An Algorithm for a 3-Level Location-Routing Problem

Leonardo Souza Ribeiro
Production Engineering Dept., COPPE/UFRJ
Cidade Universitária, CT, Bloco F, Sala 103, RJ, Brasil
Petrobras
Avenida Almirante Barroso 81, RJ, Brasil
leonardosribeiro@petrobras.com.br

Roberto Diéguez Galvão
Production Engineering Dept., COPPE/UFRJ
Cidade Universitária, CT, Bloco F, Sala 103, RJ, Brasil
galvao@pep.ufrj.br

ABSTRACT

Location-routing problems (LRP’s) are hard combinatorial optimization problems, for which both exact and heuristic methods are available. We consider a 3-level problem, where products are transported from factories to depots and from there to clients, according to routes designed in the solution procedure. After a brief literature survey of the subject, we give a set partitioning formulation for this problem, which is solved by column generation. Columns correspond to routes that are generated by a constrained shortest path algorithm. In order to generate integer solutions, we develop a branch-and-price algorithm for this problem. Computational results are given for 3 sets of data adapted from the literature, with the number of clients varying from 50 to 100.

Keywords: Location; routing; branch-and-price.

Main Area: Combinatorial Optimization.

1. INTRODUCTION

The distribution planning of goods is based on determining the optimal configuration of the network that will transport the merchandise from manufacturing facilities to the final consumer, usually through intermediate storage facilities (distribution depots). The strategic part of the planning procedure consists in determining the number and location of the distribution depots and the allocation of clients to these depots. Its strategic nature derives from the fact that location decisions imply sizeable investments and are usually related to organizational restructuring. Hence, planning based on sound quantitative analyses is a must for the survival of the enterprise.

The location of the depots determines the flow of goods from a manufacturing facility to the final client. In a 3-level distribution system the location of the depots determines the spatial configuration of the system. In such framework the distribution costs are the following: (i) transportation costs from factories to depots; (ii) depot handling costs; (iii) inventory holding costs in the depots; and (iv) delivery costs from depots to the final clients.

Assuming that client demands must be satisfied and capacity constraints must be taken into consideration, the objective is to determine the number and locations of the depots in order to minimize the costs above. This location problem has been addressed in the corresponding literature in a variety of forms, through the use of different optimization models. However, it is
usually assumed that every client is independently supplied from the corresponding depot – the
construction of delivery routes supplying more than one customer (and the corresponding effects
in distribution costs) is usually not considered.

Transportation between the first and second levels of the distribution system is usually
made in bulk quantities using containers. Estimating these transportation costs is relatively easy,
since it suffices to know bulk transportation rates between factories and depots. It is also not
difficult to estimate handling and inventory holding costs at the depots. The transportation costs
between levels two and three of the distribution system are however harder to estimate.

In conventional location models these costs are modeled assuming that every client is
supplied independently – a simplification that may conduce to erroneous estimation of these
costs. Depending on the application, this may induce to faulty decision-making at the strategic
level (location of depots). Thus the need to consider, in specific cases, the more complex
combined location-routing problem. In a recent survey Nagy and Salhi (2007) consider the
location-routing problem a research area within location analysis, with the property of paying
special attention to issues of vehicle routing. They define location-routing as “location planning
with tour planning aspects taken into account”.

Nagy and Salhi (2007) note that frequently both academics and practitioners ignore the
interrelation that exists between the two problems and solve location problems without paying
attention to routing considerations, incurring in the danger of sub-optimizing. We are in
agreement with the sub-optimization danger pointed out by these authors, but do not share their
hierarchical viewpoint of location-routing problems, substantiated in the definition of Nagy and
Salhi given above. Depending on the circumstances, we feel that treating both problems in the
same level (such as in the iterative solution methods) may be more appropriate.

Location-routing problems are hard combinatorial optimization problems, since they
combine problems that are NP-hard on their own. Exact algorithms developed so far can only
solve to optimality problems with a small number of clients. The alternative to handle larger
problems is to resort to the development and use of heuristic algorithms.

The plan of the present paper is as follows. In Section 2 we make a very brief literature
survey of location-routing problems. Then, in Section 3, we give a set partitioning formulation
for this problem, and show how the relaxed problem can be solved by column generation;
columns correspond to routes that are generated by a constrained shortest path algorithm. We
then show, in Section 4, the branching rules used in the branch-and-price algorithm.
Computational results are given in Section 5. A Conclusions Section finalizes the paper.

2. BRIEF LITERATURE SURVEY

Balakrishnan et al. (1987), Laporte (1988) and Min et al. (1998) present earlier surveys
on combined location-routing problems. As already mentioned, a complete and very recent
review of the subject is made by Nagy and Salhi (2007).

The solution of a combined location-routing problem requires that the following
interdependent tasks be carried out: (1) Location of the depots; (2) Allocation of clients to the
chosen depots; and (3) Determination of routes to deliver the products stored in the depots to the
respective clients. In the case of a 3-level distribution system, as in the present paper, the optimal
flow from factories to depots must be also determined.

Heuristic methods available for the solution of combined location-routing problems
examine the interdependent tasks mentioned above and attempt to coordinate the respective
outcomes, seeking to obtain high quality solutions for the combined problem. These solution
procedures differ among themselves by the order in which the interdependent location and
routing problems are solved. They may also differ by the method through which the coordination
of tasks is conducted: sequential, iterative or hierarchical.

Perl and Daskin (1985) present a heuristic that iterate between the location and routing
phases. The location phase is formulated as a Warehouse Location Problem (WLP). The routing
phase uses saving-type heuristics generalized for multiple depots. The phases in which WLP and
the routing heuristic are solved interact among themselves, until the improvement between two consecutive runs of the iterative procedure is less than a pre-determined value \( \varepsilon > 0 \).

Bookbinder and Reece (1988) proposed a *locate-first-route-second* heuristic method. They initially solve a warehouse location problem, which determines the location of the depots and the allocation of clients to open depots. These data allow the solution of a routing problem for each open depot. The routes’ costs, obtained in this second phase, are split among the respective clients, and this provides an estimation of the transportation costs between depots and clients, to be used in a new iteration of the location problem. Bookbinder and Reece used Bender’s Decomposition to solve the location problem, which they applied to a formulation given by Geoffrion and Graves (1974).

Jacobsen and Madsen (1980) carried out a study for a newspaper distribution company in Denmark. They examined a 3-level problem, in which newspapers are sent from printing facilities to distribution centers, from where they are delivered to readers through multi-client routes.

Srivastava (1993) proposed three heuristics for solving a location-routing problem. The first of them (CLUST) generates several clusters of clients obtained from a supporting minimal spanning tree. Clients are grouped in the desired number of clusters using a *density search cluster* technique. The number of depots to be located is equal to the number of formed clusters, each depot being located close to the centroid of the respective cluster. Routes are then obtained using the “sweep” algorithm of Gillet and Miller (1974). Several configurations may be evaluated by varying the number of clusters under consideration.

The second heuristic proposed by Srivastava (SAV1) uses a modified version of the savings routing algorithm of Clarke and Wright (1964), adapted for routing from several depots by Tillman (1969). This method starts with all potential depots open; a DROP procedure identifies the depots to be closed, one at a time, based on the values of the economies. The third heuristic (SAV2) starts with all potential depots closed and proceeds to open some of them, one at a time, based again on the values of economies.

Nagy and Salhi (1996) propose hierarchical methods they call *nested* for the solution of location-routing problems. The location algorithm is based on tabu search and an add/drop/shift neighborhood. After each move the routing solution is fully evaluated using a multi-depot routing (VRP) algorithm.

Exact methods for location-routing problems are usually based on a mathematical programming formulation of the problem. However, due to their computational complexity, exact methods can only solve relatively small instances of these problems.

Laporte and Nobert (1981) proposed a branch-and-bound approach to locate a single depot and used a fixed number of vehicles to perform the delivery routes. Laporte et al. (1983) considered locating several depots with or without fixed costs and with or without an upper limit on the number of depots. The sub-tour elimination constraints were relaxed and the facilities were located by adapting Miliotis’s (1978) REVERSE algorithm.

For a multi-depot LRP with vehicle capacity constraints, Laporte (1986) formulated an integer program that includes sub-tour elimination constraints and chain barring constraints. To overcome the potentially exponential number of constraints, the author developed a constraint-relaxation algorithm in which he initially relaxes the sub-tour elimination, chain barring and integrality constraints. As the algorithm progresses, it identifies and adds violated constraints and then branches when it identifies no other violated constraints.

Recently, Berger et al. (2007) proposed a branch-and-price approach for a LRP with 2 levels (i.e. they considered only the product flows from the depots to the clients) and distance constraints. They present a new formulation for the LRP with distance constraints. The pricing problem, which decomposes into a set of elementary shortest path problems with a single resource constraint, is solved using the algorithm of Feillet et al. (2004). Integer solutions are found by the partitioning rules proposed by Ryan and Foster (1981).

### 3. A SET-PARTITIONING FORMULATION FOR A 3-LEVEL LOCATION-ROUTING PROBLEM
We now give a mathematical programming formulation for a 3-level LRP. Let:

**SETS**
- \(J\): Set of potential depots;
- \(I\): Set of clients;
- \(K\): Set of all routes;
- \(T\): Set of factories;

**PARAMETERS**
- \(N\): number of clients;
- \(C_{ij}\): distance between points \(i\) and \(j\), \(i,j \in I \cup J\);
- \(F_j\): installation cost of depot \(j \in J\);
- \(d_i\): demand of client \(i \in I\);
- \(d_{tuj}\): distance between nodes \(u\) and \(q\), \(u,q \in I \cup J\);
- \(Q_k\): capacity of vehicle in route \(k \in K\);
- \(\hat{C}_{ij}\): unit transportation cost from factory \(t \in T\) to depot \(j \in J\);

**DECISION VARIABLES**
- \(x_{ijk}\): =1 if point \(i\) precedes point \(j\) in route \(k \in K\) (: \(i,j \in I \cup J\)); =0 otherwise;
- \(y_j\): =1 if a depot is located at \(j \in J\); =0 otherwise;
- \(z_{ij}\): =1 if client \(i \in I\) is allocated to depot \(j \in J\); =0 otherwise;
- \(v_{tj}\): variable that indicates the number of items sent from factory \(t \in T\) to depot \(j \in J\);
- \(U_{ik}\): auxiliary variable for sub-tour elimination in route \(k \in K\).

**MATHEMATICAL MODEL**

\[
\text{Min} \sum_{t \in T} \sum_{j \in J} \hat{C}_{tj} v_{tj} + \sum_{j \in J} F_j y_j + \sum_{i \in I \cup J} \sum_{j \in J} \sum_{k \in K} C_{ij} x_{ijk} \quad (1)
\]

s. to
\[
\sum_{k \in K} \sum_{j \in J} x_{ijk} = 1, i \in I, \quad (2)
\]
\[
\sum_{j \in J} x_{ijk} - \sum_{j \in I \cup J} x_{ij} = 0, k \in K, i, l \in I \cup J, \quad (3)
\]
\[
\sum_{i \in I} \sum_{j \in J} x_{ijk} \leq 1, k \in K, \quad (4)
\]
\[
\sum_{u \in U} \sum_{i \in I \cup J} x_{iuk} \leq Q_k, k \in K, \quad (5)
\]
\[
\sum_{i \in I \cup J} \sum_{j \in J} x_{ijk} d_{tuj} \leq D_k, k \in K, \quad (6)
\]
\[
U_{ik} + N x_{ijk} \leq N - 1, i \in I, k \in K, \quad (7)
\]
\[
\sum_{u \in U} (x_{ijk} + x_{iuk}) - z_{ij} \leq 1, i \in I, j \in J, k \in K, \quad (8)
\]
\[
\sum_{i \in T} v_{tj} - \sum_{i \in I} z_{ij} d_i = 0, j \in J, \quad (9)
\]
\[
z_{ij} - y_j \leq 0, i \in I, j \in J, \quad (10)
\]
\[
x_{ijk}, y_j, z_{ij} \in \{0,1\}, i \in I, j \in J, k \in K, \quad (11)
\]
\[
v_{tj} \geq 0, t \in T, j \in J, \quad (12)
\]
\[
U_{ik} \geq 0, i \in I, k \in K. \quad (13)
\]
Objective function (1) minimizes the sum of transportation costs from factories to depots, the installation costs of the depots and the transportation costs in the delivery routes from depots to clients. Restrictions (2) ensure demand satisfaction for all clients and establish that each client must be served by only one depot. Constraints (3) are the flow conservation constraints for each route and constraints (4) ensure that each route will be executed not more than once. Restrictions (5) are the vehicle capacity constraints and restrictions (6) establish the maximum length of a route. Restrictions (7) are the sub-tour elimination constraints. Constraints (8) and (10) establish that a client will be allocated to a depot only if there is a route leaving this depot that includes the client and restrictions (9) establish that the number of items sent from factories to depots must be equal to the number of items delivered to the clients. Finally, constraints (11)-(13) establish the nature of the decision variables.

An equivalent formulation of this problem may be obtained via set-partitioning. Keeping the definition of sets $I$, $J$ and $T$ given before, we define a graph $G = (N, A)$, where $N = I \cup J$ is the set of nodes of the graph and $A = N \times N$ is the corresponding set of arcs. A route $k$ associated to a depot $j$ visiting one or more clients is feasible if its total length does not exceed $MaxDist$ and its total load does not exceed $MaxDem$. Let $P_j$ be the set of such routes associated with a given depot $j \in J$. The cost of a route $k \in P_j$ is the sum of the costs of the arcs $(i,j) \in A$ associated with this route. The mathematical model may be expressed in the following way, using the notation already defined and the new notation shown below.

**PARAMETERS**

- $a_{ijk}$: = 1 if route $k \in P_j$ associated to depot $j \in J$ visits client $i \in I$; =0 otherwise;
- $C_{jk}$: cost of route $k \in P_j$ associated with depot $j \in J$;

**DECISION VARIABLES**

- $x_{jk}$: = 1 if route $k \in P_j$ is chosen; = 0 otherwise.

**MATHEMATICAL MODEL**

\[
\text{Min} \sum_{i \in I} \sum_{j \in J} \tilde{C}_{ij} y_{ij} + \sum_{j \in J} F_j y_j + \sum_{j \in J} \sum_{k \in P_j} C_{jk} x_{jk} \tag{14}
\]

s.t

\[
\sum_{j \in J} \sum_{k \in P_j} a_{ijk} x_{jk} = 1, i \in I, \tag{15}
\]

\[
y_j - \sum_{k \in P_j} a_{ijk} x_{jk} \leq 0, i \in I, j \in J, \tag{16}
\]

\[
\sum_{i \in I} \sum_{j \in J} v_{ij} - \sum_{i \in I} \sum_{k \in P_j} a_{ijk} x_{jk} d_i = 0, j \in J, \tag{17}
\]

\[
x_{jk}, y_j \in \{0,1\}, j \in J, k \in P_j, \tag{18}
\]

\[
v_{ij} \geq 0, t \in T, j \in J. \tag{19}
\]

Objective function (14) minimizes the sum of transportation costs from factories to depots, the installation costs of the depots and the costs of the delivery routes. Constraints (15) ensure demand satisfaction and establish that each client is served by exactly one delivery route. Restrictions (16) locate a depot at $j \in J$ only if a route allocated to this depot is chosen. Constraints (17) establish that the number of items sent from factories to depots must be equal to the number of items delivered to the clients. Finally, constraints (18)-(19) define the nature of the decision variables.

Problem (14)-(19) has fewer constraints than problem (1)-(13), but a large number of variables $x_{jk}$. Therefore, for practical purposes, the explicit enumeration of all possible routes would be infeasible. The column generation method is appropriate in this case. For the set-
partitioning formulation the routes (columns) may be iteratively generated through a pricing sub-problem, until routes with negative reduced costs are no longer generated by the sub-problem.

The restricted master problem (Dantzig-Wolfe decomposition) is given by (14)-(19) above, with the binary variables relaxed to being non-negative; optimal integer solutions may be then obtained via a branch-and-price algorithm. In the set partitioning reformulation of problem (1)-(13), constraints (3)-(7) are implicitly satisfied in the pricing sub-problem, which generates feasible routes utilizing a constrained shortest path algorithm.

Figure 1 shows a basic scheme for the branch-and-price algorithm. The search tree is initialized with the root node, using an initial set of columns generated by the algorithm of Tillman and Cain (1972), for routing vehicles from multiple depots. The linear relaxation of the restricted master problem is then solved by column generation. In order obtain an upper bound at the root node, we apply a standard branch-and-bound procedure to solve the integer program over the set of columns generated during the solution of the relaxed restricted master problem. If the solution is integer then it is a feasible solution for the original master problem; it is therefore compared with the current incumbent solution, as in standard branch-and-bound algorithms. If the LP solution is not integer, then branching occurs to cut off the current fractional solution. At the end of the search procedure the best integer solution is the optimal solution for the original problem.

3.1 The Pricing Sub-Problem

The pricing sub-problem seeks to determine whether there exists a route \( k \in K \) associated to a depot \( j \in J \) with a negative reduced cost. In order to compute the reduced costs we use variables \( \pi_i \), associated with constraints (15), variables \( \mu_{ij} \), associated with constraints (16) and variables \( \theta_j \), associated with constraints (17). From duality theory we know that the reduced cost of route \( k \in K \) associated with depot \( j \in J \), \( \bar{C}_{jk} \), is given by:

\[
\bar{C}_{jk} = C_{jk} - \sum_{i \in I} a_{jk} (\pi_i - \mu_{ij} - d \theta_j).
\]

Cost \( C_{jk} \) is the sum of the costs of the arcs belonging to route \( k \in K \). Consequently, if we are able to transfer the information of the variables associated with the nodes of the network to its arcs, it will be possible to compute the reduced cost of a route by adding up the reduced costs of the arcs belonging to this route. It will thus become possible to identify routes with a negative
reduced cost, by solving a constrained shortest path problem for each depot $j \in J$ of a network where the costs of the arcs are the reduced costs.

In order to construct a network for each $j \in J$ of the sub-problem we define a sub-graph $G^j \subset G$. Let $N^j = \{0, n+1\} \cup I$ be the set of nodes of $G^j$. Node 0 represents depot $j \in J$ and is the initial node of the network, while $n+1$ also represents depot $j$, being however the final node of the network (we assume $n=|I|$). Let $A^j = (\{0\} \times f \cup \times f \{n+1\})$ be the corresponding set of arcs. The cost of a route $k = (i_0, i_1, \ldots, i_n, i_{n+1})$ is defined as $\sum_{t=0}^{n} c_{i_t, i_{t+1}}$. $\overline{C}_{jk}$ is thus given by:

$$\overline{C}_{jk} = \sum_{t=0}^{n} c_{i_t, i_{t+1}} - \sum_{t=1}^{n} (\pi_{i_t} - \mu_{i_t} - d_{i_t} \theta_j) =$$

$$= c_{i_0, i_1} - \pi_{i_1} + \mu_{i_1,j} + d_{i_1} \theta_j + \sum_{t=2}^{n} (c_{i_{t-1}, i_t} - \pi_{i_t} + \mu_{i_t,j} + d_{i_t} \theta_j) + c_{i_n, i_{n+1}}.$$

We can therefore define the reduced cost of an arc $(l, m) \in A^j$ as:

$$\overline{c}_{lm} = \begin{cases} 
    c_{lm}, & \text{if } m = n + 1; \\
    c_{lm} - \pi_{m} + \mu_{mj} + d_{m} \theta_j, & \text{otherwise.}
\end{cases}$$

The information related to dual variables was thus transferred from nodes to arcs. In order to determine whether there exist routes with negative reduced costs, for each depot $j \in J$ a shortest path problem with distance and (vehicle) capacity constraints must be solved. If the reduced cost of at least one of the constrained shortest path problems for the depots $j \in J$ is negative, a column corresponding to each depot $j \in J$ associated with a route having negative reduced cost is added to the restricted master problem and this problem is solved again. The procedure ends when the reduced costs are positive for all constrained shortest path problems associated with depots $j \in J$.

The pricing sub-problems were solved using the algorithm of Righini and Salani (2006). The algorithm proposed by Righini and Salani for the solution of the resource constrained elementary shortest path problem (RCESPP) is an improvement over the algorithm proposed by Feillet et al. (2004). This algorithm is based on dynamic programming and uses two ideas to improve its performance: bidirectional dynamic programming and bounding. Bidirectional dynamic programming is a technique used to speed up Dijkstra’s (1959) algorithm for the shortest path problem. Bounding is used to limit the number of states generated at every iteration of the dynamic programming algorithm.

4. BRANCHING RULES

The solution obtained with the column generation procedure may contain fractional values. In order to find integer optimal solutions a branch-and-price algorithm is used. Branch-and-Price is the name given to the generalized version of branch-and-bound in which column generation is used to solve an appropriate linear program at each node of the tree. Depending on the branching strategy adopted, the sub-problem structure may change; thus the branching strategy has to consider the effects on the solution of the pricing problem.

The branching rule used in this work is the same proposed by Berger et al. (2007). If in the root node there are variables $y_j$ and $x_{jk}$ with fractional values, branching is first applied to the location variables $y_j$. Only when all $y_j$ variables are integer, branching is applied to the routing variables $x_{jk}$.

The following branching rule is applied for location variable $y_j$: $y_j = 1$ in one child node and $y_j = 0$ in the other child node. This branching rule is incorporated into RMP by fixing $y_j$ to 1
and then solving the pricing problem for the depot \( j \) as before in one node, and by fixing \( y_j \) to 0 and \( x_{jk} \) to 0 in the other child node. In each node with a fractional variable \( y_j \), the \( y_j \) closest to 0.5 is selected for branching.

For the fractional \( x_{jk} \) variables we used a branching rule first proposed by Ryan and Foster (1981) and adapted to the Vehicle Routing Problem with Time Windows (VRPTW) by Desrochers and Soumis (1989). This rule is based on the observation that given a fractional \( x_{jk} \) variable in the solution of RMP, there are at least two columns whose corresponding variables have fractional values: both have coefficient \( a_{ijk} = 1 \) in a common line \( i = r \), and, in another line \( i = s \), one column has coefficient \( a_{ijk} = 1 \) and the other coefficient \( a_{ijk} = 0 \). Using this branching rule, in one branch the two lines \( r \) and \( s \) will be forced to be covered by the same column, and in the other branch they will be forced to be covered by different columns.

To apply this rule to LRP, we define \( R_{j(t_1,t_2)} \) to be the set of routes associated with facility \( j \) in which customer \( t_2 \) immediately follows customer \( t_1 \). The pair of customers is chosen such that in the LP solution of the current node we have \( 0 < \sum_{j \in J} \sum_{k \in R(t_1,t_2)} x_{jk} < 1 \).

Based on this condition two branches are created:

\[ (0): \quad \sum_{j \in J} \sum_{k \in R(t_1,t_2)} x_{jk} = 0 ; \]
\[ (1): \quad \sum_{j \in J} \sum_{k \in R(t_1,t_2)} x_{jk} = 1 . \]

Branch (0) forces clients (lines) \( t_1 \) and \( t_2 \) to be covered by different columns and branch (1) forces the same clients to be covered by the same column.

In RMP, for branch (0) every column in which the clients \( t_1 \) and \( t_2 \) appear consecutively have their correspondents variables fixed to 0, and for branch (1) the columns that cover either \( t_1 \) or \( t_2 \) have their correspondents variables fixed to 0. In the pricing sub-problem, for branch (0) the arcs between \( t_1 \) and \( t_2 \) are deleted, forcing \( t_1 \) and \( t_2 \) to appear in different columns; for branch (1) clients \( t_1 \) and \( t_2 \) are combined in a single node, forcing \( t_1 \) and \( t_2 \) to appear consecutively in the same column.

The two clients \( t_1 \) and \( t_2 \) with the value of \( \sum_{j \in J} \sum_{k \in R(t_1,t_2)} x_{jk} \) closest to 0.5 are chosen for branching.

5. COMPUTATIONAL RESULTS

In order to test the algorithm, instances with 3 factories, 8 potential depots, and 50, 70 and 100 customers were generated. The customers’ locations were adapted from Solomon’s VRPTW instances (Solomon, 1987). His groups C1, R1 and RC1 were used. In C1 customers are clustered; in R1 customers are located in random positions; and RC1 is a mix of groups C1 and R1. For the depot potential locations, we generated coordinates \((x,y)\) based on a uniform distribution in the range \([1,100]\).

The cost of distribution from the factories to the depots, the fixed costs of the depots and the clients’ demands were generated form uniform distributions within the ranges \([1,10]\), \([400,500]\) and \([10,80]\), respectively.

For each instance of 3 factories, 8 potential depots, and 50, 70 and 100 customers, we tested the algorithm with the lengths of the routes limited to 40, 60 and 80 km, and with the vehicle capacities limited to 80, 90 and 100 units.

The branch-and-price procedure described above was implemented using C++ and the solver CPLEX 9.0. CPLEX was used to solve the LP relaxation of the restricted master problem; the connection with the C++ code was made using ILOG’s CONCERT technology. The algorithm of Righini and Salani (2006) was implemented in Visual C++. Tables 1, 2 and 3 show...
the results obtained for groups R1, RC1 and C1, respectively. The columns of these tables have the following meaning:

**NC**: Number of customers;  
**VC**: Vehicle capacity;  
**MD**: Maximum length of the routes;  
**TT**: Total CPU time (in seconds);  
**Nodes**: Number of branch-and-cut nodes evaluated;  
**Cols**: Number of columns generated;  
**RG (%)**: Gap between the LP relaxation at the root node (lower bound) and the upper bound obtained applying a conventional branch-and-bound algorithm to solve the integer program defined using the set of columns generated at the root node;  
**G (%)**: Integrality Gap after four hours of CPU time, expressed as a percentage; no value is reported if the Gap is zero;  
**ND**: Number of depots located;  
**NR**: Number of routes generated;  
**CR**: Maximum number of customers in a route;  
**PP**: Percentage of the time spent in the solution of the pricing sub-problem.

Table 1. Results for the R1 instances.

<table>
<thead>
<tr>
<th>NC</th>
<th>VC</th>
<th>MD</th>
<th>TT</th>
<th>Nodes</th>
<th>Cols</th>
<th>RG (%)</th>
<th>G(%)</th>
<th>ND</th>
<th>NR</th>
<th>CR</th>
<th>PP (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>50</td>
<td>80</td>
<td>40</td>
<td>12.26</td>
<td>5</td>
<td>2,422</td>
<td>0.098</td>
<td>-</td>
<td>1</td>
<td>31</td>
<td>3</td>
<td>88</td>
</tr>
<tr>
<td>50</td>
<td>90</td>
<td>40</td>
<td>45.48</td>
<td>21</td>
<td>3,355</td>
<td>0.036</td>
<td>-</td>
<td>1</td>
<td>27</td>
<td>3</td>
<td>89</td>
</tr>
<tr>
<td>50</td>
<td>100</td>
<td>40</td>
<td>65.48</td>
<td>19</td>
<td>4,012</td>
<td>0.306</td>
<td>-</td>
<td>1</td>
<td>23</td>
<td>3</td>
<td>91</td>
</tr>
<tr>
<td>50</td>
<td>80</td>
<td>60</td>
<td>29.43</td>
<td>13</td>
<td>3,180</td>
<td>0.189</td>
<td>-</td>
<td>1</td>
<td>31</td>
<td>3</td>
<td>89</td>
</tr>
<tr>
<td>50</td>
<td>90</td>
<td>60</td>
<td>57.62</td>
<td>13</td>
<td>4,567</td>
<td>0.138</td>
<td>-</td>
<td>1</td>
<td>30</td>
<td>3</td>
<td>91</td>
</tr>
<tr>
<td>50</td>
<td>100</td>
<td>60</td>
<td>109.82</td>
<td>15</td>
<td>7,094</td>
<td>0.184</td>
<td>-</td>
<td>1</td>
<td>24</td>
<td>3</td>
<td>93</td>
</tr>
<tr>
<td>50</td>
<td>80</td>
<td>80</td>
<td>182.71</td>
<td>17</td>
<td>12,030</td>
<td>0.160</td>
<td>-</td>
<td>1</td>
<td>29</td>
<td>3</td>
<td>91</td>
</tr>
<tr>
<td>50</td>
<td>90</td>
<td>80</td>
<td>2,630.78</td>
<td>93</td>
<td>60,595</td>
<td>0.160</td>
<td>-</td>
<td>1</td>
<td>24</td>
<td>4</td>
<td>93</td>
</tr>
<tr>
<td>70</td>
<td>80</td>
<td>40</td>
<td>240.98</td>
<td>27</td>
<td>4,162</td>
<td>0.040</td>
<td>-</td>
<td>1</td>
<td>45</td>
<td>3</td>
<td>91</td>
</tr>
<tr>
<td>70</td>
<td>90</td>
<td>40</td>
<td>262.70</td>
<td>17</td>
<td>4,823</td>
<td>0.146</td>
<td>-</td>
<td>1</td>
<td>38</td>
<td>3</td>
<td>93</td>
</tr>
<tr>
<td>70</td>
<td>100</td>
<td>40</td>
<td>66.56</td>
<td>1</td>
<td>5,204</td>
<td>-</td>
<td>-</td>
<td>1</td>
<td>33</td>
<td>3</td>
<td>94</td>
</tr>
<tr>
<td>70</td>
<td>80</td>
<td>60</td>
<td>14,400.00</td>
<td>263</td>
<td>71,712</td>
<td>0.344</td>
<td>0.053</td>
<td>1</td>
<td>37</td>
<td>4</td>
<td>92</td>
</tr>
<tr>
<td>70</td>
<td>90</td>
<td>60</td>
<td>5,365.94</td>
<td>61</td>
<td>40,515</td>
<td>0.059</td>
<td>-</td>
<td>1</td>
<td>34</td>
<td>4</td>
<td>94</td>
</tr>
<tr>
<td>70</td>
<td>100</td>
<td>60</td>
<td>14,400.00</td>
<td>61</td>
<td>90,347</td>
<td>0.208</td>
<td>0.175</td>
<td>1</td>
<td>30</td>
<td>4</td>
<td>94</td>
</tr>
<tr>
<td>70</td>
<td>80</td>
<td>80</td>
<td>1,599.61</td>
<td>39</td>
<td>16,982</td>
<td>0.077</td>
<td>-</td>
<td>2</td>
<td>43</td>
<td>3</td>
<td>93</td>
</tr>
<tr>
<td>70</td>
<td>90</td>
<td>80</td>
<td>14,400.00</td>
<td>103</td>
<td>81,740</td>
<td>0.190</td>
<td>0.116</td>
<td>1</td>
<td>39</td>
<td>3</td>
<td>94</td>
</tr>
<tr>
<td>100</td>
<td>80</td>
<td>40</td>
<td>90.32</td>
<td>1</td>
<td>12,431</td>
<td>-</td>
<td>-</td>
<td>1</td>
<td>54</td>
<td>3</td>
<td>96</td>
</tr>
<tr>
<td>100</td>
<td>90</td>
<td>40</td>
<td>14,400.00</td>
<td>161</td>
<td>113,650</td>
<td>0.199</td>
<td>0.109</td>
<td>1</td>
<td>48</td>
<td>3</td>
<td>97</td>
</tr>
<tr>
<td>100</td>
<td>80</td>
<td>60</td>
<td>4,458.31</td>
<td>15</td>
<td>61,901</td>
<td>1.117</td>
<td>-</td>
<td>1</td>
<td>57</td>
<td>4</td>
<td>91</td>
</tr>
<tr>
<td>100</td>
<td>90</td>
<td>60</td>
<td>14,400.00</td>
<td>15</td>
<td>114,867</td>
<td>0.120</td>
<td>0.076</td>
<td>1</td>
<td>54</td>
<td>4</td>
<td>93</td>
</tr>
<tr>
<td>100</td>
<td>80</td>
<td>80</td>
<td>13,346.00</td>
<td>37</td>
<td>180,951</td>
<td>0.119</td>
<td>-</td>
<td>2</td>
<td>60</td>
<td>3</td>
<td>94</td>
</tr>
<tr>
<td>100</td>
<td>90</td>
<td>80</td>
<td>14,400.00</td>
<td>13</td>
<td>229,011</td>
<td>0.475</td>
<td>0.025</td>
<td>2</td>
<td>56</td>
<td>4</td>
<td>95</td>
</tr>
</tbody>
</table>
### Table 2. Results for the RC1 instances.

<table>
<thead>
<tr>
<th>NC</th>
<th>VC</th>
<th>MD</th>
<th>TT</th>
<th>Nodes</th>
<th>Cols</th>
<th>RG (%)</th>
<th>G (%)</th>
<th>ND</th>
<th>NR</th>
<th>CR</th>
<th>PP (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>50</td>
<td>80</td>
<td>40</td>
<td>32.43</td>
<td>9</td>
<td>5,961</td>
<td>0.034</td>
<td>-</td>
<td>1</td>
<td>30</td>
<td>3</td>
<td>76</td>
</tr>
<tr>
<td>50</td>
<td>90</td>
<td>40</td>
<td>2,049.25</td>
<td>463</td>
<td>42,535</td>
<td>0.284</td>
<td>-</td>
<td>1</td>
<td>25</td>
<td>3</td>
<td>85</td>
</tr>
<tr>
<td>50</td>
<td>100</td>
<td>40</td>
<td>14,400.00</td>
<td>1,371</td>
<td>181,145</td>
<td>0.435</td>
<td>0.105</td>
<td>1</td>
<td>22</td>
<td>3</td>
<td>90</td>
</tr>
<tr>
<td>50</td>
<td>80</td>
<td>60</td>
<td>1,666.45</td>
<td>471</td>
<td>26,721</td>
<td>0.282</td>
<td>-</td>
<td>1</td>
<td>28</td>
<td>4</td>
<td>86</td>
</tr>
<tr>
<td>50</td>
<td>90</td>
<td>60</td>
<td>14,400.00</td>
<td>1,267</td>
<td>180,765</td>
<td>0.901</td>
<td>0.624</td>
<td>1</td>
<td>26</td>
<td>4</td>
<td>89</td>
</tr>
<tr>
<td>50</td>
<td>100</td>
<td>60</td>
<td>4,921.33</td>
<td>253</td>
<td>105,516</td>
<td>0.252</td>
<td>-</td>
<td>1</td>
<td>21</td>
<td>4</td>
<td>91</td>
</tr>
<tr>
<td>50</td>
<td>80</td>
<td>80</td>
<td>5,146.80</td>
<td>399</td>
<td>64,933</td>
<td>0.496</td>
<td>-</td>
<td>1</td>
<td>29</td>
<td>3</td>
<td>82</td>
</tr>
<tr>
<td>50</td>
<td>90</td>
<td>80</td>
<td>14,400.00</td>
<td>595</td>
<td>172,325</td>
<td>0.840</td>
<td>0.678</td>
<td>1</td>
<td>26</td>
<td>3</td>
<td>87</td>
</tr>
<tr>
<td>70</td>
<td>80</td>
<td>40</td>
<td>374.48</td>
<td>51</td>
<td>5,641</td>
<td>0.116</td>
<td>-</td>
<td>1</td>
<td>46</td>
<td>3</td>
<td>88</td>
</tr>
<tr>
<td>70</td>
<td>90</td>
<td>40</td>
<td>14,400.00</td>
<td>1,499</td>
<td>52,581</td>
<td>0.387</td>
<td>0.055</td>
<td>1</td>
<td>41</td>
<td>3</td>
<td>90</td>
</tr>
<tr>
<td>70</td>
<td>100</td>
<td>40</td>
<td>744.76</td>
<td>417</td>
<td>11,470</td>
<td>0.075</td>
<td>-</td>
<td>1</td>
<td>34</td>
<td>3</td>
<td>92</td>
</tr>
<tr>
<td>70</td>
<td>80</td>
<td>60</td>
<td>81.53</td>
<td>3</td>
<td>6,425</td>
<td>0.131</td>
<td>-</td>
<td>1</td>
<td>40</td>
<td>4</td>
<td>89</td>
</tr>
<tr>
<td>70</td>
<td>90</td>
<td>60</td>
<td>14,400.00</td>
<td>555</td>
<td>54,402</td>
<td>0.487</td>
<td>0.409</td>
<td>1</td>
<td>37</td>
<td>4</td>
<td>91</td>
</tr>
<tr>
<td>70</td>
<td>100</td>
<td>60</td>
<td>14,400.00</td>
<td>275</td>
<td>57,319</td>
<td>0.460</td>
<td>0.356</td>
<td>1</td>
<td>35</td>
<td>4</td>
<td>93</td>
</tr>
<tr>
<td>70</td>
<td>80</td>
<td>80</td>
<td>1,599.61</td>
<td>39</td>
<td>16,982</td>
<td>0.077</td>
<td>-</td>
<td>2</td>
<td>40</td>
<td>3</td>
<td>89</td>
</tr>
<tr>
<td>70</td>
<td>90</td>
<td>80</td>
<td>14,400.00</td>
<td>199</td>
<td>66,774</td>
<td>1.112</td>
<td>0.796</td>
<td>2</td>
<td>36</td>
<td>3</td>
<td>91</td>
</tr>
<tr>
<td>100</td>
<td>80</td>
<td>40</td>
<td>467.89</td>
<td>15</td>
<td>7,730</td>
<td>0.140</td>
<td>-</td>
<td>1</td>
<td>52</td>
<td>4</td>
<td>85</td>
</tr>
<tr>
<td>100</td>
<td>90</td>
<td>40</td>
<td>549.60</td>
<td>31</td>
<td>11,015</td>
<td>0.074</td>
<td>-</td>
<td>1</td>
<td>44</td>
<td>4</td>
<td>95</td>
</tr>
<tr>
<td>100</td>
<td>80</td>
<td>60</td>
<td>1,261.22</td>
<td>17</td>
<td>17,286</td>
<td>0.916</td>
<td>-</td>
<td>1</td>
<td>55</td>
<td>3</td>
<td>87</td>
</tr>
<tr>
<td>100</td>
<td>90</td>
<td>60</td>
<td>14,400.00</td>
<td>89</td>
<td>64,350</td>
<td>0.555</td>
<td>0.075</td>
<td>1</td>
<td>51</td>
<td>3</td>
<td>93</td>
</tr>
<tr>
<td>100</td>
<td>80</td>
<td>80</td>
<td>14,400.00</td>
<td>237</td>
<td>59,270</td>
<td>0.610</td>
<td>0.308</td>
<td>2</td>
<td>48</td>
<td>3</td>
<td>90</td>
</tr>
<tr>
<td>100</td>
<td>90</td>
<td>80</td>
<td>14,400.00</td>
<td>87</td>
<td>75,313</td>
<td>0.884</td>
<td>0.038</td>
<td>2</td>
<td>44</td>
<td>4</td>
<td>93</td>
</tr>
</tbody>
</table>

### Table 3. Results for the C1 instances.

<table>
<thead>
<tr>
<th>NC</th>
<th>VC</th>
<th>MD</th>
<th>TT</th>
<th>Nodes</th>
<th>Cols</th>
<th>RG (%)</th>
<th>G (%)</th>
<th>ND</th>
<th>NR</th>
<th>CR</th>
<th>PP (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>50</td>
<td>80</td>
<td>40</td>
<td>5,173.92</td>
<td>781</td>
<td>56,752</td>
<td>0.345</td>
<td>-</td>
<td>1</td>
<td>28</td>
<td>3</td>
<td>89</td>
</tr>
<tr>
<td>50</td>
<td>90</td>
<td>40</td>
<td>14,400.00</td>
<td>903</td>
<td>150,558</td>
<td>0.318</td>
<td>0.145</td>
<td>1</td>
<td>26</td>
<td>3</td>
<td>91</td>
</tr>
<tr>
<td>50</td>
<td>100</td>
<td>40</td>
<td>14,400.00</td>
<td>521</td>
<td>131,189</td>
<td>0.414</td>
<td>0.200</td>
<td>1</td>
<td>24</td>
<td>3</td>
<td>92</td>
</tr>
<tr>
<td>50</td>
<td>80</td>
<td>60</td>
<td>708.20</td>
<td>93</td>
<td>28,500</td>
<td>0.620</td>
<td>-</td>
<td>1</td>
<td>32</td>
<td>3</td>
<td>88</td>
</tr>
<tr>
<td>50</td>
<td>90</td>
<td>60</td>
<td>5,063.53</td>
<td>223</td>
<td>126,639</td>
<td>0.136</td>
<td>-</td>
<td>1</td>
<td>26</td>
<td>3</td>
<td>90</td>
</tr>
<tr>
<td>50</td>
<td>100</td>
<td>60</td>
<td>14,400.00</td>
<td>153</td>
<td>183,599</td>
<td>0.698</td>
<td>0.531</td>
<td>1</td>
<td>23</td>
<td>3</td>
<td>92</td>
</tr>
<tr>
<td>50</td>
<td>80</td>
<td>80</td>
<td>191.89</td>
<td>17</td>
<td>18,120</td>
<td>1.450</td>
<td>-</td>
<td>1</td>
<td>29</td>
<td>3</td>
<td>89</td>
</tr>
<tr>
<td>50</td>
<td>90</td>
<td>80</td>
<td>14,400.00</td>
<td>123</td>
<td>136,411</td>
<td>0.155</td>
<td>0.103</td>
<td>1</td>
<td>26</td>
<td>3</td>
<td>91</td>
</tr>
<tr>
<td>70</td>
<td>80</td>
<td>40</td>
<td>683.57</td>
<td>87</td>
<td>14,085</td>
<td>0.152</td>
<td>-</td>
<td>1</td>
<td>43</td>
<td>3</td>
<td>94</td>
</tr>
<tr>
<td>70</td>
<td>90</td>
<td>40</td>
<td>3,062.48</td>
<td>175</td>
<td>30,743</td>
<td>0.160</td>
<td>-</td>
<td>1</td>
<td>38</td>
<td>4</td>
<td>95</td>
</tr>
<tr>
<td>70</td>
<td>100</td>
<td>40</td>
<td>14,400.00</td>
<td>403</td>
<td>105,357</td>
<td>0.219</td>
<td>0.093</td>
<td>1</td>
<td>35</td>
<td>4</td>
<td>95</td>
</tr>
<tr>
<td>70</td>
<td>80</td>
<td>60</td>
<td>223</td>
<td>37,481</td>
<td>0.142</td>
<td>-</td>
<td>1</td>
<td>62</td>
<td>3</td>
<td>88</td>
<td></td>
</tr>
<tr>
<td>70</td>
<td>90</td>
<td>60</td>
<td>227.37</td>
<td>81</td>
<td>81,323</td>
<td>0.588</td>
<td>0.412</td>
<td>1</td>
<td>38</td>
<td>4</td>
<td>92</td>
</tr>
<tr>
<td>70</td>
<td>100</td>
<td>60</td>
<td>14,400.00</td>
<td>27</td>
<td>126,008</td>
<td>0.317</td>
<td>0.313</td>
<td>1</td>
<td>36</td>
<td>4</td>
<td>94</td>
</tr>
<tr>
<td>70</td>
<td>80</td>
<td>80</td>
<td>7,095.61</td>
<td>240</td>
<td>68,986</td>
<td>1.209</td>
<td>-</td>
<td>1</td>
<td>41</td>
<td>3</td>
<td>92</td>
</tr>
<tr>
<td>70</td>
<td>90</td>
<td>80</td>
<td>2,522.67</td>
<td>35</td>
<td>51,306</td>
<td>1.502</td>
<td>-</td>
<td>1</td>
<td>36</td>
<td>4</td>
<td>93</td>
</tr>
<tr>
<td>100</td>
<td>80</td>
<td>40</td>
<td>7,818.38</td>
<td>158</td>
<td>24,520</td>
<td>0.256</td>
<td>-</td>
<td>1</td>
<td>58</td>
<td>3</td>
<td>89</td>
</tr>
<tr>
<td>100</td>
<td>90</td>
<td>40</td>
<td>14,400.00</td>
<td>154</td>
<td>51,056</td>
<td>0.600</td>
<td>0.580</td>
<td>1</td>
<td>55</td>
<td>3</td>
<td>91</td>
</tr>
<tr>
<td>100</td>
<td>80</td>
<td>60</td>
<td>2,042.48</td>
<td>15</td>
<td>37,481</td>
<td>0.142</td>
<td>-</td>
<td>1</td>
<td>62</td>
<td>3</td>
<td>88</td>
</tr>
<tr>
<td>100</td>
<td>90</td>
<td>60</td>
<td>542.89</td>
<td>1</td>
<td>52,978</td>
<td>-</td>
<td>-</td>
<td>1</td>
<td>55</td>
<td>3</td>
<td>97</td>
</tr>
<tr>
<td>100</td>
<td>80</td>
<td>80</td>
<td>14,400.00</td>
<td>13</td>
<td>108,035</td>
<td>0.177</td>
<td>0.125</td>
<td>2</td>
<td>55</td>
<td>4</td>
<td>92</td>
</tr>
<tr>
<td>100</td>
<td>90</td>
<td>80</td>
<td>14,400.00</td>
<td>3</td>
<td>235,478</td>
<td>0.002</td>
<td>0.001</td>
<td>2</td>
<td>53</td>
<td>4</td>
<td>93</td>
</tr>
</tbody>
</table>
The best results were obtained for the R1 instances. We found optimal solutions for all 50 clients’ R1 instances; optimal solutions were also found for some instances with 70 and 100 clients, especially in instances with tighter resource constraints (vehicle capacity and maximum distance). For all R1 instances the integrality gap was very small with minimum and maximum values of 0.03% and 0.18 %, respectively.

For the RC1 instances, the best results were obtained for instances with tight resource constraints. The integrality gaps were larger, ranging from 0.06% to 0.80%. The results were similar for the C1 instances, with optimal solutions obtained in half of them. The C1 instances had integrality gaps ranging from 0.001% to 0.58%.

In general the integrality gaps were small in all instances, what shows the quality of the lower bound obtained with the LP relaxation of the set partitioning formulation at the root node of the branch-and-price tree.

The low number of opened depots is due to high fixed costs for opening the depots, when compared with the routing costs. The maximum number of clients in a route was 4. This is due to tight resources constraints. The vehicle capacity constraints are 80, 90 and 100 units, and the mean of the clients’ demands is between 36 and 37 units. The maximum distances allowed were 40, 60 and 80 Km, and the mean distances between clients and between clients and depots falls in the range [32, 45] Km.

As it can be seen from Tables 1, 2 and 3, the solution of the pricing sub-problems consumes a very high percentage of the algorithm’s solution time. This percentage becomes higher as: (i) the resource constraints become less tight, and (ii) the number of clients becomes higher.

There is a clear correlation between the value of the resource constraints (vehicle capacity and maximum distance) and the difficulty of the problem instance, measured by the overall time and the number of columns generated. When the resource constraints are less tight, the algorithm needs to generate a larger number of columns, and this requires a significant amount of computing time. Therefore, $MaxDem$ and $MaxDist$ can be assumed to be a rough estimate of the difficulty of the instance from the viewpoint of a branch-and-price approach.

6. CONCLUSIONS

In this paper we presented a branch-and-price algorithm for a 3-level location-routing problem. The branch-and-price procedure proved adequate for solving the proposed problem. Optimal solutions were found for instances with 50, 70 and 100 clients.

An earlier implementation of this method by Berger et al. (2006) for a 2-level location-routing problem met with success. Our problem differs from the one formulated and solved in that paper by considering one more level in the logistics chain. This additional level implies changes both in the restricted master problem and in the pricing sub-problems. In our algorithm the pricing sub-problems were solved using the algorithm of Righini and Salani (2006), which has a better performance when compared to the algorithm proposed by Feillet et al., used by Berger et al. in their paper.

Future work will investigate the use of heuristics to find columns with negative reduced costs. The dynamic programming algorithm of Righini and Salani will be used only to guarantee optimal solutions, when the heuristic cannot find columns with negative reduced costs any more. Another suggested improvement would be the use of a better heuristic to generate the initial set of columns at the root node.

REFERENCES