

Solving an Urban Waste Collection Problem Using Ant Heuristics

Joaquín Bautista¹, Elena Fernández², Jordi Pereira³

¹ Càtedra Nissan

Universitat Politècnica de Catalunya. Barcelona. Spain.

² Dpt. d' Estadística i Investigació Operativa

Universitat Politècnica de Catalunya. Barcelona. Spain.

³ Departament d'Organització d'Empreses

Universitat Politècnica de Catalunya. Barcelona. Spain.

e-mails: {joaquin.bautista, e.fernandez, jorge.pereira}@upc.edu

Abstract

This paper describes the methodology that we have applied for the solution of an urban waste collection problem in the municipality of Sant Boi de Llobregat, within the metropolitan area of Barcelona (Spain). The basic nature of the considered problem is that of a capacitated arc routing problem, although it has several specific characteristics, mainly derived from traffic regulations. We present the model that we have built for the problem, which results after an appropriate transformation of the problem into a node routing one. We also present the ant colonies heuristics that we have used to obtain the solutions to the problem. These combine constructive methods, based on nearest neighbor and on nearest insertion, with a local search that explores various neighborhoods. The application of the proposed methods gives results that improve considerably the ones that were previously used in the municipality.

1 Introduction

The management of urban waste is now undergoing a strong change and innovation phase as required by the growing environmental concern from citizens and governments. As a result, in recent years there has been an increasing number of directives from the European Union and regulations in the different countries. In particular, in 2000 the National Plan for Urban Waste (PNRU, from its name in Spanish) gave some unifying criteria to be applied in Spain in this field. According to the plan, municipalities with more than 5,000 inhabitants are forced to ensure proper separation for selective collection of garbage. New and interesting management problems arise in the affected municipalities: what collection system should be applied, where to locate the garbage accumulation areas as well as the collection points, how many containers and of what type should be assigned to each area, which are the most appropriate collection routes, which frequency of collection should be applied in each section, how big the fleet of vehicles should be, and so on.

In Catalonia alone, there are 120 municipalities affected by this regulation, 28 of which belong to the metropolitan area of Barcelona. Sant Boi de Llobregat, is one such municipality with more than 80,000 inhabitants. The need to adapt to the new legislation and the desire to renew equipment and to design a new physical collection system led the Council and the municipal transportation company, to sign an agreement with the Technical University of Catalonia in 1999 as described by Bautista in [1]. One result of this collaboration is the decision support system Integrated System of Selective Urban Garbage Collection (SIRUS, from its name in Spanish). SIRUS was designed to help decision-making in problems related to the management of garbage collection, on the basis of a set of conditions and criteria given by the new regulations. This work deals with one of the aspects of the above mentioned project. Namely, the design of collection routes in Sant Boi de Llobregat and the integration of the proposed method within SIRUS.

Similarly to other countries, in Spain waste generated in urban areas is collected by municipal organizations or its collection is run by private companies under some type of contract. Urban waste is collected and transported to its final destination, trying to offer a quality efficient service at minimum cost. Like in other countries of the European Union, in Spain most of the population is concentrated in urban areas with a very high population density, given that people usually live in apartment buildings. As a result, the collection technique has evolved away from door-to-door collection, very common in the USA, and the predominant system consists of collection points located through the city where citizens leave their refuse. Each collection point is made up of one or more refuse bins (with dimensions ranging from 1 to 3.2 m³). Different trucks collect each type of waste and transport the waste to their final destination (a refuse dump, an incinerator or a recycling plant). One key factor for proper operation of collection systems is the design of appropriate collection routes for each type of garbage (fraction), which may or may not be separately collected (in some cases due to technical design constraints and different treatment processes). In Sant Boi de Llobregat each fraction is collected separately with a fixed periodicity (established by the municipality). Therefore, in this case the collection of each type of garbage can be considered as an independent but equivalent problem. In addition, the routes should be coherent with traffic rules, taking into account forbidden turns and traffic signals while minimizing collection costs.

Note that the considered problem is an arc routing one since in the actual circumstances of the problem roads are traversed completely, even if collection points are located in specific points of the roads. In addition the capacity of the trucks is limited. Thus, the basic nature of the considered problem is that of a capacitated arc routing problem (CARP). Even if they have not been so extensively studied as their node routing counterpart, there is a rather vast literature on arc routing problems (see [10] for a comprehensive text) and on the CARP in particular (see, for instance, [13, 17, 18, 2]). Nevertheless, the problem that we address presents several characteristics that make it different from classical arc routing problems. These characteristics are mainly derived from traffic regulations. On the one hand, the underlying graph of the problem is a mixed one with both arcs and edges, since in reality some streets can be traversed in both directions whereas others have only one direction. To the best of our knowledge the CARP has not been studied on mixed graphs. On the other hand, at the junctions of the streets some turns will be forbidden, whereas other will be allowed. Again, to the best of our knowledge this is an issue that has not been considered for capacitated arc routing problems. For problems without capacities forbidden turns have been considered for routing and scheduling street sweepers in [3, 4] and for the Rural Postman Problem on mixed graphs with turn penalties in Corberán et al. [6], where the authors propose a solution approach based on penalizing forbidden turns and post-processing illegal solutions. In [22] the formulation of the Capacitated Chinese Postman Problem is adapted to the context of the Canada Post Corporation and a general heuristic is presented to solve the problem under several hypothesis where, again, penalties are assigned to forbidden turns.

In this work we build a model for the Mixed Capacitated Arc Routing Problem (MCARP) where forbidden turns are taken into account in the set of constraints instead of penalizing forbidden turns via the objective function. We believe that this is more appropriate when the number of forbidden turns is high, like in the real problem that we have addressed. In Section 2 we describe the problem and in Section 3 we present the proposed model, which results from a transformation of the problem into a node routing one. In [16] Laporte proved that this strategy can be successful for addressing several classes of arc routing problems. It has also been applied to the Rural Postman Problem on mixed graphs with turn penalties in Corberán et al. [6]. Our solution approaches are presented in Section 4: Subsections 4.1 and 4.2 describe two constructive heuristics to obtain feasible solutions, and the neighborhoods that are explored in the Local Search, respectively. These elements are integrated within two heuristics based on ant colonies (see Dorigo and Stützle [9]) that are presented in Subsection 4.3. Section 5 presents the results obtained when applying the algorithms of Section 4 to the city of Sant Boi de Llobregat. The routes generated with our heuristics are compared with the ones that were previously used, obtaining significant improvements. Finally, in Section 6, we present the conclusions of the work and we give some comments about the integration of the algorithms within SIRUS.

2 The problem

For the scope of this paper, the area of the city where the collection routes are planned is represented by means of a graph $G_0 = (V_0, E_0, A_0)$. The vertices of the graph, V_0 are associated with street junctions and dead-ends, whereas the edges, E_0 , and arcs, A_0 , are associated with

existing street sections between street junctions. Arcs represent single-way streets while edges represent two way streets. Any element of $E_0 \cup A_0$ will be called a link. There is one single depot that throughout the paper will be denoted as vertex 0. Of course, $0 \in V_0$. G_0 is in fact a multi graph, since multiple streets (corresponding to parallel edges or arcs) may connect two junctions. In addition, there is a set of rules derived from traffic regulations, that determine when it is possible to connect two adjacent links through a common vertex as well as some turn limitations associated with adjacent links when the common angle is too narrow. We will refer to those regulations as *turn constraints*. Each link $a \in E_0 \cup A_0$ has a length l_a . The set of required links $R_0 = E_0^R \cup A_0^R$, is defined as the set of links with at least one collection point. Each required link $a \in E_0^R \cup A_0^R$ has a demand d_a that represents the quantity of waste that must be collected while servicing the link a , and corresponds to the total waste generated by the citizens allocated to the associated collection point as explained by Pereira in [21]. There is an homogeneous fleet of vehicles, indexed in a set K , with capacity D , to provide service to the required links. Each required link must be serviced by one single route although, in addition, other routes might traverse it. The cost of a route is given by the sum of the lengths of the links that it services or traverses. The Mixed Capacitated Arc Routing Problem with Turn Constraints (MCARPTC) consists in finding a set of $|K|$ routes such that

- each route starts and ends at the depot,
- each required link is serviced by one route,
- the total demand of the links that are serviced by each vehicle does not exceed the capacity of the vehicle,
- the rules derived from traffic regulation are satisfied,

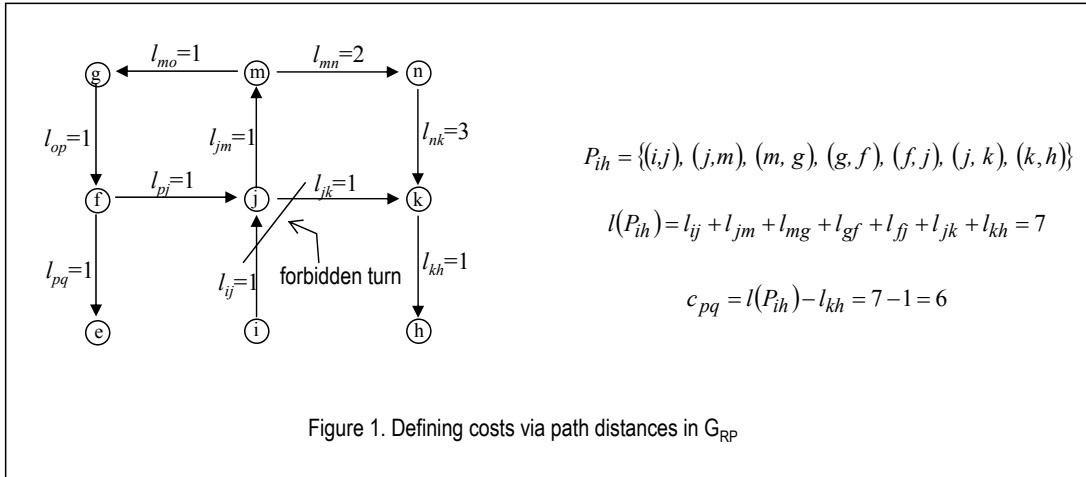
of minimum total cost.

Remark 1

We say that a path in G_0 is *feasible for the rules*, if it does not violate any of the rules and turn constraints derived from traffic regulation. Note that for any pair of vertices $i, j \in V_0$ it is possible to know if there exists an $i - j$ -path that is feasible for the rules, and to determine the value of the shortest path that is feasible for the rules, with respect to the arc lengths l . This can be achieved with an *ad-hoc* modification of the Floyd-Warshall algorithm [11, 23], although there also exist more efficient procedures based on the Dijkstra algorithm (see for instance [5]). Let $P_{i,j}$ denote a shortest $i - j$ -path feasible for the rules in G_0 , with value $l(P_{i,j})$.

Next, we will transform the arc routing problem into a node routing one. A general framework for this type of transformation for arc routing problems without capacity constraints and without turn penalties is given by Laporte in [16], and a different transformation (which basically results on the same graph) and takes turns into account is also given by Corberán et al. in [6]. Our transformation is based on the one of [16], although it incorporates some new features to handle turn penalties and capacity constraints. Also, we should mention that the way we handle turn penalties is based on the path distances resulting from Remark 1, which is simpler than the transformation used in [6].

Let $G_{RP} = (V, A)$ denote the new directed graph. Vertices in G_{RP} correspond to required links in G_0 : one vertex when the link is a required arc, and two vertices (one associated with each of the two possible traversal directions for the edge) when the link is a required edge. In particular, each required arc $a_{ij} \in A_0^R$ defines a vertex $p(a_{ij}) \in V$, whereas each required edge $e_{ij} \in E_0^R$ defines two vertices, $p(a_{ij}), p(a_{ji}) \in V$. In addition, vertex $0 \in V$, corresponding to the depot. For notational purposes we will partition the vertex set into disjoint sets (we will refer to them as *clusters*), i.e. $V = T_0 \cup T_1 \cup T_2 \cup \dots \cup T_r$ with $T_0 = \{0\}$. For $t = 1, \dots, r$, each vertex set T_k either is a singleton, when the vertex is associated with one arc of A_0^R , or contains the two vertices $p(a_{ij}), p(a_{ji})$ associated with the two opposite arcs of edge $e_{ij} \in R_0$. For each vertex $p \in V$, C_p denotes the index of the cluster it belongs to. That is, $C_p = r \iff p \in T_r$. The set of arcs, A , contains one arc (p, q) for every pair of vertices $p = p(a_{ij}), q = q(a_{kh}) \in V$, with $C_p \neq C_q$, such that in G_0 there exists a directed path connecting i and h , that is feasible for the rules. Observe that G_{RP} need not be a complete graph. The cost of an arc $(p, q) \in A$, with $p = p(a_{ij})$ and $q = q(a_{kh})$, is given by the length of the shortest $i - k$ -path in G_0 that is feasible with the rules. When $j = k$ and no turn is allowed from (i, j) to (k, h) , the lengths of all the additional arcs that must be traversed for connecting (i, j) and (k, h) in a feasible manner must be taken into account. That is, $c_{pq} = l(P_{ih}) - l_{kh}$. Note that the resulting cost matrix is asymmetric. Note also that the cost c_{pq} accounts for the lengths of all the arcs in the feasible path P_{ih} . This is a difference with the path-distance costs defined in [16] that do not account for the lengths of the required links. In principle, the lengths of the required links need not be computed since any feasible solution will service all required links exactly once and, thus, the total length of the serviced edges is a constant that does not affect to the solution of the optimization problem. However, when turn regulations have to be taken into account, the path distance between two adjacent arcs cannot be defined as zero if the turn through the common endnode is forbidden. On the contrary, it has to incorporate the lengths of all the arcs used to define the feasible path (see Figure 1).



We define a demand function in G , which is inherited from the original demand function in G_0 . Abusing notation, if $p = p(a_{i,j}) \in V$, we denote d_p to the demand of the required link (i, j) . In what follows, unless it is necessary, we will usually drop the reference to the link of R_0

associated with a given vertex and we will just write $p \in V$ instead of $p(a_{ij}) \in V$. To represent vertices associated with two opposite traversal directions, $p(a_{ij})$ and $p(a_{ji})$, we will use p and \bar{p} .

MCARPTC is equivalent to an asymmetric node routing problem on G_{RP} that consists of finding a set of $|K|$ routes in such a way that:

- each cluster is visited by exactly one route,
- for each cluster, exactly one of its vertices is visited by some route,
- the total demand of the vertices visited by each route does not exceed the capacity of the vehicle,

of minimum total cost. The above problem is an extension of the Vehicle Routing Problem to clusters of vertex sets known as the Generalized Vehicle Routing Problem (GVRP). Since in each cluster the vertex that is going to be visited has to be decided, in [15] GVRP is considered by Laporte as a location-routing problem. On the contrary, in [12] Ghiani and Improta transform the problem into a Capacitated Arc Routing Problem. On the other hand, the rules used by Noon and Bean in [20] to transform a Generalized Traveling Salesman Problem into a Traveling Salesman Problem can also be used to transform a GVRP into a VRP. Capacity can be taken into account by assigning the demand of the cluster to only one of its vertices (arbitrarily chosen). Nevertheless, given that the cardinality of each cluster is either one or two, the model that we propose in the next section does not use this last transformation and is built for the $GVRP$ on the graph G_{VR} .

3 The model for MCARPTC

For building our model for the problem we define the following decision variables:

$$x_{pqk} = \begin{cases} 1 & \text{if arc } (p, q) \in A \text{ belongs to route } k \in K; \\ 0 & \text{otherwise.} \end{cases}$$

We use the following additional notation: $\forall S \subset V$, $A(S) = \{(p, q) \in A : p, q \in S\}$; $\forall p \in V$, $\delta^+(p) = \{q \in V : (p, q) \in A\}$, $\delta^-(p) = \{q \in V : (q, p) \in A\}$. Note that if $C_p = C_q$, then $(p, q) \notin A$ and $(q, p) \notin A$. Thus, for $p \in T_t$, for some $t = 1 \dots r$, then $\delta^+(p) \notin T_t$ and $\delta^-(p) \notin T_t$.

Then, the model is:

$$(GVRP) \min \sum_{k \in K} \sum_{(p,q) \in A} c_{pq} x_{pqk} \quad (1)$$

$$\text{s.t.} \quad \sum_{k \in K} \sum_{p \in T_t} \sum_{q \in \delta^-(p)} x_{qpk} = 1 \quad t = 1, \dots, r \quad (2)$$

$$\sum_{k \in K} \sum_{p \in T_t} \sum_{q \in \delta^+(p)} x_{pqk} = 1 \quad t = 1, \dots, r \quad (3)$$

$$\sum_{k \in K} \sum_{q \in \delta^-(0)} x_{q0k} = |K| \quad (4)$$

$$\sum_{k \in K} \sum_{q \in \delta^+(0)} x_{0qk} = |K| \quad (5)$$

$$\sum_{q \in \delta^-(p)} x_{qpk} = \sum_{q \in \delta^+(p)} x_{pqk} \quad t = 1, \dots, r; p \in T_t, \forall k \in K \quad (6)$$

$$\sum_{p \in V} d_p \left[\sum_{q \in \delta^+(p)} x_{pqk} \right] \leq D \quad \forall k \in K \quad (7)$$

$$\sum_{(p,q) \in A(S)} x_{pqk} \leq |S| - |K| \quad \forall S \in V, k \in K \quad (8)$$

$$x_{pqk} \in \{0, 1\} \quad \forall (p, q) \in A, k \in K \quad (9)$$

Constraints (2) and (3) ensure that for each cluster of vertices T_r , exactly one route “enters” the cluster and exactly one route “leaves” the cluster. Similarly, constraints (4) and (5) require that the $|K|$ routes start and end at the depot. Constraints (6) are needed to guarantee that, the route that “enters” the cluster is the same than the route that “leaves” the cluster, and also, that when a route visits a cluster, the same vertex is visited when entering and when leaving the cluster. The capacity constraints (7) ensure that the capacities of the vehicles are not violated, whereas constraints (8) are the well-known subtour elimination constraints (SEC). The model has $|A| \times |K|$ binary variables and a number of constraints that is exponential in $|V|$ due to constraints (8). The number of constraints can be reduced if the SEC’s are expressed as proposed by Desrochers and Laporte in [7] for the VRP based on the SEC’s of Miller, Tucker and Zemlin [19] for the TSP. In particular, they can be expressed as

$$u_{pk} - u_{qk} + D x_{qpk} \leq D - d_q \quad \forall k \in K, \forall (p, q) \in A, p \neq 0 \text{ s.t. } d_p + d_q \leq D \quad (10)$$

$$d_p \leq u_{pk} \leq D \quad \forall p \neq 0, k \in K \quad (11)$$

where u_{pk} is an additional set of continuous variables.

4 Heuristics for the GVRP

In this section we describe the heuristics that we use for the *GVRP*. We are not aware of any heuristic previously proposed in the literature for the GVRP and the previous work on heuristics for the asymmetric Vehicle Routing Problem is quite scarce. The constructive and local search methods that we use take into account forbidden turns as well as the constraint that one single node is visited in each cluster. For the way they have been obtained, forbidden turn constraints are implicitly considered both in the graph G_{VR} , that is not necessarily complete, and in the cost function that has been defined. Therefore, we only need to take care explicitly of how to handle the constraints that require that one node per cluster is visited. Asymmetry does not add any specific difficulty to constructive methods. However, it increases notably the difficulties for local search. The two ant heuristics presented in Subsection 4.3 use adaptations of the deterministic constructive methods presented in Subsection 4.1 as well as the local search of Subsection 4.2.

4.1 Constructive methods

Two different constructive greedy heuristics have been used. The first one is a nearest neighbor method and the second one is some type of nearest insertion heuristic. In both methods the constraints that require that one vertex per cluster is visited are achieved basically by considering the sets of clusters (instead of the set of vertices) and by enumerating the two possibilities for appending or inserting one vertex of the cluster in the partial solution when a cluster is selected. For $s = 1, \dots, r$ we will denote \widehat{d}_r to the demand of the vertex (vertices) of T_r .

Nearest Neighbor Heuristic

Routes are built one at a time. When Route k is being constructed, clusters are sequentially assigned to it using the nearest neighbor criterion among the clusters that would not violate the capacity constraint. The adaptation of the nearest neighbor criterion to this case requires that for clusters with two vertices, two different possibilities are evaluated for estimating the cost of appending one of the nodes of the cluster to the partial solution that is being built. In particular, at each step of the heuristic $q_t \in \operatorname{argmin}\{c_{pq} : q \in T_t\}$, denotes the vertex of cluster t that gives the best way of connecting one vertex of cluster t to vertex p . At each step of the heuristic d_{min} denotes the minimum demand among the clusters that are not yet assigned to any route. This value is updated each time that a cluster is assigned to one route. A route terminates when the available capacity of the route does not allow to add any new vertex. A scheme of the heuristic is given in Algorithm 1.

Nearest Insertion Heuristic

This heuristic consists of two phases. Initially, clusters are ordered from furthest to closest to the depot. The distance of cluster T_s to the depot is given by the minimum distance to/from the depot of any of its vertices; i.e. $\min\{\min\{c_{0p} : p \in T_s\}, \min\{c_{p0} : p \in T_s\}\}$. For ease of presentation we will assume that clusters are indexed according to that ordering. The first phase of the heuristic builds $|K|$ partial routes one at a time. For initiating Route k , a seed is selected. The seed is one of the vertices of the cluster that is furthest to the depot, among the ones not visited so far by any of the previous $k - 1$ partial routes. Specifically, the seed is the vertex of

Algorithm 1 Nearest Neighbor

```
 $L = \{T_1, T_2 \dots, T_r\}$ 
for  $k = 1, \dots, K$  do
   $R_k = \emptyset; p = 0; B = D; d_{min} = \min\{\widehat{d}_t : t \in L\}$ 
  while  $(B - d_{min} \geq 0)$  do
     $q \in \operatorname{argmin}\{c_{pq_t} : T_t \in L, q_t \in T_t, \widehat{d}_t \leq B\}; s = C_q$ 
     $R_k := R_k \cup \{(p, q)\}$ 
     $L := L \setminus \{T_s\}$ 
     $B := B - \widehat{d}_s$ 
     $p := q$ 
  end while
   $R := R \cup \{(q, 0)\}$ 
end for
```

that cluster which is closest to the depot (in any of the two directions). Let $s_k \in V$ denote the seed at iteration k , and let also I_k denote the set of intermediate vertices in the shortest paths \hat{P}_{0,s_k} and $\hat{P}_{s_k,0}$, in G_{VRP} connecting the depot with s_k , and s_k with the depot, and suppose that these vertices are ordered from furthest to closest to the depot. Then, vertices in I_k are considered in turn and included in the initial route if there is still enough available capacity. Let $\hat{I}_k = \{0, p_1^k, p_2^k, \dots, p_{n_k}^k\}$ denote the set of vertices that are assigned to route k , ordered according to their place in the route, and B^k the available capacity of route K at the end of this phase.

For the second phase all partial routes are considered simultaneously. For each route, we identify the unassigned cluster with the smallest value with respect to the nearest insertion criterion in route k , among the ones that fit within the available capacity. That is, let s denote the index of an unassigned cluster, and $k \in K$ such that $\widehat{d}_s \leq B^k$. Define now $q_s^k \in \operatorname{argmin}\{c_{p_{j-1}^k, q} + c_{q, p_j^k} - c_{p_{j-1}^k, p_j^k} : q \in T_s, j = 1, \dots, n_k\}$, which gives the best insertion of some vertex of cluster s within the route k , and let val_s^k denote the cost of such insertion. The value of the best insertion in route k is then given by $cost^k = \min\{val_s^k : T_s \in L, \widehat{d}_s \leq B^k\}$. Let q^k denote the vertex to be inserted, s^k its cluster and j^k the index of the node in route k for the insertion. Then, the insertion that is performed is the one with minimum cost, among all the routes. A scheme of the heuristic is given in Algorithm 2.

4.2 Local Search

Several neighborhoods have been considered for the Local Search. The first one is what we call a substitution neighborhood, where the node of one cluster is substituted by the other node of the cluster. The other two are a reinsertion neighborhood and a 3-exchange neighborhood.

Substitution Neighborhood: N1 This neighborhood contains all the solutions that can be obtained by substituting one vertex p such that $|T_{C_p}| = 2$ by the other vertex of T_{C_p} . Here we do not attempt to change neither the route nor the situation of the node within the route. If we interpret the move in terms of the routes of the original arc-routing problem, this move

Algorithm 2 Nearest insertion

Initialization

Order the clusters from furthest to closest to the depot.

Define the ordered list $L = \{T_1, T_2, \dots, T_s, \dots, T_r\}$.

Phase I

for $k = 1, \dots, K$ **do**

Let $T_{s(k)}$ denote the first cluster in list L .

Select the seed $s_k \in \operatorname{argmin} \{c_{p0}, c_{0p} : p \in T_{s(k)}\}$.

Identify I_k and $\widehat{I}_k = \{0, p_1^k, p_2^k, \dots, p_{n_k}^k\}$; $B^k = D - \sum_{j=1}^{n_k} d_{s_j}$.

$L = L \setminus \{T_i : i = C_q, q \in \widehat{I}_k\}$

end for

Phase II

while $L \neq \emptyset$ **do**

for $k = 1, \dots, K$ **do**

for $s = 1, \dots, r$ **do**

if $T_s \in L$ and $\widehat{d}_s \leq B^k$ **then**

Identify $q_s^k \in \operatorname{argmin} \{c_{p_{j-1}^k, q} + c_{q, p_j^k} - c_{p_{j-1}^k, p_j^k} : q \in T_s, j = 1, \dots, n_k\}$

Evaluate val_s^k .

end if

end for

$cost^k = \min\{val_s^k : T_s \in L, d_s \leq B^k\}$

Let $q_k \in T_{s^k}$ denote the vertex to be inserted immediately before vertex j^k .

end for

Let $k^* \in \operatorname{argmin}\{cost^k : k \in K\}$; $q^* = q_{k^*}$; $j^* = j^{k^*}$, $s^* = s^{k^*}$.

$L := L \setminus \{T_{s^*}\}$

$R^{k^*} := R^{k^*} \cup \{(p_{j^*-1}^{k^*}, q^*), (q^*, p_{j^*}^{k^*})\} \setminus \{(p_{j^*-1}^{k^*}, p_{j^*}^{k^*})\}$.

$B^{k^*} := B^{k^*} - d_{s^*}$.

end while

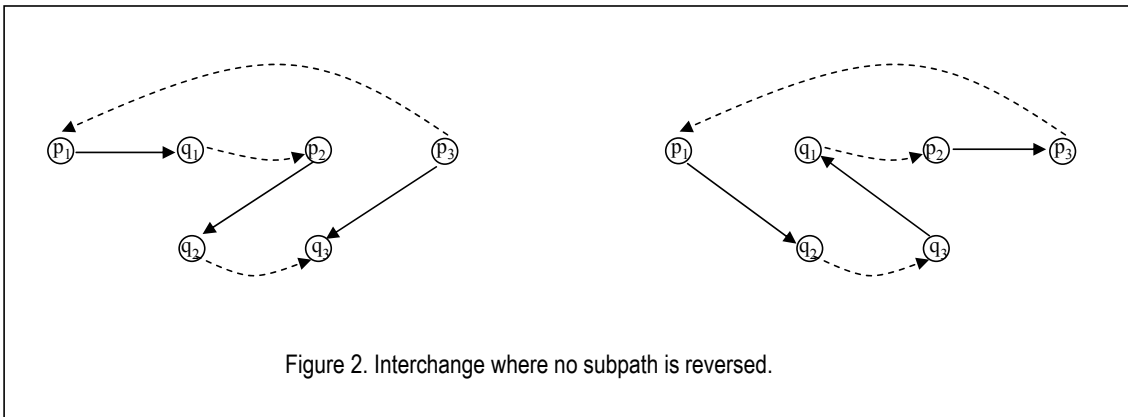
consists of changing the traversal direction of a required edge and rearranging accordingly the route. Let p denote the vertex that is going to be substituted by \bar{p} and let f and q denote its two neighbors in the route in the order they are visited. Then, the cost of the move that results when p is substituted by \bar{p} , is given by $-c_{fp} - c_{pq} + c_{f\bar{p}} + c_{\bar{p}q}$.

For exploring N1 we consider all vertices in turn and perform the first improving move.

Reinsertion Neighborhood: N2 This neighborhood contains all the solutions that can be obtained by removing one vertex and reinserting it at a different place, not necessarily within the same route. When the cluster of the vertex contains two vertices, the reinsertion also attempts to see which of the two vertices is better fitted for the new location. Let p denote the vertex that is removed, \bar{p} the vertex that is reinserted (either p or its opposite) and let f_1 and q_1 denote its current two neighbors in the route in the order they are visited. Then the cost of the move that results when p is reinserted between f_2 and q_2 , is given by $-c_{f_1p} - c_{pq_1} + c_{f_1q_1} - c_{f_2q_2} + c_{f_2\bar{p}} + c_{\bar{p}q_2}$. For exploring N2 we consider all vertices in turn and perform the first improving move.

3-Exchange Neighborhood: N3 We perform 3-interchanges as explained in [14]. Let a, b, c be three non consecutive arcs of a given route and let us consider all the routes that can be obtained by deleting a, b, c and permuting the three resulting subpaths. Note that in some of them the direction of some of the subpaths is reversed. Reversing a path is likely to change the value of the path and is much more expensive to evaluate than in the symmetric case. For this reason, we only consider moves in which none of the three subpaths is reversed (see Figure 2). The cost of the move that results when substituting (p_1, q_1) , (p_2, q_2) and (p_3, q_3) by (p_1, q_2) , (p_3, q_1) and (p_2, q_3) is given by $-c_{p_1q_1} - c_{p_2q_2} - c_{p_3q_3} + c_{p_1q_2} + c_{p_3q_1} + c_{p_2q_3}$. In our 3-Exchange moves, instead of performing the 3-exchanges within each single route, we consider the macro route that results from merging all of the $|K|$ routes into a single one (the depot is visited $|K|$ times), and we apply the 3-exchange procedure to the macro route. It is important to note that the solutions obtained with this interchange are always feasible, since the capacity constraints are taken into account. In particular, candidate interchanges that violate the capacity constraint of any of the routes are rejected.

For exploring N3 we consider all triplets of non adjacent arcs and perform the first improving move.



Algorithm 3 describes the local search that we apply.

Algorithm 3 Local Search

```
terminate ← false
while (not terminate) do
  1. while (some improving move is found in N1) do
    Explore N1
  enddo
  2. Explore N2
  3. if (some improving move has been found in N2) then go to to 1.
  4. Explore N3
  5. if (some improving move has been found in N3) then go to to 1.
  6. terminate ← true
end while
```

4.3 The ants heuristic

Ant systems (AS) [8] emulate the behavior of ant colonies in their paths from their nests towards food source: on the one hand, they deposit pheromone along the route, and on the other hand they usually follow the path where more pheromone has been previously deposited. Since better paths are followed by more ants they contain more pheromone. We have designed two ant heuristics that fit within the above paradigm and iteratively apply the following two phases: *i*) Solution building by means of a randomized constructive procedure followed by a local search, and *ii*) Pheromone updating to report back information for building of new solutions. The constructive procedure of the first heuristic, that we denote *Ant1*, is a randomized adaptation of the nearest neighbor heuristic as explained in Subsection 4.1. In the second heuristic, denoted *Ant2*, the constructive procedure follows the same template than Algorithm 2 and consists of a Phase I where the seeds for the initial routes are selected in a randomized fashion, followed by Phase II, which is a randomized adaptation of the nearest insertion heuristic of Subsection 4.1, where the actual routes are constructed.

As usual in AS's, the randomized selection criteria used by the two heuristics during their constructive procedures uses a probability distribution derived from the pheromone trail. In *Ant1*, the probability of connecting the last vertex of the partial route, say p , with vertex q (not belonging to any route) is:

$$\pi_{p,q}^1 = \frac{(\tau_{p,q})^\alpha (\eta_{p,q})^\beta}{\sum_{r \in CL} (\tau_{p,r})^\alpha (\eta_{p,r})^\beta} \quad (12)$$

where $\eta_{p,q} = \frac{1}{c_{p,q}}$ represents the heuristic information associated with the distance from the current vertex p and the candidate vertex q ; τ_{pq} is the value of the pheromone in arc (p, q) ; and CL denotes the index set for candidate vertices which are the ones in clusters not yet assigned with demand not greater than the current available capacity of the route. The initial pheromone values assign equal probability to all links. The parameters α and β determine the influence of the pheromone trails and the heuristic information, respectively.

In Phase I of *Ant2* the probability of selecting vertex p as seed for a route is given by:

$$\varphi_p = \frac{(\sigma_p)^\alpha (\eta_p)^\beta}{\sum_{q \in CL} [(\sigma_q)^\alpha (\eta_q)^\beta]}$$

where $\eta_q = c_{0q} + c_{q0}$ and σ_q denotes the value of the pheromone trail of the solutions of the previous iterations in which vertex q was the seed. CL denotes the set of vertices of clusters that do not belong to any route in the solution under construction. The parameters α and β have the same meaning as in (12).

In Phase II of *Ant2*, the probability of inserting vertex q between vertices p_j and p_{j+1} (consecutive in some route in the current solution under construction), is

$$\pi_{p_j q}^2 = \frac{[\max\{\tau_{p_j, q} + \tau_{q, p_{j+1}} - \tau_{p_j, p_{j+1}}, 0\}]^\alpha \left[\frac{1}{c_{p_j, q} + c_{q, p_{j+1}} - c_{p_j, p_{j+1}} + \epsilon} \right]^\beta}{\sum_r \sum_h [\max\{\tau_{p_h, r} + \tau_{r, p_{h+1}} - \tau_{p_h, p_{h+1}}, 0\}]^\alpha \left[\frac{1}{c_{p_h, r} + c_{r, p_{h+1}} - c_{p_h, p_{h+1}} + \epsilon} \right]^\beta}$$

Here, the pheromone trail of the arcs is given by the vector τ , and α and β have the same meaning as above. The small value parameter ϵ is included to prevent from dividing by zero.

Both in *Ant1* and *Ant2* the local search phase is the one described in Subsection 4.2 (see Algorithm 3). After the local search, the pheromone trails are updated. For the Phase I of *Ant2* the pheromone trail of each vertex is updated in two steps according to:

$$i) \sigma_p := (1 - \rho)\sigma_p$$

$$ii) \sigma_p := \sigma_p + \rho \times \Delta\sigma_p; \text{ where } \Delta\sigma_p = \frac{1}{\sum_{q \in \delta^+(p)} \left(c_{pq} \left[\sum_{k \in K} \hat{x}_{pqk} \right] \right)}$$

where \hat{x} denotes the incidence vector of the solution obtained at the end of the local search, and ρ is the pheromone evaporation rate. Similarly, in *Ant1* and in Phase II of *Ant2* the pheromone trail of each arc is updated in two steps according to:

$$i) \tau_{pq} := (1 - \rho)\tau_{pq}$$

$$ii) \tau_{pq} := \tau_{pq} + \rho \times \Delta\tau_{pq}; \text{ where } \Delta\tau_{pq} = \frac{1}{\sum_{(p, q) \in A} \left(c_{pq} \left[\sum_{k \in K} \hat{x}_{pqk} \right] \right)}$$

5 Computational experiments

In this section we report on the results obtained when applying the methodology described in the previous sections to the urban waste collection problem in Sant Boi de Llobregat, a municipality with over 80,000 inhabitants in the metropolitan area of Barcelona. We also compare the routes that we generate with the ones that were previously used for urban waste collection in the

municipality with the provided data. So far, the person in charge of the design of the routes used a combination of experience and intuition to do this task.

The road/street map of Sant Boi de Llobregat is depicted in Figure 3. This map results in a mixed graph with 777 vertices, 504 edges and 715 arcs. Required links are associated with roads where there exist collection containers. Collection containers are located on the sidewalks next to the roads in designated areas surrounded by a small border. Occasionally, in a double-direction road only one of the sidewalks is appropriate for locating the container (the other one might be too narrow, or may have trees, traffic signals, fences, etc. that do not make it possible to locate the container). Thus, it is possible that a road can be traversed in both directions (defining an edge of the graph), but can only be used (serviced) in one of the two directions for collection purposes (defining a required arc). In particular, there are 220 required edges and 459 arcs. Required links define a subgraph with 72 strongly connected components. The total length of required links is 55,993 m.



Figure 3: Map of Sant Boi de Llobregat

As explained in the previous sections, forbidden turns imply important constraints for defining feasible routes. These limitations are imposed not only due to traffic regulations but also, when the roads are narrow, because the size of the trucks make it impossible to make the turn. The map of the city contains 4150 turns, out of which 1316 are forbidden. The vast majority of forbidden turns (1008) are U-turns, whereas 308 forbidden turns correspond to turns that “connect” two different arcs/edges.

Two indicators (closely related between them) that are typically used to measure demand of waste collection are the number of inhabitants and the cubic meters per required link. In Sant Boi de Llobregat the average number of inhabitants per day per required link is 109.24 (being one and 939 the minimum and maximum values, respectively) with a total of 73,917. The remaining 6000 inhabitants in the census of Sant Boi live in residential non-urban areas and are not included in the urban waste collection program. The average number of cubic meters of waste per day per required link after compacting is 0,182 (the minimum and maximum are 0.002 and 1.565, respectively), with a total of 123,196.

The municipality owns 5 trucks of the same type, with a capacity of 25 cubic meters each. Broadly speaking this means that the capacity of each truck permits to collect the daily waste of 15,000 inhabitants. One characteristic of waste is its high compacting factor. This means that its volume in the container is considerably reduced when moved to the truck by just pressing the waste with an appropriate device of the truck. For the trucks in Sant Boi this factor is 1/6.

Two different versions of the ants heuristic described in Section 4 have been implemented. The two heuristics only differ in the construction phase. In the first one, denoted *NN*, the construction phase is the Nearest Neighbor heuristic of Algorithm 1, whereas the construction phase of the second heuristic, denoted *NI*, is the Nearest insertion heuristic of Algorithm 2. The values of the parameters in the ants heuristic are the following: $\alpha = 0.5$, $\beta = 0.5$ and $\rho = 0.9$. The total number of iterations has been fixed to 1000. All programs have been coded in C++ and the programs have been run on a Pentium IV, 3.2 Ghz and 512 Mb of RAM.

The results of the best solutions found with each heuristic are depicted in Table 1. The lengths of each of the five routes of each solution are under the columns “turn constraints”. Columns “inhabitants” and “ m^3 ”, respectively give the number of inhabitants served and the collected amount of waste after compacting, in m^3 , of each of the routes of the best solutions found. As can be seen, the *NI* heuristic gives the best results since the total length of the obtained routes is a %3.83 smaller that with the *NN* heuristic. The results of both heuristics allow to evaluate the increase in the actual costs of the routes due to forbidden turn constrains. In Table 1, for each heuristic, the column under “no turn constraint” gives the length of the routes when they service the same set of required links but without the turn constraints. When no forbidden turn constraints are imposed, the total length of the five routes obtained with the *NI* heuristic reduces from 93,561.00 m to 90,494.38 m. This supposes an increase of 3.38% on the total length of the routes due to forbidden turns. When the *NN* heuristic is used the total length of the routes reduces from 97,294.25 m to 92,903.51 m, resulting in a percent increase on the total length of the routes due to forbidden turns of 4.73. Thus, these percentages represent the marginal costs of forbidden turn constraints, that cannot be avoided.

The obtained results can also be appreciated in Figures 4 and 5 that, respectively, show graphically the behavior of the *NN* and *NI* heuristics both without forbidden turns and with forbidden turns. Both figures depict the value of the best solution known so far in terms of the number of iterations. It is easy to appreciate that both with and without turn constraints, the *NI* outperforms the *NN* heuristic, not only in that it provides a better quality solution at the end of the heuristic, but also in that the best known solution over all the iterations is attained at an earlier stage of the heuristic.

Table 2 gives the required cputimes. As can be seen, the required cputimes of both *NN* and

| | Nearest Neighbor construction phase | | | | Nearest Insertion construction phase | | | |
|---------|-------------------------------------|------------------|-------------|---------|--------------------------------------|------------------|-------------|---------|
| | no turn constraints | turn constraints | inhabitants | m^3 | no turn constraints | turn constraints | inhabitants | m^3 |
| Route 1 | 9825.975 | 19971.186 | 14808 | 24.680 | 9413.583 | 18834.308 | 14213 | 23.688 |
| Route 2 | 18723.989 | 21064.067 | 14464 | 24.107 | 17981.470 | 20003.835 | 14872 | 24.787 |
| Route 3 | 24933.970 | 21641.083 | 14956 | 24.927 | 24166.987 | 21149.188 | 14917 | 24.862 |
| Route 4 | 15404.073 | 10598.374 | 14776 | 24.627 | 15771.404 | 10384.054 | 14947 | 24.912 |
| Route 5 | 24015.505 | 24019.530 | 14913 | 24.855 | 23210.936 | 23189.850 | 14968 | 24.947 |
| Total | 92903.511 | 97294.241 | 73917 | 123.196 | 90494.379 | 93561.000 | 73917 | 123.196 |

Table 1: Results of the five routes

| | Nearest Neighbor | | Nearest Insertion | |
|-------------|---------------------|------------------|---------------------|------------------|
| | no turn constraints | turn constraints | no turn constraints | turn constraints |
| T_{best} | 16011.49 | 13030.75 | 377.86 | 14824.27 |
| T_{total} | 25095.18 | 26249.97 | 37397.90 | 37448.17 |

Table 2: Comparison of cputimes

NI are considerably smaller when forbidden turns constraints are posed. This is due to the fact that when such constraints are not required, there are many more possibilities to evaluate at each iteration. If we compare the Nearest Neighbor strategy with the Nearest Insertion strategy, since the computational burden per move is smaller in NN than in NI , the total time with the NN heuristic is smaller than that of the NI heuristic, both with and without turn constraints. However, the cputime needed to obtain the best solution is much smaller with NN than with NI when forbidden turns constraints are posed, whereas without such constraints the time of NN is slightly better than that of NI .

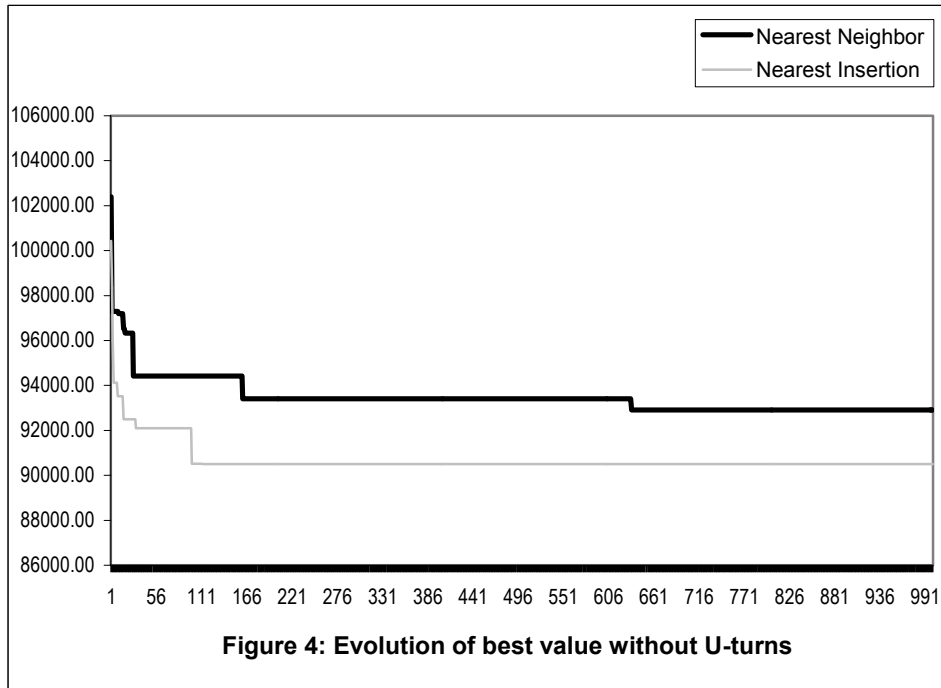


Figure 4: Evolution of best value without U-turns

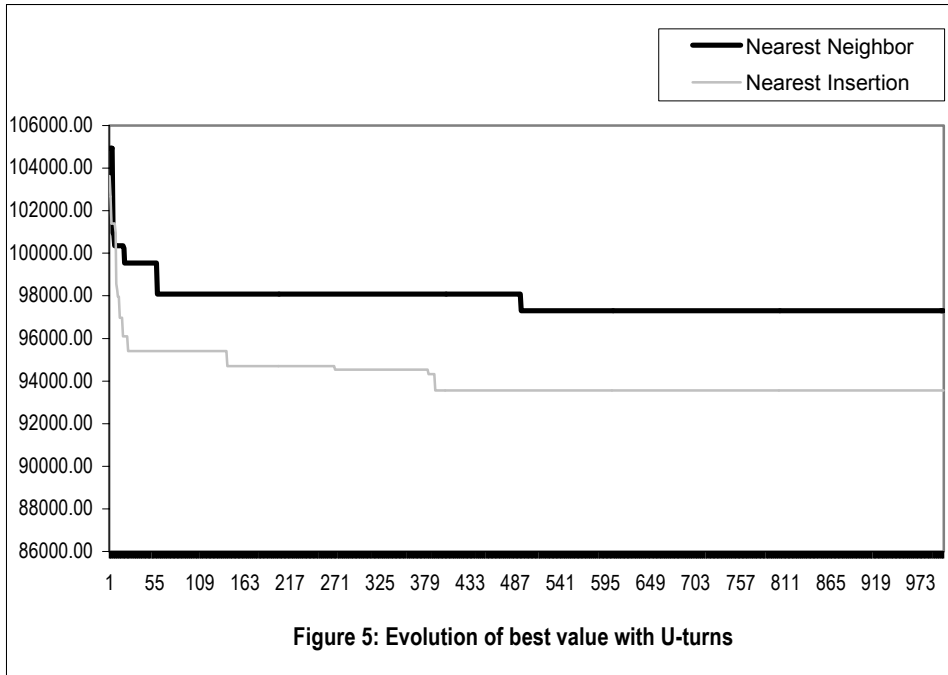


Figure 5: Evolution of best value with U-turns

The routes that were used so far for urban waste collection in Sant Boi had an average length of 30,000 m each, with a total length of approximately 150,000 m. Thus, the routes that we obtain suppose a reduction on the total length of 35% and 37% for *NN* and *NI*, respectively. Note that that these reduction percentages amount to more than 17,000 Km per year.

One factor that allows to evaluate the quality of the obtained routes is the total length of non-required links in the obtained solutions. For the *NN* heuristic this value is $97,294 - 55,993 = 41,301$ m, which represents a 42.44% of the total length of the routes, whereas for the *NI* heuristic this value is $93,561 - 55,993 = 37,568$ m, which represents a 40.15% of the total length of the routes. These percentages are indeed much smaller than the 62.67% corresponding to the $150,000 - 55,993 = 94,007$ m of the original routes. Note that these figures represent considerable reductions in the total length of the non- required links with respect to the original routes of 56.04% and 60.04% for *NN* and *NI*, respectively.

Given that the fleet that works in waste collection is fixed, the economic impact derived from this reduction must be associated to operating costs. Two main sources of operating costs are reduced: the first one is related to fuel consumption, derived from a yearly reduction of 22,000 km to collect waste. The second one is the availability of additional time from truck workers. Each truck is equipped with a driver and two operators. The additional time of the personnel can be used for other tasks of the municipality (as, for instance, maintenance and sweeping of streets).

We must, however, point out that there are additional types of benefits, difficult to quantify currently, derived from this considerable reduction on the length of the routes. The first and most important one is the reduction in the night acoustic contamination. The reduction in the length of the routes indicates that there will also be a considerable reduction on the total time

that the trucks will be collecting waste and, therefore, fewer inhabitants that will be disturbed by the noise of the trucks.

6 Conclusions

In this paper we have presented the methodology that we have applied for the solution of an urban waste collection problem in Sant Boi de Llobregat within the metropolitan area of Barcelona (Spain). The nature of the problem is that of an arc routing one, but by means of a transformation it has been modeled in terms of an arc routing problem with additional constraints for forbidding some turns. Solutions to the problem have been obtained with heuristics based on ant colonies and different neighborhood strategies have been compared.

As reported in the Computational Experiments Section The obtained solutions improve considerably the ones that have been used so far both in terms of the operating cost and of acoustic contamination.

The applicability of the proposed procedures to real-life circumstances is subject to their integration within support systems to aid planners in decision-making. It should be possible to modify routes constantly to take into account changes on urban circulation. SIRUS [1], was the software application designed in this case. The system provides a graphical tool to simplify different tasks required by the managers as the location of collection areas, allocation of refuse bins, routing, etc.

Acknowledgements The research of the first and the third author has been partially supported through the Spanish CICYT grant BEC2003-03809. The research of the second author has been partially supported through the SPANISH MEC grant TIC2003-05982-C05-04. These supports are gratefully acknowledged.

References

- [1] J. Bautista. Proyecto integral de gestión de de residuos urbanos en el municipio de Sant Boi de Llobregat (in Spanish). CPDA, Barcelona, 2001.
- [2] J.M. Belenguer and E. Benavent. The capacitated arc routing problem: valid inequalities and facets. *Computational Optimization and Applications*, 10:165–187, 1998.
- [3] L. Bodin and S.A. Kursh. A computer-assisted system for the routing and scheduling of street sweepers. *Operations Research*, 26(4):525–537, 1978.
- [4] L. Bodin and S.A. Kursh. A detailed description of a computer system for the routing and scheduling of street sweepers. *Computers and Operations Research*, 6:181–198, 1979.
- [5] A. Boroujerdi and J. Uhlmann. An efficient algorithm for computing least cost paths with turn constraints. *Information Processing Letters*, 67:317–321, 1998.

- [6] A. Corberán, R. Martí, Eulalia Martínez, and D. Soler. The rural postman problem on mixed graphs with turn penalties. *Computers and Operations Research*, 29:887–903, 2002.
- [7] M. Desrochers and G. Laporte. Improvements and extensions to the Miller-Tucker-Zemlin subtour elimination constraints. *Operations Research Letters*, 10:27–36, 1991.
- [8] M. Dorigo, V. Maniezzo, and A. Coloni. The ant system: Optimization by ant colony of cooperating agents. *IEEE Transactions on Systems, Man and Cybernetics - Part B*, 26:29–41, 1996.
- [9] M. Dorigo and T. Stützle. *Ant Colony Optimization*. MIT Press, 2004.
- [10] M. Dror, editor. *Arc Routing: Theory, Solutions and Applications*. Kluwer, Boston, 2000.
- [11] R.W. Floyd. Algorithm 97: Shortest path. *Communications of the ACM*, 5(6):345, 1962.
- [12] G. Ghiani. and G. Improta. An efficient transformation of the generalized vehicle routing problem. *European Journal of Operational Research*, 122:11–17, 2000.
- [13] B.L Golden and R.T. Wong. Capacitated arc routing problems. *Networks*, 11:305–315, 1981.
- [14] D. Johnson, G. Gutin, L.A. McGeoch, A. Yeo, W. Zhang, and A. Zverovitch. Experimental analysis of heuristics of the atsp. In G. Gutin and A. Punen, editors, *The Traveling Salesman Problem and its Variations*, pages 445–487. Kluwer Academic Publishers, 2002.
- [15] G. Laporte. Location-routing problems. In B. L. Golden and A. A. Assad, editors, *Vehicle routing: Methods and Studies*, pages 163–197. North-Holland, Amsterdam, 1988.
- [16] G. Laporte. Modeling and solving several classes of arc routing problems as traveling salesman problems. *Computers and Operations Research*, 24(11):1057–1061, 1997.
- [17] G. Laporte and Y. Nobert. A branch and bound algorithm for the capacitated vehicle routing problem. *Operations Research Spektrum*, 5:77–85, 1983.
- [18] A.N. Letchford. *Polyhedral results for some constrained arc-routing problems*. PhD thesis, Department of Management Science, Lancaster University, 1997.
- [19] C. E. Miller, A. W. Tucker, and R. A. Zemlin. Integer programming formulations and traveling salesman problems. *Journal of the Association for Computing Machinery*, 7:326–329, 1960.
- [20] E. Noon and J.C. Bean. An efficient transformation of the generalized traveling salesman problem. *INFOR*, 31:39–44, 1993.
- [21] J. Pereira. Modelización y resolución de problemas de diseño de sistemas de recogida de residuos urbanos (in Spanish). Ph. D Thesis, Universitat Politècnica de Catalunya, 2004.
- [22] S. Roy and Rousseau J.M. The capacitated canadian postman problem. *INFOR*, 27(1):58–73, 1989.
- [23] S. Warshall. A theorem on boolean matrices. *Journal of the ACM*, 9(1):11–12, 1962.