Chapter 2

The Deterministic Location-Routing Problem*

Introduction

In this chapter, we introduce the LRP that we study in this thesis and we present both, an upper and a lower bound.

We assume that we are given a set of customers with associated demands, a set of potential facility locations, each with a fixed operating cost and a capacity, and the traveling costs between any two points. The goal is to determine the number and locations of the facilities to be open and to design one route from each selected location in such a way that each customer belongs to exactly one route, capacity constraints on the facilities are satisfied, and the total costs (fixed plus routing costs) are minimized. As we will see later, this problem is an extension of the capacitated VRP.

The problem is modeled in terms of finding a family of paths in an auxiliary graph that satisfy some side constraints. The solution to a reinforced *linear programming* (LP) relaxation of this model is used in a rounding procedure to obtain a first lower bound and as starting point for a TS heuristic. The TS heuristic consists of a series of intensification and diversification iterations. The diversification phase operates mainly at the location level and selects new subsets of open facilities. The intensification phase is a local search phase that is applied to the current routing subproblem. Due to the capacity constraints feasibility is not easy to ensure in the intensification phase. For this reason we use a strategic oscillation scheme that allows violation of feasibility and considers a modified objective function that includes a penalty term associated with infeasible solutions. The weight given to such a penalty is dynamically updated based on the history of the search as in Díaz and Fernández (2001).

Additionally, we propose a lower bound that generally provides a major improvement with respect to the LP bound of the proposed model that, in general, is very weak. The new lower bound has two terms that are derived from the structure of each of the two main components of the original problem: The first refers mainly to the location costs and the second to the routing costs. The term of the location costs is obtained by solving a *Knapsack Problem* (KP) whereas the term of the routing costs is obtained by solving an *Asymmetric Traveling Salesman Problem* (ATSP).

The quality of the upper and lower bounds has been tested in a series of computational experiments. The results show the good quality of both the new lower bound and the TS heuristic. The required

*The contents of this chapter are partially included in Albareda-Sambola, Díaz, and Fernández (2002)

times are small for problems of this difficulty especially taking into account that most of the test problems could not be optimally solved by CPLEX 6.5 within 48 hours of CPU time.

The chapter is organized as follows. In Section 2.1 we define the problem and we propose a model defined in terms of an auxiliary network. Section 2.2 describes the rounding procedure and the TS heuristic. In Section 2.3 we show how to obtain a new lower bound to the problem. The details of the computational experiments as well as the obtained results are presented in Section 2.4. We end the chapter with some conclusions and final remarks.

2.1 A compact model

Let I denote the set of indices of potential locations for the plants and J the set of indices for the clients. For each $i \in I$, let f_i be the cost for opening the plant at site i and let b_i be the capacity of plant i. For each $j \in J$, d_j denotes the demand of client j, and D stands for the aggregated demand. Also let $C = (c_{ij})$, $i, j \in I \cup J$ be the matrix that contains the travel costs from clients to plants, and between clients. We want to find a set of plants to be opened, and a set of routes to service clients from the open plants such that the opening costs plus the costs of the routes are minimized. The cost of a route is defined to be the sum of the costs of each travel in the route. We assume that *i)* only one route takes place from each open plant; *ii)* the routes start and end at the same plant; *iii)* the vehicles are uncapacitated (or, equivalently, the capacity of an open plant, also represents the capacity of the vehicle for the route associated with the plant); and *iv)* the demand of each client must be satisfied from one single plant; that is, each client is visited by exactly one route.

Let us note that the LRP considered here can be seen as an extension of the classical Capacitated Vehicle Routing Problem (CVRP). On the one hand, multiple depots are considered. There are, indeed, some mentions to the multiple depot CVRP present in the literature but generally they only refer to extensions of results for the single depot case. On the other hand, in the problem we consider there is an additional level of decision, since the set of depots to be used has to be selected.

To formulate the problem in terms of a network problem with additional constraints we define the network $N = (V, A)$. The set of nodes is defined to be $V = \{s, t\} \cup I \cup I' \cup J$, where I' is a copy of set I, and s and t are, respectively, a pseudo source node and a pseudo sink node (see Figure 2.1a)). The set of arcs is

$$\begin{aligned} A = &\{(s,i)|\ i \in I\} \cup \{(i,j)|\ i \in I, j \in J\} \cup \{(j_1,j_2)|\ j_1, j_2 \in J; j_1 \neq j_2\} \cup \\ &\{(j,i')|\ j \in J, i' \in I'\} \cup \{(i',t)|\ i' \in I'\} \cup \{(s,t)\}, \end{aligned}$$

i.e., there are arcs from the source to each plant, from plants to customers, among any pair of customers, from customers to the copies of the plants, from the copies of the plants to the terminal, and also the dummy arc that goes from the source to the terminal.

The cost function $\tilde{c}$ (see Figure2.1b)) defined on the arcs of N is given by:

- $\tilde{c}_{si} = f_i, \forall i \in I$;
- $\tilde{c}_{ij} = \tilde{c}_{ji'} = c_{ij}, \forall i \in I, j \in J$ (i' denotes the copy of plant i);
- $\tilde{c}_{j_1 j_2} = c_{j_1 j_2}, \forall j_1, j_2 \in J, j_1 \neq j_2$;
- $\tilde{c}_{i't} = 0, \forall i' \in I'$;
- $\tilde{c}_{st} = 0$.

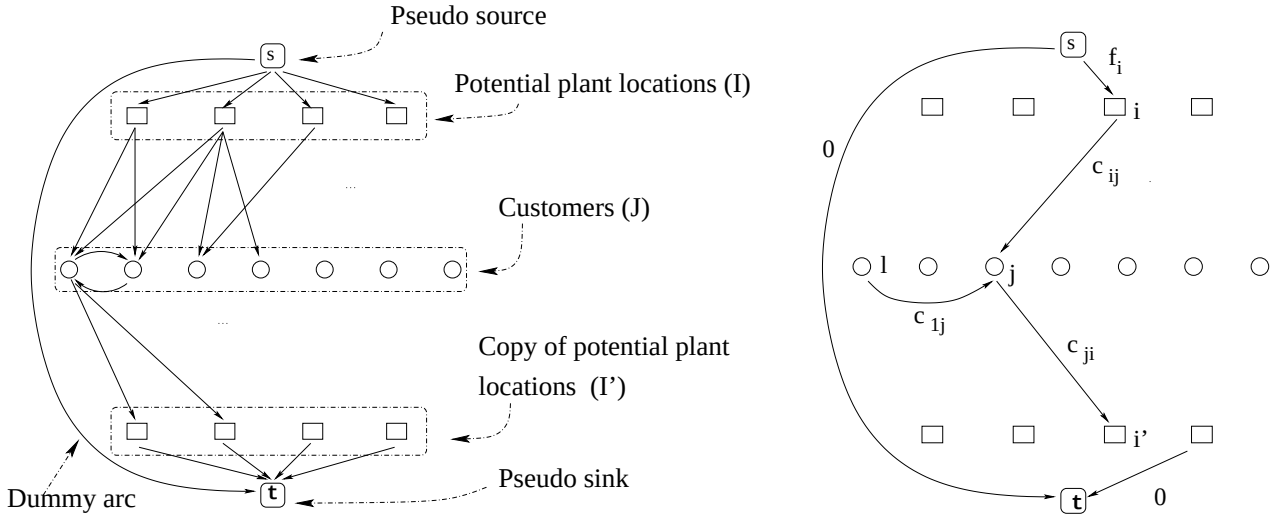


Figure 2.1: Auxiliar network N(V,A)

We next explain how to use the above network to find a set of paths, one associated with each potential plant, that will represent the set of plants to be opened together with the routes to service clients from open plants. Consider any path P from s to t in N. If $P = \{(s,t)\}$, P is called *trivial path* whereas if P does not contain the arc (s,t), it is a *non-trivial* path. Let i_s and i'_t denote the indices of the plants connected with s and t respectively in a non-trivial path P. P can be seen as a route starting at i_s and ending at i'_t. When $i_s = i'_t$, the route starts and ends at the same plant. By definition of the cost function $\tilde{c}$, the cost of P is the sum of the opening cost for plant i_s plus the overall cost of the route. On the other hand, trivial paths are related to not opening a plant and, thus, they have cost 0. The equivalence between solutions to the LRP and paths in the defined network can be further appreciated in Figure 2.2.

Therefore, the combined LRP can be modeled as the problem of finding a set of paths in the network that fulfill some additional constraints. Specifically, for $v \in V$, let

$$A(v)^+ = \{a \in A : a = (v,u)\},$$

$$A(v)^- = \{a \in A : a = (u,v)\}$$

and let

$$A(S) = \{a \in A : a = (u,v), u,v \in S\}, \text{ for } S \subseteq I.$$

We define the decision variables $x_a^k, a \in A, k \in I$ to be 1 if arc a is used in route k and 0 otherwise. Then, the problem can be modeled as:

$$\text{(LR1)} \quad \text{minimize} \quad \sum_{k\in I}\sum_{a\in A} \tilde{c}_a x_a^k \tag{2.1}$$

$$\text{subj. to} \quad \sum_{k\in I}\sum_{a\in A(s)^+} x_a^k = |I| \tag{2.2}$$

$$\sum_{a\in A(v)^+} x_a^k = \sum_{a\in A(v)^-} x_a^k \qquad \forall v \in V\backslash\{s,t\}, \forall k \in I \tag{2.3}$$

$$\sum_{k\in I}\sum_{a\in A(t)^-} x_a^k = |I| \tag{2.4}$$

$$x_{(s,i)}^k = x_{(i',t)}^k \qquad \forall i,k \in I \tag{2.5}$$

$$\sum_{j\in J}\sum_{a\in A(j)^+} d_j x_a^k \leqslant \sum_{i\in I} b_i x_{(s,i)}^k \qquad \forall k \in I \tag{2.6}$$

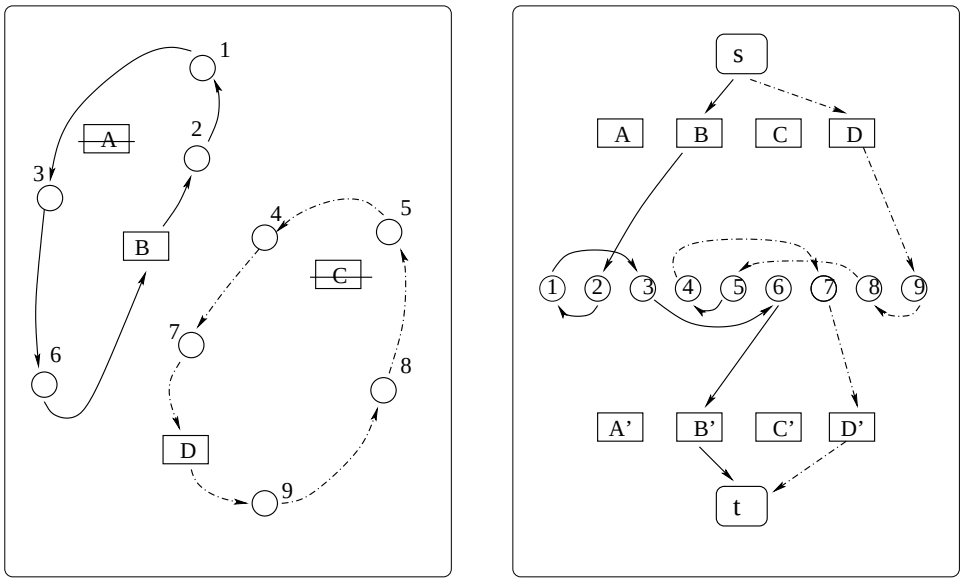
$$\sum_{k\in I}\sum_{a\in A(j)^-} x_a^k = 1 \qquad \forall j \in J \tag{2.7}$$

$$\sum_{k\in I}\sum_{a\in A(S)} x_a^k \leqslant |S| - \left\lceil \frac{\sum_{j\in S} d_j}{v_{max}} \right\rceil \qquad \forall S \subseteq J \tag{2.8}$$

$$x_{(s,i)}^k = 0 \qquad \forall i,k \in I, i \neq k \tag{2.9}$$

$$x_a^k \in \{0,1\} \qquad \forall a \in A, \forall k \in I. \tag{2.10}$$

Constraints (2.2)-(2.4) are the flow conservation equations that guarantee that the solution defines $|I|$ s-t-Paths. Equations (2.5) guarantee that the routes start and end at the same plant while inequalities (2.6) ensure that the capacity of vehicles is not violated and that no client is serviced from a plant that is not open. Equalities (2.7) require that each client is visited by exactly one route. Constraints (2.8) forbid routes that are not connected with any plant. They are a generalization of



a) Solution of the LRP b) Equivalent solution in $N(V, A)$

Figure 2.2: Typical solution in both schemes

the well-known subtour elimination constraints that, additionally, take into account the capacity of the plants. Here v_{max} denotes the maximum capacity of all plants. Constraints (2.8) are a variation of a class of inequalities that were proposed by Laporte and Nobert (1983) in a somewhat different context of pure routing problems. Note that the aggregation over all plants of the subtour elimination constraints associated with a given set of clients is valid, given that feasible sets of routes are node disjoint. Constraints (2.9) are used as a means to associate each path with a specific plant, by only permitting the path to go through one of the plants.
LR1 is a linear integer program with $|I|\ (2\,|I| + 2\,|I| \cdot |J| + |J|\,(|J| - 1) + 1)$ binary variables. However, (2.9) together with constraints (2.2)-(2.5) imply that $x^k_{(i',t)} = 0 \quad \forall i' \neq k; i', k \in I, \quad x^k_{(i,j)} = 0 \quad \forall i \neq k; i, k \in I; j \in J$, and $x^k_{(j,i')} = 0 \quad \forall i' \neq k; i', k \in I; j \in J$. Therefore, in LR1 the actual number of variables is reduced by $2|I|\,(|I| - 1)\,|J|$.
As usual in routing problems, there is a number of subtour elimination constraints (2.8) that is exponential in the number of clients. Additionally, the model has a number of constraints of other types that is bounded by $O(|I| \cdot |J|)$.

Indeed, the LRP that we consider in this paper can also be modeled without stating the problem in terms of an auxiliary network. For instance, the most general three index model of Perl and Daskin (1985) can be adapted to our problem. In this case the set of decision variables necessarily should include binary variables that indicate if client j is visited immediately after client l in route k, which are a subset of set of the decision variables used in $LR1$. Although this could be avoided, typically one would define additional decision variables to know if a facility is open at site k as well as other decision variables to indicate if client i is visited by route k. The resulting model that does not differ essentially from LR1, can also be easily derived from $LR1$ with a slight transformation.

We use a solution to the LP relaxation of LR1 as a starting point of the heuristic approach that is described in the next section. The LP relaxation of LR1 is strengthened by including the following aggregated demand constraint:

$$\sum_{i \in I} b_i x^i_{(s,i)} \geqslant \underline{b} \tag{2.11}$$

where $\underline{b}$ is the solution to the following Subset Sum Problem

$$\underline{b} = \text{ minimize } \sum_{i \in I} b_i y_i \tag{2.12}$$

$$\text{subj. to } \sum_{i \in I} b_i y_i \geqslant D \tag{2.13}$$

$$y_i \in \{0, 1\},\ i \in I. \tag{2.14}$$

Constraint (2.11) guarantees that the overall capacity of the plants that are open is enough as to satisfy the overall demand of the clients. Note that (2.11) is at least as good as the classical aggregated demand constraint where the right hand side is D.

Let RLR1 denote the LP relaxation to model LR1 strengthened with constraint (2.11) and where the condition that no client is serviced from a plant that is not open implied by constraints (2.6) has been expressed in its disaggregated form:

$$x^i_a \leqslant x^i_{(s,i)}, \forall i \in I, \forall a \in A \setminus \{(s,t)\}. \tag{2.15}$$

When solving RLR1, in order to have a linear programming problem of a reasonable size, initially

we relax constraints (2.8). We then apply an iterative LP solver scheme where inequalities (2.8) are added only when they are violated in the current continuous solution.

2.2 Tabu Search Heuristic

The TS that we apply to obtain solutions for LR1 consists of a series of iterations, each of which performs an intensification phase followed by a diversification phase. Initially we apply a rounding procedure to the optimal solution to RLR1 that provides the first binary solution to which the intensification phase is applied. In subsequent iterations, intensification is applied to the result of the diversification phase. All the solutions that we consider in the TS heuristic satisfy all constraints (2.2)-(2.10) excepting, possibly, capacity constraints (2.6). Since solutions that violate some capacity constraint are allowed, the heuristic presents a strategic oscillation behavior.

Rounding Heuristic

The Rounding heuristic is a simple constructive procedure that builds a solution from a solution of the relaxation RLR1, $\bar{x}$. As it is described in Algorithm 2.1, it has three parts. At a first step, the set of open plants O is chosen, as the set of plants that have a positive *non trivial flow* (overall flow along non trivial paths) in $\bar{x}$. A threshold ε is fixed to avoid opening plants that are only marginally used in $\bar{x}$. Once the set O is fixed, the allocation phase is entered. In this phase each customer is taken in turn, and allocated to one plant. Among all the open plants whose flow reaches a given customer (j), the one with higher ratio

$$\delta_i^j = \frac{\sum_{a \in A(j)^-} \bar{x}_a^i}{1 - \bar{x}_{(s,t)}^i} \tag{2.16}$$

is chosen. Note that δ_i^j is the ratio between the flow that arrives to client j in the route of plant i, and the actual non trivial flow of the route. Finally, once all the customers have been allocated, unused open plants are closed, as long as the overall capacity does not go under $\underline{b}$.

The result of the rounding procedure is a set of open plants and a set of routes, each of them associated with one different open plant. Some of the routes may violate the capacity constraints of their associated plants, since capacity constraints (2.6) are not taken into account in the rounding procedure. However, by construction, the obtained solution satisfies all other constraints (2.2)-(2.10). The TS heuristic tries to improve the solutions of the rounding procedure while reducing unfeasibility. To this end i) we add a penalty term to the objective function that weights the violation of capacity constraints, and ii) we explore two simple neighborhoods that modify the assignment of at least one client (this is required to recover feasibility when the solutions violate some capacity constraint). For any solution x, let

$$s(x) = \sum_{i \in I} \max \left\{ \sum_{j \in J} \sum_{a \in A(j)^+} x_a^i - b_i \, x_{(s,i)}^i, 0 \right\} = \sum_{i \in O(x)} \max \left\{ \sum_{j \in J} \sum_{a \in A(j)^+} x_a^i - b_i, 0 \right\} \tag{2.17}$$

denote its overall violation of capacity constraints. The modified objective function that we consider both in the intensification and the diversification phases is:

$$\sum_{i \in I} \sum_{a \in A} \tilde{c}_a^i x_a^i + P \, s(x) = \sum_{i \in O(x)} \sum_{a \in A} \tilde{c}_a^i x_a^i + P \, s(x), \tag{2.18}$$

Algorithm 2.1 Rounding

Initialization: $O = \emptyset, b = 0$ {Set of open plants and their overall capacity}
Solve RLR1 $\longrightarrow \bar{x}$
{Selection of the open plants}
for $(i \in I)$ **do**
  if $(\bar{x}^i_{(s,t)} < 1 - \varepsilon)$ **then**
    $O \leftarrow O \cup \{i\}$
    $b \leftarrow b + b_i$
  end if
end for
{Allocation of customers to plants}
for $(j \in J)$ **do**
  Compute $\delta^j_i \ \forall i \in O$
  Take $i(j) \in \arg\min\{\delta^j_i, i \in O\}$
  Apply nearest insert to place customer j on route $i(j)$.
end for
{Close unnecessary plants}
for $(i \in O)$ **do**
  if $(\ \{j | i = i(j)\} = \emptyset$ **and** $c - b_i \geqslant \underline{b})$ **then**
    $O \leftarrow O \setminus \{i\}$
    $b \leftarrow b - b_i$
  end if
end for

where P is the weight factor and $O(x)$ denotes the set of indices of open plants for solution x. Note that for feasible solutions the two objectives (2.1) and (2.18) give the same value. The penalty weight P is dynamically updated taking into account the history of the search. The strategy that we use for updating the penalty weight has been previously used in Díaz and Fernández (2001) for the GAP. In particular, P is updated according to the expression $P := P\,\alpha^{\beta}$, where α is dynamically updated using the information provided by the medium-term memory, and β is also dynamically updated but using the information of the short-term memory. In our case, $\beta = \frac{\eta}{\theta_1 - 1} - 1$, where η is the number of infeasible solutions obtained in the last θ_1 iterations and α is initially set to the value 2 and updated every θ_2 iterations according to $\alpha := min\,\{\alpha + 0.01, 3\}$. Note that for a given value of α the value of P varies in $\left[\alpha^{-1} \times P, \alpha^{\frac{1}{\theta_1 - 1}} \times P\right]$. Thus, the bigger is the value of α, the larger is the range of variation for P. Additionally, since the values of α are always greater than or equal to one, P will only increase when all the solutions found in the last θ_1 iterations are infeasible.

The Intensification Phase

The intensification phase is basically oriented to look for good-quality feasible solutions. Since the initial solutions may violate some capacity constraint (2.6), one of the goals in this phase is reducing unfeasibility when it occurs.

As mentioned previously, in order to recover feasibility it is necessary to modify the assignment of at least one client. This is achieved by exploring two simple neighborhoods. The first one is the reassignment neighborhood (denoted $N1$), and the second one is the interchange neighborhood (denoted $N2$). For a given solution x, $N1(x)$ contains all the solutions that only differ from x in the assignment of one single client, which is visited by a different route (see Figure 2.3).

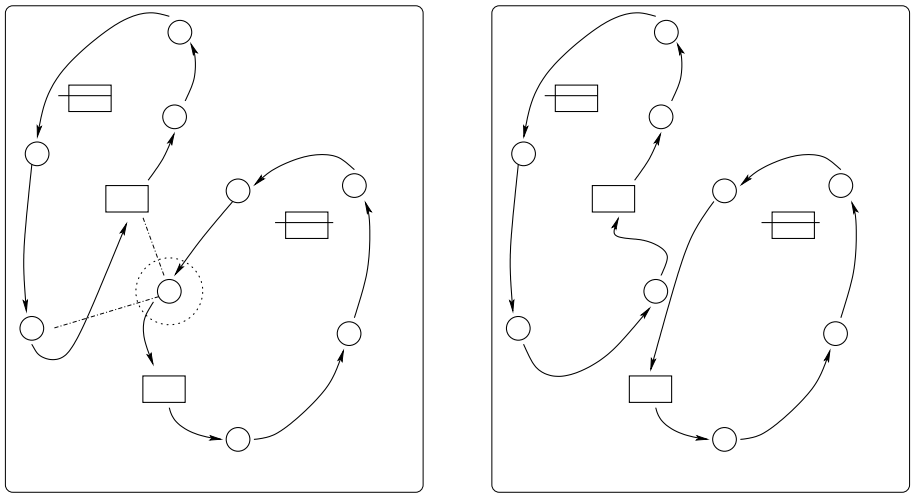


Figure 2.3: Neighborhood $N1(x)$. Reassignment of one customer

Similarly, $N2(x)$ contains all solutions that only differ from x in the assignment of exactly two clients that interchange the routes that visit them (see Figure 2.4). Note that, for a given solution x,

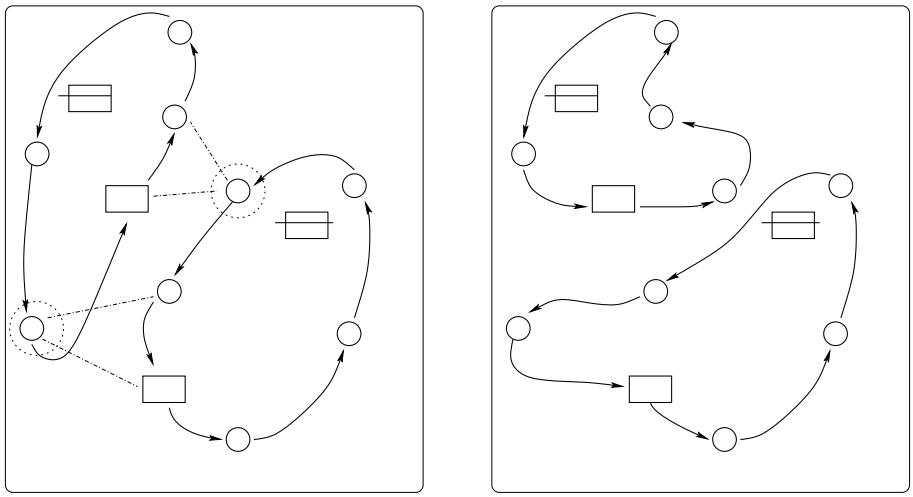


Figure 2.4: Neighborhood $N2(x)$. Interchange the assignment of two customers

moves in $N2(x)$ have a fixed number of clients in each route. On the other hand, moves in $N1(x)$ are more restrictive in terms of feasibility. For this reason, we chose to explore $N1(x) \cup N2(x)$, which is a more flexible option. For a given solution x we evaluate all non-tabu moves in $N1(x) \cup N2(x)$ and select the one with a better value with respect to (2.18). When a client j is reassigned from plant $i1$ to plant $i2$ the reverse move is tabu active (and thus, forbidden) for a fixed number of iterations. We apply the aspiration criterion to bypass tabu restrictions if they result in solutions that improve the incumbent. As usual, the incumbent solution is the best feasible solution found so far. At the earlier stages of the search when no feasible solution has been found so far, the incumbent is taken to be the *least infeasible* solution (the one with smallest value $s(.)$) known so far.

Additionally, at some steps of the intensification phase we consider swap interchanges within the current routes. That is, we consider each route independently and we perform a sequence of

interchanges of two arcs within the route until we obtain a 2-opt route. In what follows this will be referred to as exploring $N_{2-opt}(x)$. Algorithm 2.2 outlines the intensification phase. In the algorithm, a move is considered acceptable if either it is non-tabu, or it satisfies the aspiration criterion. The algorithm terminates when i) the maximum number of iterations has been reached, or ii) the maximum number of consecutive iterations without finding an acceptable move has been reached.

Algorithm 2.2 Intensification($\hat{x}$)

Initialization: $StopCriterion \leftarrow$ **false**
repeat
  Explore $N_{2-opt}(\hat{x})$ and update $\hat{x}$
  Select the best *acceptable move* in $N1(\hat{x}) \cup N2(\hat{x})$ according to 2.18
  Update $\hat{x}$ and $StopCriterion$
until $StopCriterion$

The Diversification Phase

The neighborhoods considered during the Intensification Phase may change the assignments of clients to open plants but, in all cases, the set of open plants remains fixed throughout that phase. However, for finding good quality solutions it may be necessary to consider different sets of open plants, in order to explore larger areas of the solutions space. The Diversification Phase performs moves that affect the set of open plants, so new solutions associated with different sets of open plants can be generated by the Intensification Phase. The three neighborhoods that are explored in this phase are the following:

$N3$ **Close plant:**
For a given solution x, $N3(x)$ (see Figure 2.5) considers solutions y where the set of open plants differs from $O(x)$ exactly in one plant that is open in x and is closed in y. That is, $O(y) = O(x) \setminus \{i_1\}$ for some $i_1 \in O(x)$. Plants that are not tabu active are ordered in a list by increasing values of the ratio $used\_cap/b_i$. The plant that is closed is the first one from the list that can be closed without violating the aggregated demand constraint. If such a plant does not exist, no move is performed. When a plant is closed its assigned customers are reassigned to the cheapest route (w.r.t. the original objective function) with the nearest-insert criterion. Customers are reassigned in the same order they had in the route of the closed plant.

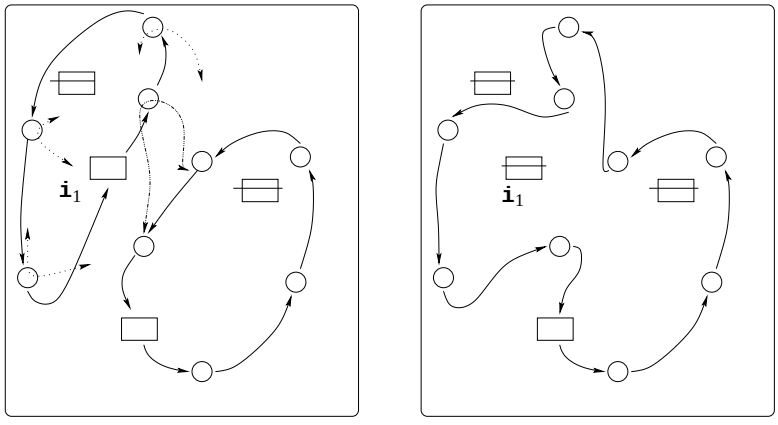


Figure 2.5: Neighborhood $N3(x)$. Close a plant and reassign its customers

$N4$ Switch plants:
For a given solution x, $N4(x)$ (see Figure 2.6) considers solutions y where the set of open plants differs from $O(x)$ exactly in one plant that is open in x and is closed in y, plus another plant that is open in y and is closed in x. That is, $O(y) = O(x) \setminus \{i_1\} \cup \{i_2\}$ for some $i_1 \in O(x), i2 \notin O(x)$. We consider all non-tabu pairs (i_1, i_2), $i_1 \in O(x), i2 \notin O(x)$ such that: *a)* $O(x) \setminus \{i_1\} \cup \{i_2\}$ satisfies the aggregated demand constraint (2.11), and *b)* the total demand of clients assigned to i_i does not exceed the capacity of plant i_2.

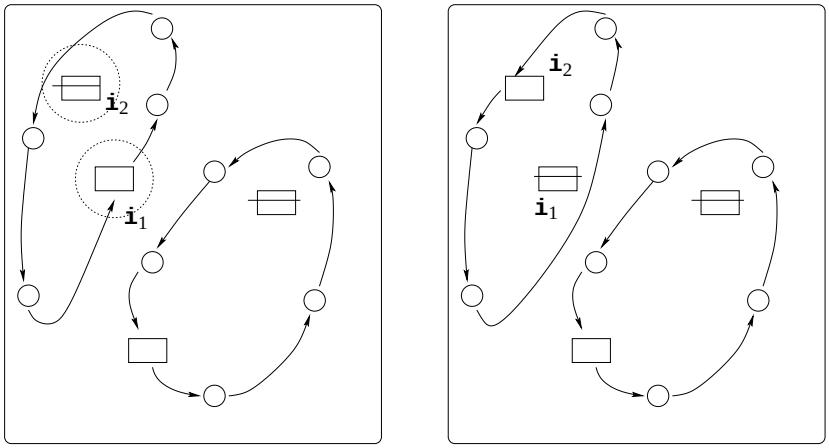


Figure 2.6: Neighborhood $N4(x)$. Switch open plant i_1 and closed plant i_2 by closing i_1 and opening i_2. Assign to i_2 the route of i_1.

Condition *a)* does not guarantee that the selected sets of plants are feasible. The reason is that a set of plants that satisfies (2.11) may not have any feasible assignment of clients. However, we use condition *a)* as a surrogate for the feasibility of the sets of open plants because the decision problem to know if a given set of plants has a feasible assignment is itself NP-complete. We select the admissible move with the best value with respect to the modified objective function (2.18). All clients assigned to i_1 in solution x, are reassigned to i_2 in solution y. The relative order of clients on the route of i_2 is the same as in the route of i_1, although the starting (and the ending) client of the route has to be decided. This is done by applying the nearest-insertion criterion to place plant i_2 on the route.

$N5$ **Open a plant:**
For a given solution x, $N5(x)$ (see Figure 2.7) considers solutions y where the set of open plants differs from $O(x)$ exactly in one plant that is closed in x and is open in y. That is, $O(y) = O(x) \cup \{i_1\}$ for some $i_1 \notin O(x)$. The plant that is opened is the one with the smallest value $\frac{f_i}{b_i}$ among the non-tabu plants.

When a plant i_1 is opened, some customers from other routes are assigned to i_1 as follows: Customers j are considered in turn by increasing values of $c_{i_1 j}$. While the capacity of plant i_1 is not violated clients are reassigned to i_1, using the nearest-insert criterion to place clients on the route of i_1.

After exploring the above three neighborhoods, N_{2-opt} is explored to improve the routes if a move in any of $N3$, $N4$, or $N5$ has been performed. Also, if a plant has been opened or closed it remains tabu active (cannot be closed if it has been opened, or cannot be opened if it has been closed) for a fixed number of Diversification Phases. In the case of $N4$, the aspiration criterion is used to allow tabu moves that improve the incumbent solution. The order in which the neighborhoods are explored

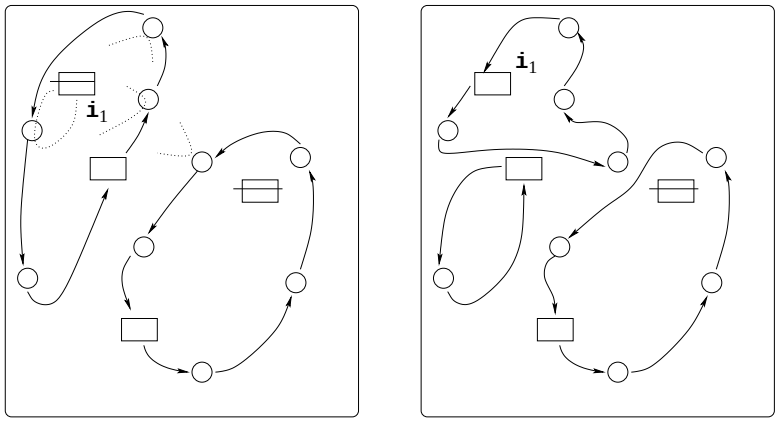


Figure 2.7: Neighborhood $N5(x)$. Open a plant, and assign some customers to it.

is as follows. $N3$ is first explored. If no admissible move in $N3$ is found, $N4$ is explored. $N5$ is only explored when no admissible move has been found neither in $N3$ nor in $N4$. If no feasible solutions are known for the current set of open plants at the beginning of the Diversification Phase, $N3$ is not explored and the Diversification Phase starts by exploring $N4$. The algorithm terminates when no acceptable moves have been found in any of the three neighborhoods. Also, a limit of iterations is considered. When the algorithm terminates because of that limit, an extra intensification phase is applied.

The structure of the TS we propose is presented in Algorithm 2.3.

Algorithm 2.3 Tabu Search

Initialization: $StopCriterion \leftarrow$ **false**
$Move \leftarrow$ **false** {**true** if an acceptable move has been found}
$Feasible \leftarrow$ **false** {**true** if a feasible solution is known for the last set of plants}
Rounding $\longrightarrow \hat{x}$
repeat
  Intensification($\hat{x}$)
  Update $Feasible$
  {Diversification}
  if ($Feasible$) **then**
    Explore $N3(\hat{x})$
    Update $\hat{s}$ and $Move$.
  end if
  if (**not** $Move$) **then**
    Explore $N4(\hat{x})$
    Update $\hat{s}$ and $Move$.
  end if
  if (**not** $Move$) **then**
    Explore $N5(\hat{x})$
    Update $\hat{x}$ and $Move$.
  end if
  $StopCriterion \leftarrow$ (**not** $Move$)
until ($StopCriterion$)

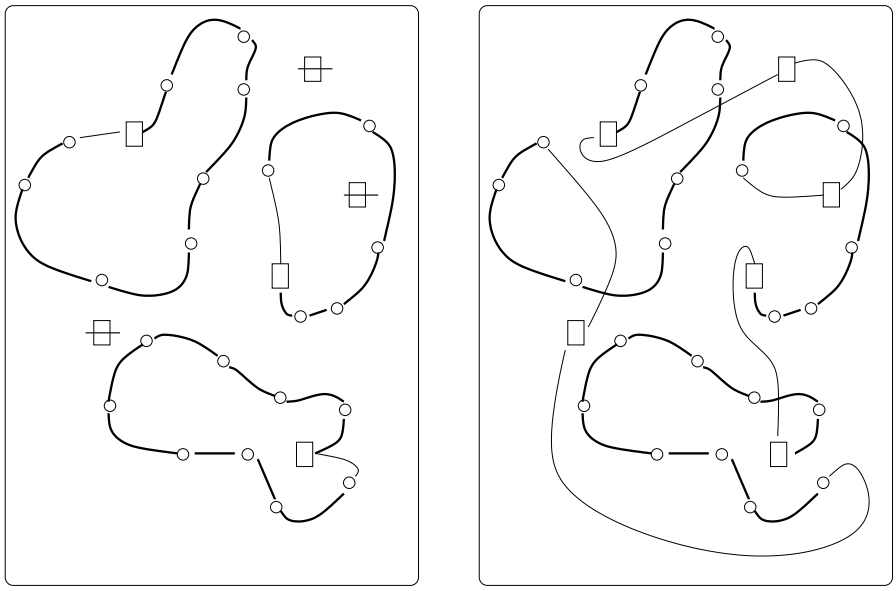
2.3 Lower Bound for the Location-Routing Problem

We now describe a lower bound for the considered LRP that is not derived from the model LR1. The bound consists of two terms. The first one is derived from the costs of the edges that connect clients among them, as well as the costs for starting the routes (connecting plants with clients). The second term is derived from the costs incurred when opening the plants, as well as the costs for terminating the associated routes (connecting clients with plants).

In order to derive the first term we consider an ATSP defined on a complete digraph K_{m+n} where the set of nodes is given by $I \cup J$. That is, there is one node associated with each site for a possible location as well as one node associated with each client. We define the following cost function on the arcs of K_{m+n}:

$$\hat{c}_{ij} = \begin{cases} c_{ij} & \text{if } i \in I, j \in J \text{ or } i, j \in J \\ 0 & \text{otherwise .} \end{cases}$$

The cost function for the ATSP, $\hat{c}$, takes into account the original costs c only for i) the arcs that connect plants with clients, and ii) the arcs that connect clients among them. The $\hat{c}$-costs relative to the arcs that connect clients with plants and plants among them are defined to be 0. Let $\underline{z}_{ATSP}$ denote the value of the optimal solution to the $ATSP$ defined on K_{m+n}. As depicted in Figure 2.8 any feasible set of routes to the considered LRP can be transformed into a feasible solution to the $ATSP$ defined on K_{m+n}, after arbitrarily assigning a direction to the routes of the LRP solution.



a) Solution to the LRP

b) Derived feasible solution for the ATSP

Figure 2.8: Bound on the routing costs

The $\hat{c}$-cost of this transformed solution is equal to the c-cost of the arcs of the routes that connect plants with clients, plus the c-cost of the arcs of the routes that connect clients among them, since those are the only non zero $\hat{c}$-costs. Note that the $\hat{c}$-cost of this transformed solution does not take into account the c-cost of the last stage of the route. That is, it does not consider the cost of connecting the last client of a route with the plant. Therefore, $\underline{z}_{ATSP}$ is a valid lower bound on the routing costs

(if we don't take into account the costs for terminating the routes) for any set of routes that visits all clients.

We next obtain the term derived from the cost for opening the plants and for terminating the routes. For $i \in I$, if plant i were to be open, we would certainly incur a cost f_i but, since some non-empty route should be associated with i, we would also incur the cost for terminating its associated route. Let $j_1(i) \in \arg\min_{j \in J} c_{ij}$ denote the index of the closest client to plant i. Therefore, $\tilde{f}_i = f_i + c_{ij_1(i)}$ is a lower bound on the cost for opening plant i and terminating its associated route. Let $\underline{z}_{KP}$ be the optimal value to the following KP:

$$\underline{z}_{KP} = \text{ minimize } \sum_{i \in I} \tilde{f}_i y_i \tag{2.19}$$

$$\text{subj. to } \sum_{i \in I} b_i y_i \geqslant D \tag{2.20}$$

$$y_i \in \{0,1\},\ i \in I. \tag{2.21}$$

Note that the optimal solution to (2.19)-(2.21) would be the same if the right hand side of constraint (2.20) were $\underline{b}$.

It is clear that $LB = \underline{z}_{ATSP} + \underline{z}_{KP}$ is a valid lower bound on the value of LR1.

In fact, the term $\underline{z}_{KP}$ can be strengthened by taking into account that possibly not all the capacity of a given plant can be actually consumed by the clients' demands. In particular, the *profitable* capacity of a plant $i \in I$, $\bar{b}_i$, is the maximum capacity of the plant that can be consumed by the clients demands. It is given by:

$$\bar{b}_i = \text{ maximize } \sum_{j \in J} d_j z_j$$

$$\text{subj. to } \sum_{j \in J} d_j z_j \leqslant b_i$$

$$z_j \in \{0,1\},\ j \in J.$$

Now, $\underline{z}_{KP}$ can be strengthened to $\underline{z}_{\overline{KP}}$ obtained by solving (2.19)-(2.21) after substituting the coefficients b_i by $\bar{b}_i$.

2.4 Computational Results

In order to evaluate the quality of the proposed lower and upper bounds we have performed a series of computational experiments. To the best of our knowledge there are no available benchmark instances for the LRP described in this work. For this reason we have obtained a set of 125 test instances from the data sets for to the Two-Stage Capacitated Facility Location Routing instances of `http://troubadix.unisg.ch/klose/problems/problems.html\#TSCFLP` that where generated according to Cornuéjols, Sridharan, and Thizy (1991).

Test Instances

Instances are classified in five groups according to their dimensions. Each group consists of twenty five instances. Groups S1, S2 and S3 contain instances with 5 plants and, respectively, 10, 20 and

30 clients. Groups M2 and M3 contain instances with 10 plants and, respectively, 20 and 30 clients. Within each group, we have classified instances according to the ratio between the total capacity and the aggregated demand. The possible values for this ratio are $r = 4.5, 4, 2.5, 2$ and 1.5. Therefore, within each group, there are five subgroups (labeled with a, b, c, d and e) each of which contains five instances with ratio $r = 4.5, 4, 2.5, 2$ and 1.5, respectively.

Since the instances of the above URL correspond to the Two-Stage Capacitated Facility Location Routing Problems, we had to adapt the existing instances to instances for the LRP that we study in this work. To this end, we have proceeded in the following way. First, we have only considered one stage. That is, the plants and clients of our instances correspond, respectively, to depots and customers of the existing instances. Second, when the existing instance had more plants and/or clients than we needed, we have ignored the data of the exceeding plants and/or clients. Finally, we evaluated the ratio $\hat{r}$ between the total capacity and the aggregated demand for each instance, and the demands of the clients were scaled by the factor $r/\hat{r}$, so the resulting instance had the desired ratio r.

Algorithms

All the algorithms are coded in C language. Some routines of the CPLEX 6.5 callable library (?? 1999) are used in the program that solves the LP relaxation RLR1. The exact solution to the ATSP that is solved to obtain the lower bound presented in Section 4, is obtained with the Toth and Carpaneto (1995) code for the ATSP.

The KP that has to be solved to complete the above lower bound is solved with the Martello and Toth (1990) code. This code is also used to obtain the right hand side for constraint (2.11) in the LP relaxation RLR1. The experiments have been performed on a SUN sparc station 10/30 using one of its 4 hyperSPARC processors at 100 MHz., SPECint95 2.35. After some tuning the following values for the parameters were used in the TS heuristic:

- The intensification phase terminates after 2000 iterations or after 1000 consecutive iterations without finding any feasible solution.
- The tabu tenure for a tabu move is 5 iterations, both in the intensification and the diversification phases.
- The overall heuristic terminates after $|I|$ diversification phases or if during an intensification phase no feasible move was found.
- For updating the penalty term $P := P \times \alpha^{\beta}$ in the objective function, α is initially set to the value 2 and updated every $\theta_2 = 100$ iterations according to $\alpha := min\{\alpha + 0.01, 3\}$. The value of β is initially set to 1 and updated every $\theta_1 = 10$ iterations according to $\beta = (\eta/(\theta_1 - 1)) - 1$, where η is the number of infeasible solutions obtained in the last θ_1 iterations.

We have used CPLEX 6.5 to solve exactly the small instances. The value of the optimal solution provides us with a reference to compare our results, both the lower and the upper bounds. Despite their small size, the instances in these groups were very difficult to solve exactly for CPLEX 6.5, especially those in subgroups with smaller value of r (tighter ratio capacity over total demand). For this reason we added some termination criterion to stop the search for the optimal solution when optimality could not be proved within some pre-specified limits. Using this termination rule we were able to optimally solve all instances in group S1, all instances in subgroups S2a and S2b, and three instances in subgroup S2c. We could not find the optimal solution for the remaining 12 instances. In particular, the termination criteria used was to stop the procedure after 48 hours of running time.

Quality of the bounds

Tables 2.1- 2.3 depict a summary of the results obtained with our test instances.

Table 2.1 gives the deviations from the optimal solution for the small instances solved with CPLEX (Groups S1 and S2, with 5 plants and 10 and 20 customers, respectively). For the instances where the optimal solution could not be found, the deviations have been evaluated relative to the valid lower bound provided at termination by CPLEX. In Columns 2-9, each group of two columns under the same heading gives the average value and the maximum value of the percent deviation of our lower or upper bounds from the optimal/lower-bound solution obtained with CPLEX (*best*).

Table 2.1: Percent Deviation from Optimal/Lower-Bound for Small Instances

	% dev lb1		% dev lb2		% dev ub1		% dev ub2		# optima	
	avg	max	avg	max	avg	max	avg	max	CPLEX	ub2
S1a	14.15	23.19	0.88	1.07	12.35	19.03	0.00	0.00	5	5/5
S1b	11.10	18.79	0.89	1.07	8.56	16.36	0.00	0.00	5	5/5
S1c	20.14	23.50	0.71	0.94	20.82	30.13	0.00	0.00	5	5/5
S1d	11.59	21.77	0.69	0.94	3.77	5.75	0.00	0.00	5	4/5
S1e	10.79	13.80	0.56	0.80	11.46	18.24	0.03	0.13	5	4/5
S2a	15.29	26.23	0.69	1.03	9.86	26.66	0.00	0.00	5	5/5
S2b	13.93	29.70	0.67	0.94	2.48	7.24	0.00	0.00	5	5/5
S2c	16.47	18.81	0.46	0.65	21.42	38.56	0.02	0.05	3	1/3
S2d	9.44	13.68	0.45	0.51	8.85	14.64	0.23	0.45	0	
S2e	7.40	19.73	1.88	10.30	6.85	10.48	0.88	2.91	0	

Columns 2-3 (% dev lb1) give the results of the percent deviation (100(*best*-lb1)/*best*) of the lower bound obtained with the LP relaxation RLR1 (lb1). Columns 4-5 (% dev lb2) give the values of the percent deviation (100(*best*-lb2)/*best*) of the lower bound derived in Section 2.3 (lb2) (this bound has been calculated without reinforcing the term $\underline{z}_{KP}$). The next two columns, Columns 6-7, give the results of the rounding heuristic. In particular, columns under % dev ub1 give the percent deviation (100(ub1-*best*)/*best*) of the upper bound obtained with the rounding heuristic (ub1). The results of the TS heuristic are presented in Columns 8-9 (% dev ub2) that depict the values of the percent deviation (100(ub2-*best*)/*best*) of the upper bound obtained with the TS heuristic (ub2). Finally, columns under #optima give the number of instances of each subgroup optimally solved by CPLEX and with our TS heuristic (ub2), respectively.

As can be seen, the lower bounds derived from RLR1 are quite weak, since the average percent deviation from the optimal is nearly always above 10 percent. On the contrary, the lower bounds lb2 derived in Section 4 improve notably on the values of lb1 and give an average percent deviation from the optimal values which never exceeds 1%, excepting for Subgroup S2e for which the average deviation is close to 2%. This bad average is due to the effect of one specific instance where the deviation is as big as 10%. For this instance the term $\underline{z}_{KP}$ of lb2 corresponds to a set of plants with an overall capacity that is exactly equal to the overall demand. We have checked that this set of plants is not feasible (there is no feasible assignment of clients within it). However, if for this instance we calculate lb2 with the reinforced term $\underline{z}_{\overline{KP}}$ the obtained percent deviation reduces to less than 1%. It should also be noted that there is one instance in Subgroup S2e for which our lower bound lb2 is 2% better than the lower bound provided by CPLEX at termination. Despite the simplicity of the rounding heuristic, for all the instances in S1 and S2 a feasible solution was obtained. However, as

Table 2.2: Percent Gaps Between Upper and Lower Bounds for all Instances

	dimension	r	% gap ub1-lb1		% gap ub2-lb2		distribution %gap ub2			
			avg	max	avg	max	$< 1\%$	[1,1.25)	[1.25,5)	> 5
S1a	5×10	4.5	31.99	48.50	0.99	1.15	1	4	0	0
S1b		4	22.62	41.50	0.98	1.15	1	4	0	0
S1c		2.5	51.53	70.10	0.86	1.15	3	2	0	0
S1d		2	17.82	29.45	0.73	0.99	5	0	0	0
S1e		1.5	25.12	37.17	0.65	1.04	4	1	0	0
S2a	5×20	4.5	31.64	59.47	0.78	1.15	4	1	0	0
S2b		4	20.90	42.24	0.69	0.94	5	0	0	0
S2c		2.5	45.74	70.66	0.58	0.77	5	0	0	0
S2d		2	25.27	43.83	0.66	0.77	5	0	0	0
S2e		1.5	16.33	36.37	3.24	13.17	4	0	0	1
S3a	5×30	4.5	43.64	102.85	1.01	1.13	2	3	0	0
S3b		4	50.89	99.75	0.96	1.13	2	3	0	0
S3c		2.5	31.27	60.84	0.61	0.68	5	0	0	0
S3d		2	27.51	54.39	0.59	0.71	5	0	0	0
S3e		1.5	14.27	32.05	0.48	0.61	5	0	0	0
M2a	10×20	4.5	43.62	76.68	0.91	1.04	4	1	0	0
M2b		4	36.76	59.76	0.94	1.11	3	2	0	0
M2c		2.5	24.46	42.16	1.25	2.78	3	1	1	0
M2d		2	13.43	22.45	0.8	1.15	4	1	0	0
M2e		1.5	10.89	10.89	0.84	0.9	5	0	0	0
M3a	10×30	4.5	38.64	69.52	0.74	0.82	5	0	0	0
M3b		4	37.91	56.90	1.28	3.34	4	0	1	0
M3c		2.5	21.39	32.12	0.78	0.93	5	0	0	0
M3d		2	12.05	24.04	0.57	0.69	5	0	0	0
M3e		1.5	9.81	12.71	0.63	1.02	4	1	0	0

could be expected, the upper bounds obtained with the rounding heuristic are not very good and some of the average deviations (% dev ub1) are above 20%. This is because the rounding heuristic tends to open plants with low ratio f_i/b_i even if their capacity is larger than necessary. On the contrary, the results obtained with the TS heuristic are very satisfactory, since 34 out of the 38 instances optimally solved by CPLEX, were also optimally solved with our heuristic. The small values of the deviations dev ub2 show that for the instances that were not optimally solved the quality of the obtained solution is always very good.

Columns 2 and 3 in Table2.2, "dimension" and r respectively, give the dimensions of the instances (plants $\times$ customers) and the value of the parameter r. In the main, the results of Table2.1 are confirmed by those of Table2.2 where we show percent gaps between the upper and lower bounds. In Columns 4-7 each group of two columns under the same heading gives the average and the maximum value of the percent deviation of the upper bound from the lower bound. In particular, Columns

4-5 (% gap ub1-lb1) give the percent deviation (100(ub1-lb1)/lb1) of the upper bound obtained with the rounding heuristic (ub1) from the lower bound obtained with the LP relaxation RLR1 (lb1). Similarly, Columns 6-7 (% gap ub2-lb2) give the percent deviation (100(ub2-lb2)/lb2) of the upper bound obtained with the TS heuristic (ub2) from the lower bound of Section 2.3 (lb2). For instances in groups S1 and S2 the reader can check that the values of Columns 4 and 6 of Table2.2 are indeed an upper bound of the sum of the individual deviations of the lower and the upper bounds from the optimal solution, shown in Columns 2 and 6 of Table 2.1 for % gap ub1-lb1, and in Columns 4 and 8 of Table 2.1 for % gap ub2-lb2. For the instances in Groups S3, M2, and M3 the obtained results follow the tendency of the smaller instances. More specifically, the rounding heuristic succeeded in finding a feasible solution in 118 out of the 125 instances although, in general the quality of the upper bound is quite poor.

The values of % gap ub1-lb1 show again the weakness of both the lower bound lb1 and the upper bound ub1, whereas the values of % gap ub2-lb2 are in general small, confirming the good results obtained for the small instances with the lower bound lb2 and the upper bound ub2. Figure 2.9 gives an example of the evolution of % gap ub2-lb2 in the TS heuristic for one instance of dimension 10×20 that was selected arbitrarily for illustration. The thin dotted lines represent the times when diversification steps took place whereas the thick break points represent the times where the upper bound ub2 was updated, and the actual values of % gap ub2-lb2.

More insight about the quality of ub2 and lb2 can be obtained from the last columns of Table 2.2 that depict the number of instances of each subgroup for which % gap ub2 is between some fixed threshold. It can be observed that for 98 out of the 125 instances the gap between ub2 and lb2 is below 1%. For 24 instances this percent gap is in $[1, 1.25)$, for 2 instances it is in $[1.25, 5)$, and only for 1 instance this gap is above 5%.

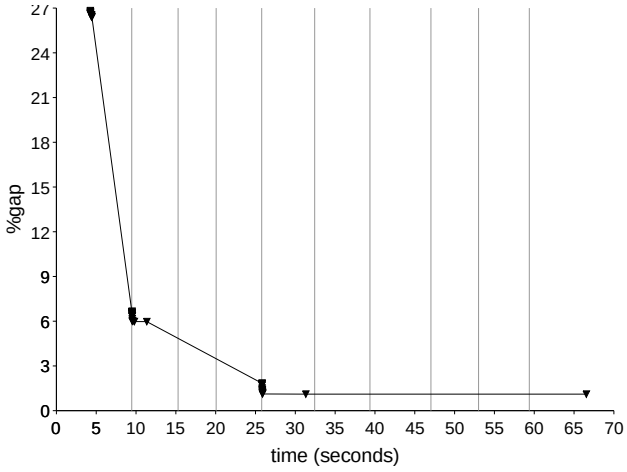


Figure 2.9: Evolution of $ub2$ on Problem 3M2b

We have further investigated the instances for which the % gap ub2-lb2 is big. In one case, the gap reduces from 13.17% to 0.67% when the lower bound lb2 is calculated with the strengthened term $\underline{z}_{\overline{KP}}$. In the remaining cases, it is the bad quality of the upper bound ub2 what causes this big percent gap. Values of ub2 far from the optimum seem to occur when the TS procedure does not succeed in finding the optimal set of plants to open.

CPU Time requirements

The CPU times required to obtain the lower bounds lb1 and lb2, and the upper bound ub2 are depicted in in Columns 2-7 of Table 2.3. Again each group of two columns under the same heading gives the average and the maximum values. We have not decomposed "time lb2" in the times required by each of the two procedures used to obtain lb2 (the time required to solve the ATSP, and the time required to solve the KP) because the time for solving the KP is negligible with respect to the time for solving the ATSP. Note that the lower bound lb2 is obtained with a small computational effort, since the values in "time lb2" are quite small and considerably smaller than the corresponding values in "time lb1". As to the times required to obtain the upper bound ub2, again we can conclude that they are good taking into account *i*) the difficulty for optimally solving the instances with CPLEX, and *ii*) the quality of the bounds obtained with the TS heuristic. Columns 8-9 of Table 2.3, show the times required by the TS heuristic to obtain the best solution found. This information is also depicted in Figure 2.10, where each stack bar corresponds to one test instance and represents the total time spent by the TS heuristic on the instance. The dark part of each bar shows the time required to obtain the best solution found, so the light part of each bar shows the time spend since the best solution was found until termination. As can be seen, the times to obtain the best solutions are considerably smaller than the total times. Thus, we could have adjusted the termination criteria of the heuristic to obtain similar results in smaller computation times.

As usual, we can observe an increase in the computational times as the size of the instances grows. We have not performed a rigorous analysis but the times required to solve RL1 seem to increase approximately quadratically with $|I| + |J|$. For the TS heuristic, the increase of times seems to be approximately quadratic with $|I| \cdot |J|$. As to the times to obtain lb2, they are very small. It is surprising that for many instances in the group M computation times are smaller than for other instances in group S, even with the same number of customers. We attribute this to the fact that a code for the ATSP is used. Note that the only source of asymmetry in the ATSPs that are solved in this work is related to plants. Thus, instances in group S are much more symmetric than instances in group M.

As expected, we have not appreciated any influence of the value of the parameter r on the difficulty for solving RL1. For the lower bound lb2 we can appreciate a slight increase on the computational effort as the value of r decreases (for instances of the same size). In the case of the TS heuristic, the influence of the value of r on the required times is smaller to what we thought it would be when designing the computational experiments. Nevertheless, in general terms, instances with intermediate values of r seem to require higher times.

Table 2.3: CPU Times and Solutions' Characteristics

	time lb1		time lb2		time ub2		time best found		% used cap		plants open	
	avg	max	avg	max	avg	max	avg	max	avg	max	avg	max
S1a	0.28	0.30	0.02	0.03	4.46	5.68	0.28	0.29	85.49	92.67	1	1
S1b	0.27	0.29	0.03	0.04	5.30	7.02	0.27	0.30	88.45	99.57	1	1
S1c	0.27	0.32	0.02	0.03	5.42	6.64	2.00	2.90	95.79	100	2	2
S1d	0.28	0.31	0.02	0.03	5.28	6.11	1.62	1.82	94.62	98.48	2	2
S1e	0.25	0.26	0.02	0.04	5.43	7.02	1.89	2.04	95.86	98.58	3	3
S2a	1.26	1.37	0.09	0.18	22.90	31.17	1.20	1.37	84.59	94.85	1	1
S2b	1.22	1.37	0.09	0.16	23.54	30.00	1.20	1.38	80.32	92.71	1	1
S2c	1.25	1.42	0.10	0.17	25.13	28.38	8.11	10.50	93.19	99.37	2	2
S2d	1.30	1.43	0.14	0.25	23.43	28.12	9.95	15.77	96.51	99.11	2.2	3
S2e	1.34	1.42	0.14	0.28	20.23	26.52	9.63	17.22	96.75	100	2.8	3
S3a	4.53	4.95	0.24	0.48	52.55	83.72	7.93	21.89	83.47	95.64	1.2	2
S3b	4.77	5.88	0.24	0.51	64.91	97.66	7.06	17.31	72.85	98.56	1	1
S3c	4.69	5.70	0.24	0.51	71.50	86.91	18.23	25.12	87.27	94.35	1.8	2
S3d	4.72	5.50	0.39	0.82	75.11	97.87	11.93	23.76	89.86	99.68	2	2
S3e	5.26	6.14	0.39	0.84	50.86	77.15	26.00	35.23	95.22	99.28	2.8	3
M2a	4.05	4.60	0.04	0.06	68.88	74.95	15.52	28.95	96.57	99.41	2	2
M2b	3.72	4.21	0.04	0.05	74.36	79.08	14.20	31.38	98.13	99.12	2	2
M2c	3.94	4.42	0.04	0.06	89.85	91.42	13.51	16.98	97.82	100	3.2	4
M2d	4.01	4.28	0.08	0.11	93.42	95.64	16.27	37.26	98.82	99.05	3.8	4
M2e	3.98	4.19	0.08	0.09	77.99	94.04	9.46	13.39	99.36	100	5	6
M3a	19.76	27.20	0.13	0.34	203.58	228.16	38.80	46.99	95.85	99.64	2	2
M3b	21.33	25.20	0.14	0.33	215.97	229.12	52.69	88.76	95.87	100	2	2
M3c	17.43	24.82	0.13	0.33	236.97	247.79	41.98	56.22	99.66	100	3	3
M3d	15.52	19.40	0.26	0.52	236.61	241.26	42.10	56.40	98.51	100	3.8	4
M3e	17.47	22.62	0.26	0.53	162.08	229.96	52.26	86.01	98.87	99.82	5.4	6

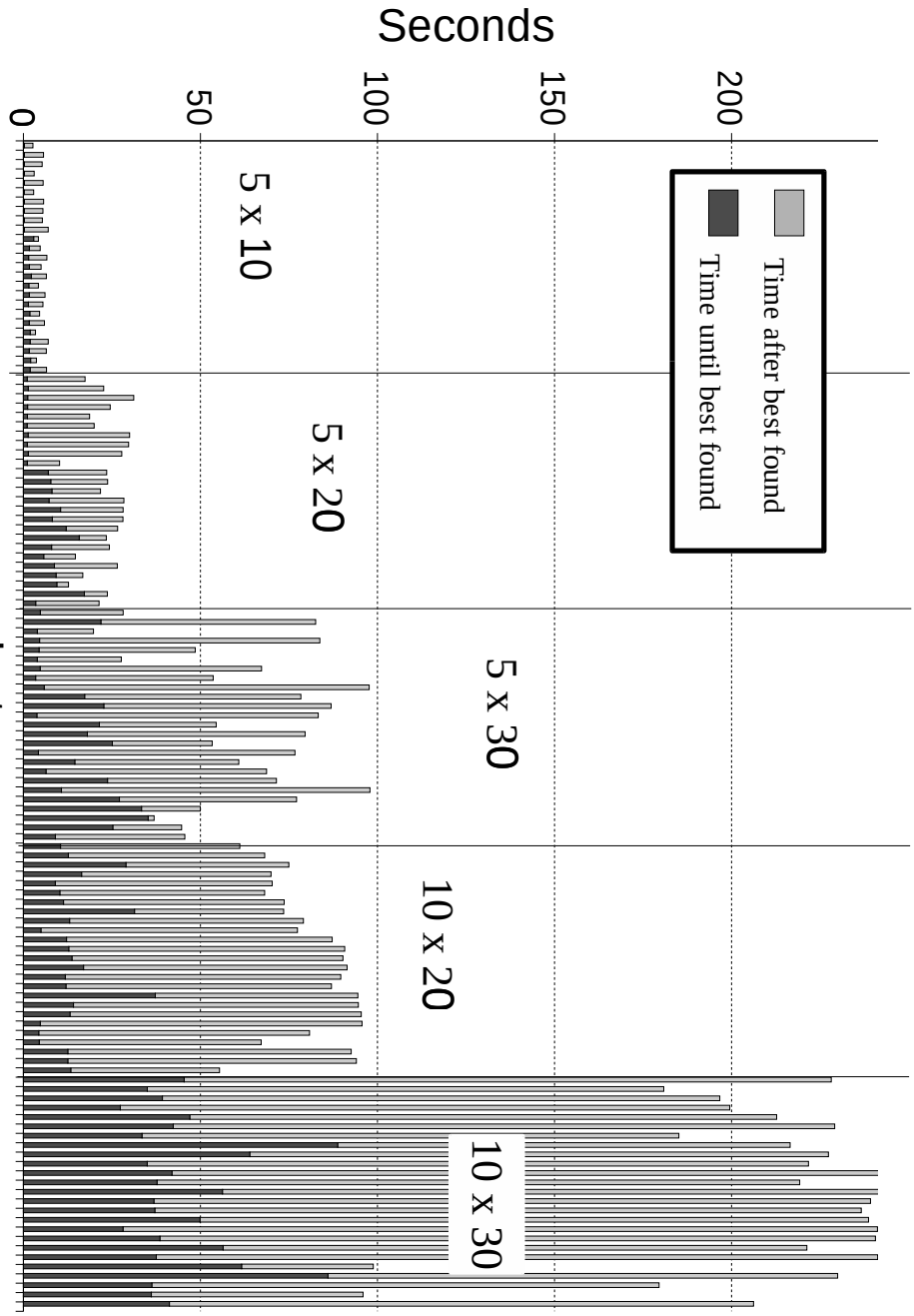


Figure 2.10: Tabu Search:Time to Find the BestSolution vs. Total CPU Time

The Solutions

Columns 10-13 in Table2.3 give information about the structure of the best solutions found with the TS heuristic. In particular, Columns % used cap show the percentage of the overall capacity of the set of open plants that is consumed by the clients' demands, and Columns plants open show the number of open plants in the best solutions of the TS heuristic. We can see that the ranges for the values % used cap are, in general very high and that in some cases, the maximum value of a subgroup is 100%, which means that the procedure tends to open as few plants as possible. To a large extent this can be explained by the fact that in the considered instances the opening costs are much larger than routing costs. We should expect that if the opening costs were smaller, the number of open plants would increase and the percent of consumed capacity would tend to decrease.

2.5 Conclusions

In this section we have presented a single source LRP with capacitated facilities and one single vehicle for each open facility. We use an auxiliary network to model this LRP as a network flow problem with side constraints instead of using the three index model proposed in other works for general LRPs. The solution to a reinforced linear relaxation of this model is used in the algorithmic approaches we propose. A deeper study of the model itself as well as the derivation of valid inequalities for it are envisaged as a subject of future research.

We propose a TS heuristic that provides solutions of high quality. In particular, 34 out of the 38 instances for which the optimal solution is available were solved to optimality, and only in one case the percent deviation with respect to the optimum went over the 1%. The main features of this TS are the simple neighborhoods explored and the strategic oscillation scheme that monitors the crossings of the feasibility border along the search. Time requirements for this heuristic are reasonable taking into account the quality of the solutions obtained and the difficulty of the problem.

The initial solution for the TS is provided by a simple rounding heuristic that derives a feasible solution from the solution to the LP relaxation of the model we propose. In spite of being extremely simple, this rounding heuristic never failed to obtain a feasible solution, even though the quality of such solutions was often low since, in general, the required overall capacity of the set of open plants was overestimated.

Finally, we present a lower bound. Broadly speaking, it is obtained from bounding separately the location and the routing components of the problem. The results obtained with this approach are satisfactory despite the fast growth of the CPU time requirement as the number of customers increases. This is due to the fact that an ATSP has to be solved and the only asymmetries the instances contain are related to the plant sites. So, the lower the proportion between the number of plants and the number of customers is, the more symmetric the problem becomes, and its resolution by means of algorithms designed for the ATSP becomes more difficult. For instances with higher number of customers than those reported in this work we propose to use lower bounds for the TSP instead of solving it exactly.