Inventory Location Routing Problem: A Column Generation Approach

Mateo Gutiérrez Granada and Catalina Wills Silva
Department of Industrial Engineering
Universidad de los Andes
Colombia

Abstract

This paper presents a solution method based on column generation for the Inventory Location Routing Problem. It chooses simultaneously the location of a set of depots to decide upon the inventory management and the set of routes to satisfy customers' demands, in order to minimize the total system cost. The original problem is decomposed into a Master Problem (Location Problem) and an Auxiliary Problem (Inventory-Routing Problem). The Auxiliary Problem is solved by a backtracking, depth search algorithm, which was tested on random generated instances. The computational experiments indicate that the solution method is efficient for small size problem instances.

Keywords
Operations research, Location Routing Problem, Column Generation

1. Introduction
The main purpose of companies is to focus their efforts on maximizing profits. All the organizations have processes that lead to the success or failure of this goal. It is therefore important for companies to consider that operational processes within the supply chain should concentrate on maximizing performance and reducing costs [2]. In today´s competitive environments, companies must make decisions concerning operational processes to maintain efficiency and some of the most important are related to the *distribution network design*, which can be divided in three sub-problems: *depot location, vehicle routing and inventory management.*

Ballou [2] explains how *effective* logistics management should include three levels of decisions: (1) Strategic decisions, involving plans for a relatively long time period and important capital investment, (2) tactical decisions, involving plans for the short term (a year, a semester…) and moderate investments; and (3) operational decisions, involving day to day operations and low capital investments. Connecting these three levels of decisions to the distribution network, the depot location is related to long term decisions and can be classified as a strategic decision. On the other hand, inventory management includes decisions for multiple months and is thus related to tactical decisions. Finally, vehicle routing is a day to day operation and part of the operational decisions. In addition, in various settings, the decisions for finding a solution to each of these problems are interrelated [9]. As each decision will affect the others, it is beneficial to consider all simultaneously and not sequentially as the traditional approach suggests [19]. As a result, companies that succeed in coordinating these three sub-problems will remain competitive and will maintain effective logistics management in today´s environment.

Naturally, Operational Research is oriented in applications based on real world problems [9]. The coordination of these three problems in a supply chain represents an important basis for the design of different distribution networks. To mention some applications of ILRP we highlight some examples exposed by Nagy and Salhi [13] such as delivery to retail shops, food and drink industries, and the distribution of various consumer goods. Yu et al. [19] also mention different works where the LRP is applicable to a wide variety of fields such as the newspaper delivery, waste colletion and again, food and drink distribution.

Oftentimes, finding a solution which includes these 3 decisions has a large number of possible solutions, and therefore it also leads to a problem with an exponential number of variables. Consequently, the ILRP can be

classified as a NP-Hard problem with an exponential number of variables. As the Column Generation (CG) is used for solving linear problems with an exponential number of variables [5], it is important and interesting to present a present a solution method based on CG for the Inventory-Location-Routing Problem (ILRP). It was first implemented by Gilmore and Gomory in 1961 to solve the cutting stock problem.

The structure of this paper continues as follows. In the next section we present a concise literature review on vehicle routing problems, more specifically on the ILRP and we also present a brief review on the column generation approach. In Section 3 we present the ILRP definition and we present the assumptions for the simplification of our model. In Section 4 we expose the main idea of a column generation approach and we expose our formulation for the ILRP for this approach. In the next section we describe the algorithm used for the Auxiliary-Problem. In Section 6 we present the computational experiments and finally we present some conclusions given in Section 7.

2. Literature Review

The goal of the location routing and inventory problem, as mentioned before, is to choose depots from multiple possible locations, to schedule vehicles' routes to satisfy customer's demands and to determine the inventory policy based on the minimization of the total system cost [5]. In the literature, there are several papers which include the combination of two of these three problems and even though the consideration of these three decisions is an important issue in designing distribution systems, the idea of combining the three of them has not been deeply explored. There is a vast collection of research about a combination of the Vehicle Routing Problem (VRP) with the Location problem (Location Routing Problem, LRP) [13,19], and the combination of the VRP with inventory constraints, better known as Inventory Routing Problem (IRP) [15,18], but there is little literature concerning how to optimize these three problems in one model.

Many routing problems, including the LRP and the IRP, have been solved through a CG approach and the solutions to these methods have shown good results compared with other approaches. Besides, these methods have been very successful in the search for optimal solution (Oppen, Lokketangen, & Desrosiers, 2009). However, as the ILRP has not been extensively explored, there is not a CG approach for this problem and therefore it presents a rich research opportunity.

Oppen et al. [15] present a solution method based on CG for the livestock collection problem (LCP). This is a VRP extended with inventory constraints so it can be classified as an IRP. They argue that there are no exact methods that can handle this efficiently for small instances. The CG solution method proposed solves instances up to 25 orders and gets better results than what has done been before. In addition, it is based on a richer model closer to a real world problem. They conclude the paper by remarking that heuristic methods can often obtain much better solutions than manual planning for larger problems. Nevertheless, they claim that if a CG approach is to be implemented for solving real world instances, the problem has to let go the idea of optimality. However, you can claim that this approach, compared to previously reported results for the LCP is powerful because it helps to find a good solution closer to the real-world problem with large instances.

On the other hand Jin et al.[7] present a CG approach for the split delivery vehicle routing problem (SDVRP) with large demand. This approach, however, has a restriction of a fixed number of vehicles but without the requirement of a limited distance of each trip. Columns include route and delivery information with a pricing sub-problem that generates a route by minimizing the reduced costs. With the computational results, they conclude that the problem solved by a CG approach shows a better solution than what has been done in the literature. Nonetheless, they claim that more studies are necessary to improve the solution by modifying the algorithm for the sub-problem.

Another example where CG is a good solution for complex problems is demonstrated in [3]. They expose a vehicle routing problem with a heterogeneous fleet of vehicles (different capacities and variable costs). Through a CG approach they find a successful solution for the VRP with time windows. With the computational results they claim that the algorithm used is more powerful than the existing ones in terms of quality and computational time. They finish the paper by commending the potential of using a CG approach for the solution to difficult problems.

Sambola [17] presents a model based on this approach for the LRP as well. The author recommends the use of CG for this problem due to the high number of variables derived from the model. The variables are associated with feasible routes that originate at each of the potential location sites. The Master Problem in this case is obtained via

Lagrangian relaxation and the sub-problem consists of identifying new columns to include in the Master Problem. The purpose is to find routes with associated reduced cost, with the condition of not violating the capacity of the plant. To solve this sub-problem they have modeled it as an Elementary Shortest Path Problem with Resource Constraints (ESPPRC), which is considered to be *NP*-hard. In order to solve the sub-problem they present a simple heuristic that allows them to obtain efficient results in general.

A SA heuristic is proposed in [1] for the ILRP. These authors expose the idea of combining the three sub-problems to simultaneously optimize Location, Inventory and Routing decisions in a supply chain and present a heuristic method based on Tabu Search and SA. The main contribution of this paper is an effective and efficient result of a model that combines the three main processes of a supply chain distribution network. The model can be extended in several realistic directions.

In the literature there is also another approach for the multi-depot-location routing problem with inventory control decisions. Liu and Lee (2003) propose a mathematical model for this problem that consists of a two-phase heuristic. They use a route-first, location-allocation second approach to find a better solution in Phase 1. Phase 2 consists of improving the initial solution. With the obtained solution they show that the proposed heuristic is better in terms of system cost and CPU time than those not considering inventory decisions. They conclude that they have developed an effective and efficient heuristic method for the multi-depot location routing problem with inventory decisions, which is still an unachieved goal.

On the other hand, Lin and Liu [9] present a heuristic method for the combined location routing and inventory problem (CLRIP). They split it into two sub problems: a depot location problem and routing-inventory problem. In the same way as the model exposed in [1], they combine a Tabu search with Simulated Annealing to solve the problem and show that the proposed heuristic in terms of average system cost and computation time is better than existing methods.

In conclusion, it is evident that the CG is a good solution for problems involving an exponential number of variables, such as the ILRP. Additionally there are not many papers dealing with the ILRP and that it is a problem that has not been extensively explored. Even though the results of the ILRP have demonstrated efficient and effective results, it is a challenge for us to present a solution to the ILRP based on a CG Approach.

3. Inventory Location Routing Problem: Problem Definition

The ILRP can be described as follows. Given a set of customers, a set of possible depots and a set of periods, the goal of this problem is to choose simultaneously the location of the depots, to decide the inventory policies for customers (retailers) and to establish a set of routes that vehicles should use to meet customers´ demands, in order to minimize the total system cost, which includes the route costs, the depot location costs and the inventory management cost. The ILRP consists of three sub problems that are interrelated to one another: depot-location problem, inventory control problem and vehicle routing problem. The distribution network in this case consists of two echelons: depots (supply points) and customers (demand points). There is a direct delivery from depots to customers (we are not considering distribution centers, between both of them). Each customer faces a dynamic demand per period for one type of product (mono/product). The depots' mission is to send the product to the customers over a finite and discrete time horizon. The depots are responsible for customer inventory management and they must decide if a customer is satisfied before the corresponding period, exactly at the requiered period or they backlog the demand which may or may not be satisfied in the future [18]. The depots are also responsible for the management of their own inventory and for making necessary orders to satisfy customer's demands. In order to tackle the problem, our model considers the following assumptions:

- The location of the candidate depots is known.
- Each depot has an infinite capacity and infinite stock of product.
- Split deliveries are not allowed. Vehicles are allowed to visit each client only once per period.
- Each customer faces a different demand for a single item per time period.
- The demand of each customer is known, deterministic and dynamic.
- Backlogging is allowed, but it has an associated extra cost.
- The initial inventory at customers is zero.
- Vehicles` capacities are the same and the fleet type is homogeneous

4. Column Generation Based Formulation

4.1 Column Generation

As previously mentioned, CG approaches have been implemented in many routing problems with good results [3, 7, 15]. When the problem includes constraints which lead to difficulty in solving them, it is useful to create sub problems which can handle the problematic constraints in an efficient manner. Therefore, this paper proposes this approach for solving an ILRP.

For the CG Approach we start with the original ILRP problem, which is going to be split into two sub problems: we call the depot location problem, the 'Master Problem' and the inventory routing problem the 'Auxiliary Problem'. The principal idea of the CG is to find non-basic variables with a negative reduced cost to include in the Master Problem [5]. Probing the reduced cost of the variables relies on the sub-problem. In this case variables are associated with routes starting and ending at the same depot with delivery information. These routes will be included in the Master Problem which is going to be solved with these new routes. Then, the dual variables are introduced to the Auxiliary Problem to identify new variables with negative reduced cost. When there are no more variables with negative reduced cost, then the original problem is optimized. This dynamics is illustrated in Figure 1.

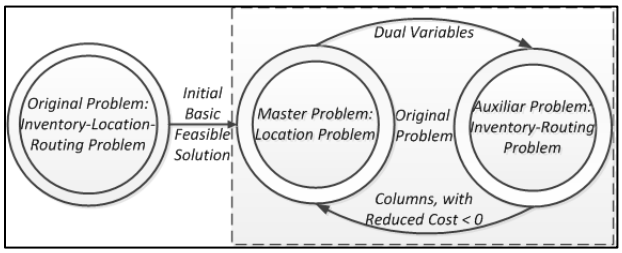

Figure 1: Column Generation

4.2 Graph Topology

For the CG Formulation we propose the next topology. Parting from the original ILRP formulation, we have a set *I* of depot candidates, a set *J* of customers and a set *P* of periods. Customers are indexed from 1 to *m*, periods from 1 to *p* and depots from 1 to *n*. Now the transformation can be represented by a Graph *G=(V,E)*, where V is the set of vertices, including depot nodes *I*, Customer-Nodes *C* and Auxiliary-Nodes *A*; $|V| = |J||P||P|+(|P|+1)|I|$ and $E=\{ (i,j):i\neq j\}$ is the set of undirected arcs connecting nodes in the graph.

The Customer-Nodes *C* is a set of nodes, associated with the decision of replenish a particular customer *j*, during period *p* to satisfy the demand in period *r* and is defined as follows: $C = \{(j,p,r) | j \in J, p \in P, r \in P\}$. The Auxiliary-Nodes *A* are used to preserve the same depot for a route during all periods. In other words, when the route ends a period, it goes directly to the Auxiliary-Node that represents the original depot. And finally let Ω be the set of feasible routes in *G*, starting and ending at the same depot with delivery information. The idea of the Graph transformation can be clearly seen in Figure 2.

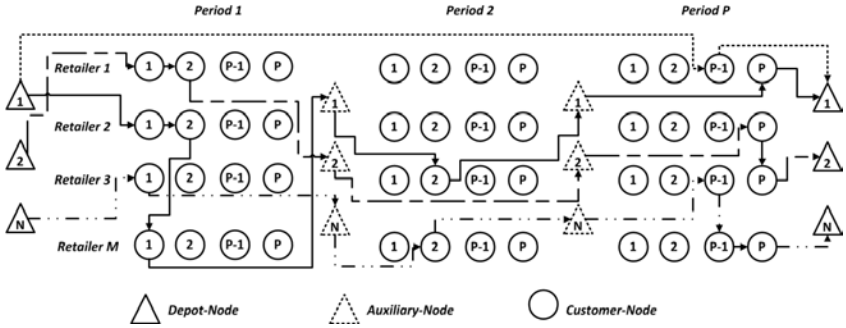

Figure 2: Graph Topology of the ILRP

Associated with each arc *(i,j)* is a cost of traveling $dist_{ij}$ from node $i \in V$ to node $j \in V$. Since one of the decisions to make with this model is the number and location of the depots to be opened, each depot $i \in I$ has an opening cost o_i. Each Customer-Node $j \in C$ faces a demand d_j. Besides, each item held in inventory at the end of a period implies a holding cost h_j for each customer $j \in J$ and there is also a backlogging cost g_j. Because there is a penalty cost for the unmet demand that is backlogged, the backlogging cost is greater than the holding cost. Besides, there is also a cost s_{jr} of not supplying customer $j \in J$ for period $r \in P$.

In addition, there is a limited number of vehicles *U* in the distribution network. There is also a fixed cost *F* for using each vehicle and each one has the same capacity Cap. In addition each customer has a limited storage capacity q_j.

For the Graph transformation we incorporate the inventory costs in each arc *(i,j)* as each Customer-Node is associated with the decision to replenish a particular customer *j*, during period p to satisfy the demand in period *r*, and consequently the use of an arc *(i,j)* also is associated with the inventory holding and backlogging costs h_j and g_j. As a result, the total cost c_{il} of the arc *(i,l)* $\in E$, including inventory costs and traveling costs is defined as follows:

$$c_{il} = \begin{cases} dist_{il} + d_{j_1} h_{j_1}(r_2 - r_1) & l = (j_1, p, r_1), i = (j_2, p, r_2) \;\; r_2 \geq r_1 \\ dist_{il} + d_{j_1} g_{j_1}(r_1 - r_2) & l = (j_1, p, r_1), i = (j_2, p, r_2) \;\; r_1 > r_2 \end{cases}, \forall\, i \in V, l \in V$$

4.3 Master Problem

For the Master Problem M we define the binary variables $y_{ik} = 1\ iff$ path k $\in \Omega$, that starts in depot $i \in I$ is used, $z_i = 1\ iff$ depot $i \in I$ is opened and $w_{jr} = 1\ iff$ customer $j \in J$ is never supplied for period $r \in P$. Now the Master Problem can be formulated as follows:

$$\boldsymbol{M}: min\ \sum_{i \in I} \sum_{k \in \Omega} y_{ik}(c_{ki} + F) + \sum_{i \in I} z_i o_i + \sum_{j \in J} \sum_{r \in P} w_{jr} s_{jr} \tag{1}$$

subject to

$$\sum_{i \in I} \sum_{k \in \Omega} y_{ik} \leq U \tag{2}$$

$$\sum_{k \in \Omega} y_{ik} \leq z_i |J||P| \;\; \forall\, i \in I \tag{3}$$

$$\sum_{i \in I} \sum_{k \in \Omega} y_{ik} a_{jrk} + w_{jr} \geq 1 \quad \forall\, j \in J, \forall\, r \in P \tag{4}$$

$$\mathrm{y_{ik}} \in \{0, 1\},\ \forall\, i \in I, k \in \Omega \tag{5}$$

$$z_i \in \{0, 1\},\ \forall\, i \in I \tag{6}$$

$$w_{jr} \in \{0, 1\},\ \forall\, j \in J, r \in P \tag{7}$$

The objective function (1) of the Master Problem is to minimize the sum of depot opening costs, routing costs and the costs of not satisfying demands. Constraints (2) ensure that there is a maximum of *U* routes in the problem. Constraints (3) guarantee that a depot has to be opened, if a route, starting in this depot is used in the solution. Constraints (4) establish that the demand for a period is satisfied in any given period or this demand is never satisfied. Constraints (5), (6) and (7) specify the binary variables for this formulation.

4.4 Auxiliary Problem

The objective of the Auxiliary Problem **Aux** is to find a route with negative reduced cost in order to introduce this route in the set Ω of the Master Problem. It can be classified as a Shortest Path Problem with capacity constraints (SPPCC).

For this purpose it is necessary to define the dual variables associated with the Master Problem constraints. In the model λ^0 is the nonnegative dual variable associated with the maximum number of routes in the problem. λ_i^1 is the nonnegative dual variable associated with the opening of depot *i* (Constraints (3)). Finally λ_{jr}^2 is the nonnegative

dual variable associated with the satisfaction of the demand for period r of customer j. The Auxiliary Problem uses the dual variables to generate a new column.

In addition, we define the binary variables $b_{ij} = 1\ iff\ arc\ (i,j) \in E$ is used in the solution, $a_{jr}\ 1\ iff$ customer $j \in J$ is visited in period $r \in P$ and $depos_i = 1\ iff$ the route belongs to depot i.

Besides, for the construction of some constraints in the formulation of the Auxiliary Problem we define sub-sets, explained as follows: For each customer j there is sub-set B_j of nodes, that represents the decision of replenish customer j during period p to satisfy the demand in period r. The mathematical definition of this sub-set is as follows:

$$B_j = \{(j,p,r)|p \in P, r \in P\} \subseteq C \qquad C = \{\cup_{j\in J} B_j\}$$

Another sub-set created for this formulation is D_p. For each period p, there is a sub-set of nodes representing the customers and the periods supplied in p, for all customers and is mathematically defined as follows:

$$D_p = \{(j,p,r)|j \in J, r \in P\} \subseteq C \qquad C = \{\cup_{p\in P} D_p\}$$

Besides this two sub-sets, R_{jr} and L_{jp} were also created. For each period r that the demand of each customer j has to be satisfied, there is a sub-set R_{jr} of periods in which the customer is supplied. In addition, for each period a customer is supplied, there is a sub-set of nodes L_{jp} representing for which period r the customer is supplier (such that r is bigger than p). For a better understanding of this two sub-sets, we present them again in a mathematical form:

$$R_{jr} = \{(j,p,r)|\ p \in P\} \subseteq B_j$$

$$L_{jp} = \{(j,p,r)|r \in P, r > p)$$

Now with the new sub-sets and variables defined, the formulation is as follows:

$$\boldsymbol{Aux}:$$

$$min \sum_{i\in V}\sum_{l\in V} b_{il}\, c_{il} - \sum_{j\in J}\sum_{r\in P} a_{jr}\lambda_{jr}^2 - \sum_{i\in I} depos_i \lambda_i^1 - \lambda^0 \tag{8}$$

$$subject\ to$$

$$\sum_{j\in V} b_{ji} = \sum_{j\in V} b_{ij} \quad \forall\, i \in V\backslash(A\backslash A_{|P|}) \tag{9}$$

$$\sum_{j\in D_p}\sum_{i\in V} b_{ij} d_j \le Cap \quad \forall\, p \in P \tag{10}$$

$$\sum_{i\in V}\sum_{\substack{p\in P\\ |p<t}}\sum_{l\in L_{jp}} b_{il} d_l \le q_j\ \forall\, j \in J, \forall\, t \in P \tag{11}$$

$$\sum_{i\in V}\sum_{l\in R_{jr}} b_{il} = a_{jr}\ \forall\, j \in J, \forall\, r \in P \tag{12}$$

$$\sum_{j\in V} b_{ij} = \sum_{j\in V} b_{lj} \quad \forall\, i \in I, l \in (A_i \cap A_{|P|}) \tag{13}$$

$$\sum_{i\in I}\sum_{j\in V} b_{ij} = 1 \tag{14}$$

$$\sum_{j\in V} b_{ij} = depos_i \quad \forall\, i \in I \tag{15}$$

$$b_{ij} \in \{0,1\}, \qquad \forall\, i \in I, \forall\, j \in V \tag{16}$$

$$a_{jr} \in \{0,1\} \; \forall \, j \in J, \forall \, r \in P \qquad (17)$$

$$depos_i \in \{0,1\}, \; \forall \, i \in I \qquad (18)$$

The objective Function (8) is to minimize the reduced cost of the routes. The Auxiliary Problem generates a route by minimizing the total costs of the arcs minus the benefit from the inclusion of a route, from the opening of a depot and from the satisfaction of a demand. Constraints (9) are flow conservation constraints. Constraints (10) and (11) concern vehicle´s capacity and customer´s storage capacity respectively. Constraints (12) are the definition and the relationship between b_{ij} and a_{jr}. Constraints (13) ensure that the route must end in the corresponding Auxiliary Node. Constraints (14) ensure that each route starts from a depot. Constraints (15) establish if the route starts from depot *i* or not. Finally, with constraints (16)-(18) the formulation involves binary variables.

5. Proposed Algorithm for the Auxiliary Problem

Although the CG is used to reduce the complexity of solving problems with large number of variables, the Auxiliary-Problem cannot be solved in a reasonable amount of time by commercial solvers. The Dijkstra´s algorithm is used to solve shortest path problems, but it has the requirement of only accepting non-negative costs in the arcs [12]. As we are searching for arcs with negative reduced cost it is not possible to implement this algorithm in our model. Therefore, we propose a method based on depth search and backtracking algorithm for solving the Auxiliary Problem in an efficient way.

Backtracking is an algorithm for finding a solution to a problem among all available options. Gurari [6] explains that when it is necessary to make a series of decisions, you do not know what you really have to choose and it is required to make sequence of choices to find a solution to your problem, backtracking is a good algorithm for solving this type of problems. It tries different sequences of decisions, until finding one with a valid solution. It starts with an empty solution, then it adds nodes to it, in our case to the route. If it identifies that the node provides an invalid solution, the algorithm removes this candidate and searches for other possible nodes that can complete a valid solution. It explores the graph with a depth search [4].

Nevertheless, in this model, if the algorithm does not find a route with negative reduced cost, it proceeds with the Simplex Algorithm, solving the mathematical model described in Section 4. This guarantees that the model finds the optimum, either with the backtracking algorithm or with the Simplex algorithm. The backtracking algorithm is described as follows:

The algorithm starts at a depot *i*, exploring the nodes from the first period. Then it searches for the node that represents the minimum negative reduced cost for the route. Now it proofs whether it is possible to go directly from this node to the depot having negative reduced cost. If it doesn't then the algorithm adds this node to a list of backtracking candidate nodes and it also adds the node to the route. It continues to explore the next nodes in this period, and searches for the next one with a negative reduced cost. It repeats the same procedure concerning the backtracking list until there does not exit a node with negative reduced cost in this period. As soon as it finds a node that can go directly to the depot with a negative reduced cost, all the nodes from the backtracking list are removed and the list remains empty. Now if it does not find nodes with negative reduced cost and the backtracking list is nonempty, it removes all the elements contained in this list from the partial route and adds the first element to a set in order not to add this node as the next element in the route. It continues searching the next node to be added and once it finds one, it removes the elements from this set. Then the procedure continues as previously described. Once there are no more nodes with negative reduced cost and the backtracking list is empty, it continues with the next period doing the same steps, iterating for every period.

All these steps are repeated for every depot, saving one route per depot. At the end it chooses the route with the minimum reduced cost. The algorithm is showed in Figure 3:

$\boldsymbol{For}\ \boldsymbol{v} \in \boldsymbol{V} \rightarrow$
  $\boldsymbol{For}\ \boldsymbol{u} = (\boldsymbol{j}, \boldsymbol{p}, \boldsymbol{r}) \in \boldsymbol{C} \rightarrow$
    $cost_{vu} := c_{vu} + \lambda^2_{jr}$
  $\boldsymbol{rof}$
$\boldsymbol{rof}$
$\boldsymbol{For}\ \boldsymbol{i} \in \boldsymbol{I} \rightarrow$
  $current,\ sumcost,\ used,\ min,\ backtracking,\ last,\ former,\ nodecap := i,\ 0,\ 0,\ -1,\ 0,\ i,\ [],\ []$
  $route_i := <>$

```
For p ∈ P →
    done := false
    review := {}
    Do done = false →
        found := false
        For v = (j, p, r) ∈ (D_p \ review) →
            if v < 0 →
                v := current
            fi
            if 0 > cost_{current,min} > cost_{current,v} and sumcost + cost_{current,v} < 0 and used + d_v < Cap and nodecap_j + d_v < q_j →
                min, found := v, true
            fi
        rof
        if found = true and sumcost + cost_{current,min} + cost_{min,−i} − λ_i^1 < 0 →
            backtracking += 1
            former_{backtracking} := current
        else →
            backtracking, former := 0, []
        fi
        if found = true →
            route_i += < min >
            sumcost, used += cost_{current,v}, d_min
            last, current, done, review, min := current, min, false, {}, −1
            if rmin > p →
                nodecap_jmin += d_min
            fi
            For v ∈ V →
                For u ∈ B_jmin →
                    cost_vu := MAX_INT
                rof
            rof
        fi
        if found = false and backtracking > 0 →
            Do backtracking > 0 →
                last := former_backtracking
                For v ∈ V →
                    For u = (jcurrent, pcurrent, rcurrent) ∈ B_jcurrent →
                        cost_vu := c_vu + λ^2_{jcurrent,rcurrent}
                    rof
                rof
                route_i −= < current >
                sumcost, used −= cost_{last,current}, d_current
                if jcurrent > p →
                    nodecap_jcurrent −= d_jcurrent
                fi
                if backtracking = 1 →
                    review += {current}
                fi
                backtracking, current, found := rev − 1, last, true
            od
        fi
        if found = false →
            sumcost, used, done, review, current := cost_{current,−|I|*p−i}, 0, true, {}, −|I| * p − i
            route_i += < current >
        fi
    od
rof
rof
```

Figure 3: Backtracking algorithm

6. Computational Experiments and Comparative Evaluations

The proposed CG based on the algorithm previously describe was coded using Mosel-Language and was run on a PC with an Intel Core™ 2 CPU 6300 (1.86 GHz) and 4 GB RAM memory.

As the parameter selection influences the efficiency of the computational results [19], different problem instances were designed in order to measure the performance of the CG approach and the algorithm used for the Auxiliary Problem. Considering all the combinations between 4, 6, 8 customers, 3, 5, 7, 9 periods and 6 depots, 12 different instances were created. Besides, the demand for each customer and for each period of time is randomly selected from a uniform distribution U[30,100]. The location (x,y) of each depot and each customer is also randomly chosen from a uniform distribution U[0,100]. The distances between customers and depots are calculated using the Manhattan distance function. The holding costs and backlogging costs are randomly selected from uniform

distributions as well: U[1,10] and U [5,15] respectively. Besides, the cost of not supplying a customer for a period is selected from a uniform distribution: U [300,2000]. The fixed cost F and the capacity Cap of a vehicle are respectively 200 and 300, and the maximum number of vehicles U is chosen to be 7.

Table 1 shows the results for different created instances. The maximum range of customers and periods that were solved with this method is shown. We tried to solve bigger instances, but during the computational experiments the method consumed all the RAM memory, so that it was impossible to finish the problem for them. However RAM space can easily be freed up by implementing an algorithm to lose previously unused columns, thus allowing our method to solve problem instances up to 1500 nodes within less than 90 hours. This estimation is based on the figure 4, there is shown the estimated polynomial equation, so that one can calculate an approximation of the time it would take to solve the instance given the nodes it has.

Table 1. Comp. results for different instances

J(Customers)	P (Periods)	Time	Iterations	Nodes
4	3	3,401 sec	104	60
5	3	4,056 sec	92	69
6	3	8,486 sec	146	78
8	3	12,511 sec	148	96
10	3	36,972 sec	313	114
4	5	47,205 sec	326	136
6	5	4,537 min	567	186
8	5	13,656 min	860	236
4	7	5,953 min	475	244
6	7	48,48 min	1256	342
4	9	1,361 h	1483	386

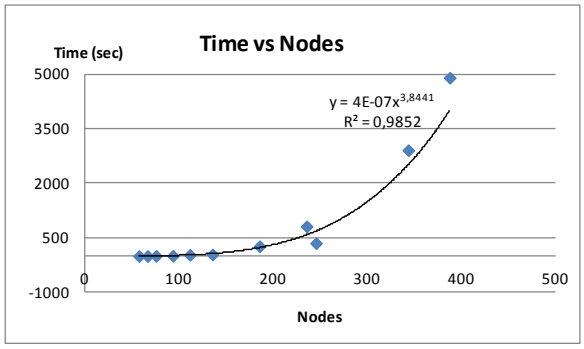

Figure 4: Computational time vs. Nodes

7. Conclusions

In this paper we have introduced an heuristic CG approach to solve the ILRP. As it is a problem that has not been extensively explored, it still remains as a potential research opportunity. Therefore, we present a CG for this problem, where the original problem (ILRP) is decomposed into a Master Problem (Location Problem) and an Auxiliary Problem (Inventory-Routing Problem). For solving the Auxiliary Problem we have developed an heuristic based on a backtracking algorithm, that searches columns with negative reduced costs, but not necessary being the ones with the minimum reduced cost. However, it is important to mention that the model explores all the search space, until there is no column with negative reduced costs, because it uses alternatively the Simplex Algorithm to search for the column with minimum negative reduced cost, when the heuristic does not provide one.

The CG proposed was tested on different random generated instances, varying the number of customers and also the number of periods. Computational results and comparisons are presented. The model solves the problem for small size instances in a reasonably period of time. Even though this size is small, we have chosen them so that they can represent a real world problem. The computational experiments have also showed that the RAM memory is a limiting resource for solving the problem based only on this approach, and therefore it was not possible to measure the performance of bigger instances. However this problem can easily be solved with a non time-expensive algorithm. It was also evident that because of the topology of the graph the computational time increases in a polynomial way as the number of customers or periods increase. This fact increments also the number of feasible routes and therefore the computational time increases in this way. Though the computational time increases in a rapidly manner, the method proposed for solving the ILRP guarantees the optimum of the problem, by mixing the heuristic algorithm with the Simplex algorithm. Besides, the heuristic algorithm permits to solve the problem in a reasonably amount of time.

For future work, it would be interesting to compare the proposed CG with another one implemented in the literature of the ILRP in order to measure the efficiency of what we propose. It would be also interesting to implement an algorithm to free up some RAM so that it would be possible to measure the performance of much bigger instances. Besides, as Operational Research is oriented in applications based on real world problems, the proposed model can be adapted to different realistic problems.

References

1. Ahmadi Javid, A., and Azad, N. (2010). incorporing location, routing and inventory decisions in supply chain network design. *Transportation Research*, Part E, 582-597.
2. Ballou, R. (2004). Logística. Administración de la cadena de suministro. (5ta ed.). México: Person Educación.
3. Choi, E., and Tcha, D.-W. (2007). A column generation approach to the heterogeneous fleet vehicle routing problem. *Computers and Operations Research*, 34, 2080–2095.
4. Cormen, T. H., Leiserson, C. E., Rivest, R. R., and Stein, C. (1990). Introduction to Algorithms (2nd Edition ed.). McGraw-Hill.
5. Feillet, D. (2010). A tutorial on column generation and branch-and-price for vehicle routing problems. *Journal of Operations Research*, 8:407-423.
6. Gurari, E. (1999). Backtracking algorithms "CIS Data Structures: Chapter19: Backtracking Algorithms".
7. Jin, M., Liu, K., and Eksioglu, B. (2008). A column generation approach for the split delivery vehicle routing problem. *Operations Research Letters*, 3, 265-270.
8. Laporte, G., Nobert, Y., and Taillefer, S. (1988). Solving a Family of Multi-Depot Vehicle Routing and Location-Routing Problem. *Transportation Science*, 22(3), 161-172.
9. Lin, C., and Liu, S. (2005). A heuristic method for the combined location routing and inventory problem. *The International Journal of Advanced Manufacturing Technology*, 26(4), 372-381.
10. Liu, S. C., and Lee, S. B. (2003). A two-phase heuristic method for the multi-depot location routing problem taking inventory control decisions into consideration. *The International Journal of Advanced Manufacturing Technology*, 22(11), 941-950.
11. Medaglia, A. L. (2002). Programación A Gran Escala: Generación De Columnas. Notas de Clase.Optimización Avanzada.
12. Moshe, S. (2006). Dijkstra's algorithm revisited: t dynamic programming connexionh. Controland Cyberneticsvol. 35.
13. Nagy, G., and Salhi, S. (2007). Location-Routing: Issues, models and methods. *European Journal of Operational Research*, 177(2), 649-672.
14. Nguyen, V., Prins, C., and Prodhon, C. (2010). GRASP with learning process for a two-Echelon Location Routing Problem. MOSIM'10 (8th International Conference of Modeling and Simulation).
15. Oppen, J., Lokketangen, A., and Desrosiers, J. (2010). Solving a rich vehicle routing and inventory problem using column generation. *Computers and Operations Research*, 37, 1308-1317.
16. Orman, Z. (2005). The Inventory-Routing Problem with Inventory Control Considerations: Formulation, Solutio Method and Computational Results.
17. Sambola, M. A. (2003). Models and algorithms for location-routing and related problems. Tese de Doutorado, Universidade Politécnica da Catalunha, España.
18. Solyali, O., Cordeu, J.-F., and Laporte, G. (2010). Robust Inventory Routing under Demand Uncertainty. Centre interuniversitaire de recherche ser les réseaux d'entreprise, la logistique et le transport. CIRRELT-2010-17.
19. Yu, V. F., Lin, S.-W., Lee, W., and Ting, C.-J. (2010). A simulated annealing heuristic for the capacitated location routing problem. *Computers and Industrial Engineering*, 58, 288-299.