, \ \forall k \in K \quad (2.26)$$

$$w^2_{j0t} \in \mathbb{R}^+ \ \ \forall j \in J, \ \forall t \in H' \quad (2.27)$$

$$w^1_{itl} \in \mathbb{R}^+ \ \ \forall i \in I, \ \forall t \ inH_0, \ \forall l \in H|l \geq t \quad (2.28)$$

The objective function (2.1) sums the opening and ordering costs at depots, the costs of using a vehicle at least once, holding costs at depots and retailers with the distribution costs. Constraints (2.2) force to satisfy the demand. Inventory flow conservation is forced by constraints (2.3). The sum over the horizon of the quantity kept on stock from period zero up to period p+1 is equal to the initial stock as stated by constraints (2.4)-(2.5). Capacity for depots and retailers is guaranteed by (2.6) and (2.7). Ordering decisions at depots are forced by constraint set (2.8). Each retailer must be allocated to a single opened depot as stated by equations (2.9)-(2.10). Constraints (2.11)-(2.12) guarantee that if a retailer is replenished on period l with route k, it must be visited accordingly. Traditional vehicle flow conservation constraints are equations (2.13)-(2.16). Equations (2.17) link the cost of using vehicles with the routing decisions. Vehicles have limited capacity to deliver (2.18). Retailers can be assigned to a depot only if a route linking them is performed as stated by equations (2.19). Equations (2.20) are standard subtour elimination constraints. Constraints (2.21)-(2.28) state the nature of the decision variables.

3. Hybrid Heuristic

Exact procedures can only solve the model for very small instances within a reasonable time (see section 4). Heuristic methods seem to be able to find high quality solutions. The proposed heuristic framework tries to solve subproblems while exchanging information when moving between solution spaces.

Also, the embedded supply chain design problem (SCDP), neglecting the routing construction, might be solved to optimality using commercial solvers in reasonable computation time. Then, the problem is decomposed into decisions that are computed by exact methods and those obtained heuristically. The suggested pattern makes cooperate exact and heuristic procedures leading to a hybrid heuristic defined as a matheuristic [19]. Thus, the ILRP resolution induces: First, a supply chain design $S$is fully described by three elements: i) the set of the depots to open, ii) for each retailer, its assigned depot, iii) for each facility, the inventory policies. Second, a routing evaluation through $R_{(S)}$ the set of routes indicating the sequence in which retailers will be replenished at each period for a given S.

In the following subsections, the components of the approach will be described in order to assemble the proposed hybrid method in subsection 3.6 at algorithm 3.

3.1 Supply chain design

Assume the $m \times n \times p$ matrix C^* to be known in which each element c^*_{ijt} is the cost of delivering product from depot i to retailer j in period t. The MIP presented in section (2) is reduced to obtain a SCDP.

Decision variables x_{ijkt} are replaced by $\hat{x}_{ijt}, \forall\ i\ \in I, \forall j\ \in J, \forall\ t \in H,$ representing a binary variable equal to 1 iff depot i replenishes the retailer j in period t. w^2_{ijtlk} is replaced by $\hat{w}^2_{ijtl}, \forall\ i\ \in I, \forall j\ \in J, \forall\ t \in H, \forall\ l\ \in H'$ representing the quantity replenished by depot $i$in period t to stock until period l at retailer j. Accordingly, the objective function (2.1) is replaced by:

$$(3.1) \qquad \min \sum_{i\in I}\left(O_i y_i + FT_i + \sum_{l\in H} s_i z_{li}\right) + \sum_{i\in I}\sum_{l\in Ho}\sum_{t\in H'} q_{jtl} w^1_{itl} + \sum_{t\in H'} q_{j0t} w^2_{jot} + \sum_{i\in I}\sum_{j\in J}\sum_{t\in H}\sum_{l\in H'} q_{jtl}\hat{w}^2_{ijtl} + \sum_{i\in I}\sum_{j\in J}\sum_{t\in H} c^*_{ijt}\hat{x}_{ijt}$$

The index k is easily removed from constraints (2.2),(2.3), (2.7), (2.18), and (2.26). Equations (2.4)-(2.6),(2.8)-(2.10), (2.21)-(2.25), (2.27), and (2.28) remain unchanged. The following constraints are added to complete the formulation.

$$(3.2) \qquad \min(Q, W_j)\cdot \hat{x}_{ijt} \geq \sum_{l=t}^{p+1} \hat{w}^2_{ijtl} \ \forall i\in I,\ \forall j\in J, \forall t\in H$$

$$(3.3) \qquad f_{ij} \geq \hat{x}_{ijt} \ \forall\ i \in I,\ t\in H,\ \forall j\in J$$

$$(3.4) \qquad \sum_{j\in J}\sum_{l=t}^{p+1} \hat{w}^2_{ijtl} \leq T_i\cdot Q \ \forall t\in H,\ \forall i\in I$$

Equations (3.2)-(3.3) relate distribution activities to inventory flow and allocation constraints. Constraint (3.4) estimates that the minimum number of vehicles to use times the vehicle capacity Q is larger or equal than the total quantity to replenish.

Since the distribution cost depends on the embedded routing problem which can only be solved once the quantities to deliver per period are known, the initial matrix C^* is estimated to be a random fraction of the direct delivery cost. Each element of the matrix$c^*_{ijt} = \xi_1 \cdot c_{ij}\ \forall\ i \in I, \forall j\ \in J, \forall\ t \in H$, where $\xi_1$is an uniform random variable $\sim$*Unif*[α,2α] . The parameter α is fixed a priori.

Routing procedures cooperate with subsequent supply chain designs. C^* is updated every time feasible routing costs are computed. This update is performed with the information of a feasible solution by the cost of detour of the route.

3.2 Randomized routing heuristic

Even when the supply chain is designed, the remaining routing decisions are difficult to solve since it reduces to the well-known VRP for every period and depot [21]. We consider the presolved assignment as a mean to provide a subset of promising depots to open and to tackle an LRP per period. A simple heuristic procedure that provides good solutions for the LRP is the RECWA, randomized extended Clarke and Wright algorithm, implemented as in [18]. It is an extension for the multi-depot case of the Clarke and Wright saving's algorithm [8]. A randomization on the selected merge allows some diversification. However, retailers' allocation to depots must be the same along the horizon, which is not warranted by solving an LRP per period and requires a repairing operator. It repairs the solution with a randomized Clarke and Wright algorithm for each period while fixing the allocation decisions to one of the assigned depots.

3.3 Local search

The order of the neighborhoods in our local search is:

- MOVE: The visit of a retailer is shifted from its current position to a different position within the same route or to other routes from the same depot and period.
- SWAP: The positions of two different retailers are exchanged. Both must share depot and period but they might be in the same route or not.

- 2-OPT: Two arcs are removed from the solution and new arcs are included to assure feasibility of the solution. The removed arcs might belong or not to the same route but they must share the same depot and period.
- SHIFT DELIVERY DATE: The first delivery date for a retailer is shifted to the earliest date. It will be replenished with the maximum capacity to minimize any chance of stockout. Shipping in latter periods is revisited.
- 2-SHIFTED DELIVERY DATE: Same as SHIFT DELIVERY DATE but considering a couple of customers sharing the same depot. This movement aims to synchronize the deliveries of retailers over the time dimension.
- DEPOT REALLOCATION: The allocation the set of scheduled visits of a retailer is shifted to another depot.
- DEPOT ALLOCATION SWAP: Two retailers allocated to different depots are exchanged in their depot allocation. The scheduled visits remain unchanged.

A variable neighborhood descent (VND) is chosen [11]. It evaluates each neighborhood and if an improving movement is found, the search starts again from the first neighborhood. If no improvement movement is found, the search continues with the next neighborhood. LS applies a first improvement movement strategy except for 2-SHIFTED DELIVERY DATE that uses best-improving movement.

3.4 Intensification

Algorithm 1 presents the procedure to re-evaluate inventory-routing decisions with a dedicated procedure. For *n* iterations, a dynamic lot-sizing problem (DLSP) is solved with a MIP solver and the *C** matrix is updated. This problem consists on determining the optimal quantities to keep on stock at the facilities if location-allocation decisions are fixed. The MIP presented in section (3.1) is reduced by fixing *y* and*f* variables with the values in S.

Once the inventory policies are revisited in step 3, routing decisions are recomputed in step 4. A randomized version of the Clarke and Wright algorithm is proposed to build $R_{(S)}$ fixing *y* and *f*values. The solution is improved by LS in step 5. To avoid local optima, a perturbation procedure is applied in step 14. For subsequent calls of the DLSP procedure, some random elements of the *C** matrix are perturbed to zero. Additionally, some cuts are added to the MIP by forcing the solution to visit some retailers at random periods.

Algorithm 1. $Intensification(C^*, S, R_{(S)})$

1: $n_p = 0$
2: **for** $k_3 = 1$ *to* n **do**
3: $\quad S \leftarrow DLSP(C^*, S)$
4: $\quad R_{(S)} \leftarrow RCWA(S)$
5: $\quad LocalSearch(S, R_{(S)})$
6: $\quad Update(C^*, R_{(S)})$
7: $\quad$ **if** $C(S, R_{(S)}) < C_{best}$ **then**
8: $\quad\quad C_{best} = C(S, R_{(S)})$
9: $\quad\quad S_{best} \leftarrow S$
10: $\quad\quad R_{best} \leftarrow R_{(S)}$
11: $\quad$ **else**
12: $\quad\quad n_p = n_p + 1$
13: $\quad\quad$ **if** $n_p = n_0$ **then**
14: $\quad\quad\quad C^* \leftarrow perturbation(C^*)$
15: $\quad\quad\quad n_p = 0$
16: $\quad\quad$ **end if**
17: $\quad$ **end if**
18: **end for**

3.5 Post-optimization

Given the best solution (S, $R_{(S)}$) found, a post-optimization procedure on allocation-routing decisions is proposed in the form of an iterated local search (ILS) [14]. First, S and $R_{(S)}$ are mutated by reallocating a subset of retailers to a depot with available capacity or a new depot is opened. If LS does not improve the best solution, it is discarded and mutation is repeated on the best solution. The procedure is repeated for up to N_1 improving iterations or N_2 iterations without improvement. The procedure is detailed in algorithm 2.

Algorithm 2. *Procedure: $ILS(S, R_{(S)})$*

$i = 1$ *and* $j = 1$
while $i \leq N_1$ *and* $j \leq N_2$ **do**
    $mutate(S, R_{(S)})$
    $LocalSearch(S, R_{(S)})$
    if $C(S, R_{(S)}) < C_{best}$ **then**
        $C_{best} = C(S, R_{(S)})$
        $S_{best} \leftarrow S$
        $R_{best} \leftarrow R_{(S)}$
        $i = i + 1$
    else
        $S \leftarrow S_{best}$
        $R_{(S)} \leftarrow R_{best}$
        $j = j + 1$
    end if
end while

3.6 Algorithm overview

The components described are integrated in a multi-start hybrid heuristic. Algorithm 3 details the procedure. At step 7, a supply chain design S is computed by solving the SCDP using a commercial solver. The randomized routing heuristic detailed in section 3.2 is performed to optimize $R_{(S)}$in step 8. Subsequently, the C* matrix is updated. The intensification procedure is called in step 11 (see 3.4). Steps 13-17 updateS_{best} and R_{best} . Steps 6 to 18 are repeated until no improvement is perceived or MAX_{iter} iterations are performed. In step 19, a perturbation procedure is called to induce diversification. It creates a tabu list with a single depot that was opened in the last call of SCDP, forcing its closure in the next call of the procedure. Step 24 calls the post-optimization procedure.

Algorithm 3. *Main Algorithm (Overview)*

1: $S, R(S) \leftarrow 0$
2: $S_{best}, R_{best} \leftarrow 0$
3: $C_{best} = \infty$
4: **for** $k_1 = 1$ *to* m **do**
5:     $C^* \leftarrow random \cdot C$
6:     **while** *(Improvement or less than* MAX_{iter} *iterations)* **do**
7:         $S \leftarrow SCDP(C^*)$
8:         $R_{(S)} \leftarrow RECWA(S)$
9:         $LocalSearch(S, R_{(S)})$
10:         $Update(C^*, R_{(S)})$
11:         $ILS(C^*, S, R_{(S)})$
12:         $Update(C^*, R_{(S)})$
13:         **if** $C_{best} > C(S, R_{(S)})$ **then**
14:             $C_{best} = C(S, R_{(S)})$
15:             $S_{best} \leftarrow S$

```
            R_best ← R_(S)
        end if
    end while
    RandomDepotClosure(S)
end for
if m > 1 then
    S ← S_best
    R_(S) ← R_best
    ELS(S, R_(S))
end if
```

4. Computational Study

The algorithm was coded in Mosel and solved with Xpress-IVE 7.0. Tests correspond to an Intel Xeon with 2.80 GHz processor and 12 GB of RAM. 15 ILRP instances were generated with the following size: m = 5 depots, $n \in \{5, 7\}$ retailers, $p \in \{5, 7\}$ periods. They are labeled as *m-n-p-x* and *x* is used to itemize.

Demand at retailer j for period t is $d_{jt} \sim N(\mu_j, \sigma_j)$, were $\mu_j \in [\,5, 15]$ and $\sigma_j \in [\,0, 5]$. The opening costs O_i is generated with a normal distribution with parameters (μ_j, σ_j) chosen from the set of pairs {(1000,20), (5000,100), (8000,300)}. s_i is chosen from the set {100, 500}. The coordinates (X_i, Y_i) for facility i are randomly generated in a square of size 100×100. Transportation cost c_{ij} equals the closest integer of a hundred times the euclidean distance from i to j. Q is a random integer in the interval [15, 75]. The cost F is selected from the set {350, 1000, 5000}. Depot capacity W_i is randomly generated in the interval [D/3, D], were $= \sum_{i \in J} \sum_{t \in H} d_{jt}$. Retailer's capacity W_j are randomly generated in the interval $[g_j, 3g_j]$ were $g_j = max_t \, d_{jt}$. Initial inventories B_j were chosen from the set $\{0, d_{1j}\}$ for retailers and B_i from the set $\{0, 10D/n\}$for depots. Inventory holding costs for a single period $t \in H_o$ at retailers and depots $j \in V$, $q_{j,t,t+1}$ is generated in the interval [0.03,0.50]. The inventory holding costs for k periods as $q_{j,t,t+k} = \sum_{l=t}^{t+k-1} q_{j,l,l+1} + k\xi_2$, were $\xi_2 \sim Unif[0.01, 0.02]$ represents the obsolescence penalty cost per period.

18 classical LRP instances with capacitated vehicles, 5 capacitated depots and up to 100 retailers available at *http://prodhonc.free.fr* were used to test our approach. Also, 40 "order-up to level" benchmark instances for the IRP with up to 40 customers, 3 periods and holding cost between [0.01, 0.05] are considered. They are available at http://www-c.eco.unibs.it/~bertazzi/.abls.zip. We added the constraint forcing to replenish a retailer such that the stock rises up to W_j if visited for the "order-up to level" policy.

Preliminary tests using the non-parametric test of Friedman lead us to conclude that the most stable results are for α=0.4, $MAX_{iter} = 4$, and $N_1 = N_2 = 15$.

Table 1 presents, for ILRP instances, the comparison between the commercial solver UB with a time limit of 2.5 hours, the results of our hybrid heuristic and a sequential heuristic (H1) that emulates the traditional sequential approach. H1 is equivalent to build a supply chain design using a commercial solver and to make inventory-routing decisions through the procedure described in section 3.4. Columns two and three present the cost of the best feasible solution found by the solver within 20 and 60 seconds respectively. Column four presents the gap between the solution found in 60 seconds and the best found solution within the time limit that is presented in column five (UB). Column six (CPU UB) presents the time when UB was found in seconds. Columns seven, eight and nine present the average cost, gap to UB and computation time of our heuristic for three runs. Columns ten and eleven present the gap to UB and CPU for H1.

Our hybrid heuristic outperforms the solution found by the commercial solver by 0.58%. 7 out of 15 new best solutions are found, and other 4 solutions have a gap to UB inferior to 0.4%. The only interest of H1 is its speed. The traditional approach provides solutions that are about 2.41% more expensive than UB found by the solver. The solver was not able to find a feasible solution within the time limit for instance 5-7-5-c.

	SOLVER					HYBRID HEURISTIC			H1	
Instance	SOL (20 s)	SOL (60 s)	GAP (60 s)	UB	CPU UB	COST	GAP (%)	CPU (s)	GAP (%)	CPU (s)
5-5-5-a	97223.6	97221.9	3.7	93718.5	173	93625.3	-0.10	25.2	-0.10	4.7
5-5-5-b	89066.3	69035.8	10.5	62494.5	2176	62206.9	-0.46	22.4	0.65	4.0
5-5-5-c	86432.0	86432.0	23.9	69760.3	1356	70881.0	1.61	48.0	1.73	7.9
5-5-5-d	97424.4	97424.4	3.9	93801.2	592	93451.2	-0.37	28.6	-0.37	4.7
5-5-5-e	89066.3	69035.8	10.4	62494.5	2178	62454.2	-0.06	20.1	1.25	3.9
5-5-7-a	20150.5	84911.3	9.7	77404.1	1093	70966.5	-8.32	320.7	-1.85	89.3
5-5-7-b	203182.7	167289.2	50.8	110940.0	4243	107478.5	-3.12	394.9	1.23	46.7
5-5-7-c	171860.4	121341.7	28.9	94150.2	8163	94152.9	0.00	328.3	6.66	212.0
5-5-7-d	99841.3	99841.3	13.8	87744.2	3951	87744.2	0.00	62.1	4.57	8.8
5-5-7-e	-	-	-	69025.9	1819	67275.4	-2.54	176.4	2.89	44.8
5-7-5-a	150801.3	143879.9	110.1	68485.2	873	69739.3	1.83	55.0	4.44	7.9
5-7-5-b	-	-	-	76339.1	1317	78662.5	3.04	114.8	3.65	51.0
5-7-5-c	-	-	-	-	-	138998.0	-	4964.9	-	1409
5-7-5-d	-	-	-	99988.9	7432	100001.0	0.01	222.3	6.60	26.2
5-7-5-e	-	92184.8	48.7	62010.1	1404	62234.0	0.36	81.2	2.33	11.5
Average	110504.9	102599.8	28.6	86969.9	4063	83991.39	-0.58	457.66	2.41	128.8

TABLE 1. Performance on ILRP instances.

			Hybrid Heuristic					Prins et al. (2006)[16]		
Instances	♯	BKS	cost	gap	CPU	gap_{min}	gap_{max}	cost	gap	CPU
20R-5D-1P	4	45086.7	45153.8	0.13	0.97	0.10	0.19	45143.7	0.10	0.17
50R-5D-1P	8	74112.6	74782.4	0.68	4.69	0.26	1.23	74700.6	0.81	2.14
100R-5D-1P	6	199164.7	201957.4	1.37	28.7	1.04	1.74	202210.2	1.55	22.1
Average		109346.4	110589.9	0.79	11.9	0.48	1.17	110635.6	0.90	8.35

TABLE 2. Benchmark on Location-Routing problem

Table 2 sums up the results on LRP instances. The number of instances (#), Best known solution (BKS) from *http://prodhonc.free*, the average cost, gap to BKS and CPU of our hybrid heuristic for three runs and the minimum and maximum gap to BKS are shown in columns 2-8 respectively. Average results for instances with 5 depots, 20 retailers (20R-5D-1P), 50 retailers (50R-5D-1P) and 100 retailers (100R-5D-1P) are presented. They show an average gap of 0.79% computed in 11.9s. We compare our methodology with the method of [18] (coded in C++ and executed on a Dell OPTIPLEX GX260, 512MB of RAM, with a 2.4 GHz processor) detailed in columns 9-11. Our method is competitive with the algorithm of [18] even if it is not dedicated to the LRP.

			Hybrid Heuristic					Benchmark		
Instances	♯	z*	aver. cost	aver. gap	aver. CPU	gap_{min}	gap_{max}	Archetti (2011)[2]	Bertazzi (2002)[6]	aver. CPU
5R-1D-3P	5	1418.7	1418.7	0	3.0	0	0	0	2.88	3.0
10R-1D-3P	5	2228.7	2236.2	0.34	7.8	0	0.49	0	0.78	12.8
15R-1D-3P	5	2493.5	2520.4	1.08	28.9	0.03	2.44	0	2.56	41.4
20R-1D-3P	5	3053.0	3160.2	3.51	82.6	1.71	6.61	0.02	3.83	104.2
25R-1D-3P	5	3451.1	3532.9	2.37	191.1	0.38	4.71	0	2.99	258.8
30R-1D-3P	5	3643.2	3731.0	2.41	413.2	0.94	4.81	0.02	3.60	515.0
35R-1D-3P	5	3846.9	4016.9	4.42	812.0	2.29	7.16	0.04	4.46	808.8
40R-1D-3P	5	4125.7	4393.9	6.50	1256.4	5.4	7.7	0.06	6.46	1168.6
Average		3032.6	3126.29	2.58	349.36	1.35	4.24	0.02	3.45	364.08

TABLE 3. Benchmark on Inventory-Routing problem instances

Results are also competitive on IRP instances as shown in table (3). (#) is the number of instances. Results of our heuristic for average, minimum, and maximum gap, and CPU and CPU to best solution (CPU_{best}) for three runs are presented in columns 4-8. Results show an average gap to optimal solution (z*) of 2.58%. The % gap of [2] and [6] are presented in columns 9-11 using an Intel Dual Core 1.86 GHz and 3.2 GB RAM and coded in C++. CPU for [2] (not available for [6]) are given in column 11. The approach computes solutions of intermediate quality between [2] and [6] with similar CPU, while it is not a dedicated method.

5. Conclusions

The Inventory-Location-Routing Problem (ILRP) is presented as an approach to supply chain design considering inventory and routing cost to overcome the fact that traditional approach of decomposing strategic, tactical/operational decisions often provides sub-optimal solutions. Our hybrid heuristic approach solves the supply chain design problem using exact methods while the remaining routing decisions are computed by heuristic procedures. By alternating between decisions spaces and information sharing, the algorithm manages to optimize globally the components of the problem without oversimplifying it. Results for randomly generated instances show important cost savings over the traditional approach and efficient computation when compared to commercial solvers. The ILRP reduces to the LRP and the IRP under certain conditions. Our tests show a robust performance over larger benchmark instances for both the LRP and the IRP. Future research comprises the ILRP with two routing decision levels as in the 2E-LRP.

Acknowledgement: This research is partially supported by Champagne-Ardenne Regional Council (France) and Centro de Estudios Interdisciplinarios Básicos y Aplicados - CEIBA (Colombia).

References

[1] **Ambrosino,D., Scutellà, M.G.** (2005), Distribution network design: New problems and related models. *European Journal of Operational Research*, 165(3):610-624.

[2] **Archetti, C., Bertazzi, L., Hertz, A. and Speranza. M.G.** (2011) A hybrid heuristic for an inventory routing problem. INFORMS Journal on Computing, 1:1-16.

[3] **Archetti, C., Bertazzi, L., Laporte, G. and Speranza. M.G.** (2007) A branch-and-cut algorithm for a vendor-managed inventory-routing problem. *Transportation Science*, 41(3):382-391.

[4] **Armentano, V., Shiguemoto, A. and Lokketangen, A.** (2011) Tabu search with path relinking for an integrated production distribution problem. *Computers & Operations Research*, 38(8):1199-1209.

[5] **Belenguer, J.-M., Benavent, E., Prins, C., Prodhon, C. and Wolfler-Calvo , R.** (2011) A branch-and-cut method for the capacitated location-routing problem. *Computers & Operations Research*, 38(6):931- 941. [6]

[6] **Bertazzi, L., Paletta, G. and Speranza, M.G.** (2002) Deterministic order-up-to level policies in an inventory routing problem. *Transportation Science*, 36(1):119-132.

[7] **Boudia, M. and Prins, C.** (2009) A memetic algorithm with dynamic population management for an integrated production-distribution problem. *European Journal of Operational Research*, 195(3):703-715.

[8] **Clarke, G. and Wright, J.** (1964). Scheduling of vehicles from a central depot to a umber of delivery points. *Operations Research*, 12(4):568-581.

[9] **Guerrero, W., Prodhon, C., Velasco, N. and Amaya, C.A**. (2011) A matheuristic for the inventory location routing problem with deterministic demand. In MIC 2011 (Metaheuristics International Conference), Udine, Italie.

[10] **Guerrero, W., Prodhon, C., Velasco, N. and Amaya, C.A.** (2012) Hybrid heuristic for the inventory location-routing problem with deterministic demand. In 5th International Workshop on Freight Transportation and Logistics, Mykonos, Greece.

[11] **Hansen, P. and Mladenovic, N.** (2003) Variable neighborhood search. In F. Glover and G. Kochenberger, editors, *Handbook of Metaheuristics*, volume 57 of International Series in Operations Research & Management Science, pages 145-184. Springer New York.

[12] **Javid, A. and Azad, N.** (2010) Incorporating location, routing and inventory decisions in

supply chain network design. *Transportation Research Part E: Logistics and Transportation Review*, 46(5):582-597.

[13] Liu, S. and Lin, C. (2005) A heuristic method for the combined location routing and inventory problem. *The International Journal of Advanced Manufacturing Technology*, 26:372-381.

[14] Lourenço, H., Martin, O. and Stützle, T. (2003) Iterated local search. In F. Glover and G. Kochenberger, editors, *Handbook of Metaheuristics*, volume 57 of International Series in Operations Research & Management Science, pages 320-353. Springer New York.

[15] Ma, H. and Davidrajuh, R. (2005) An iterative approach for distribution chain design in agile virtual environment. Industrial Management & Data Systems, 105(6):815-834.

[16] Melo, M., Nickel, S. and da Gama, F.S. (2009) Facility location and supply chain management: A review. *European Journal of Operational Research*, 196(2):401-412009.

[17] Prins, C. Prodhon, C. Ruiz, A. Soriano, P. and Wolfler-Calvo, R. (2007) Solving the capacitated location-routing problem by a cooperative lagrangean relaxationgranular tabu search heuristic. *Transportation Science*, 41(4):470-483.

[18] Prins, C., Prodhon, C., and Wolfler-Calvo, R. (2006) Solving the capacitated location-routing problem by a GRASP complemented by a learning process and a path relinking. *4OR: A Quarterly Journal of Operations Research*, 4:221-238.

[19] Raidl, G. and Puchinger, J. (2008). Combining (integer) linear programming techniques and metaheuristics for combinatorial optimization. In C. Blum, M. Aguilera, A. Roli, and M. Sampels, editors, *Hybrid Metaheuristics*, volume 114 of Studies in Computational Intelligence, pages 31{62. Springer Berlin / Heidelberg.

[20] Salhi, S. and Rand, G. (1989) . The effect of ignoring routes when locating depots. *European Journal of Operational Research*, 39(2):150-156.

[21] Toth P. and Vigo D.(2001) editors. The vehicle routing problem. *Society for Industrial and Applied Mathematics*, Philadelphia, PA, USA.