THE CAPACITATED GENERAL ROUTING PROBLEM ON MIXED GRAPHS

Julio César Angel Gutiérrez, Departamento de Informática y Sistemas, Universidad Eafit, Medellín, Colombia
David Soler y Antonio Hervás, Departamento de Matemática Aplicada, Universidad Politécnica de Valencia, España

ABSTRACT
The Capacitated General Routing Problem (CGRP) on mixed graphs is one of the more complex combinatorial optimization problems on vehicle routing. It consists basically of finding a set of routes on a mixed graph, beginning and ending at the same vertex (depot), with minimum total cost, satisfying demands located at links and vertices and with a capacity restriction on the demand satisfied by each route. Several particular cases of this problem have deeply been studied in the operational research literature but, in order to solve the general problem, we only have found a heuristic procedure based on route-first-partition-next. We present here a new heuristic that seems to work much better according to our computational results.

Key words: Capacitated vehicle routing, mixed graphs, heuristic.

RESUMEN
El Problema General de Rutas con Capacidades (CGRP) sobre grafos mixtos es uno de los más complejos problemas de optimización combinatoria sobre rutas de vehículos. Básicamente consiste en buscar un conjunto de rutas sobre un grafo mixto, que empiezan y terminan en el mismo vértice (depósito), con coste total mínimo, que satisfacen demandas localizadas en vértices y enlaces y con una restricción de capacidad sobre la demanda satisfecha por cada ruta. Varios casos particulares de este problema han sido estudiados con profundidad en investigación operativa pero, de cara a resolver el problema general, sólo hemos encontrado un algoritmo heurístico basado en ruta-primero-partición-después. Presentamos aquí un nuevo heurístico que funciona bastante bien de acuerdo con nuestros resultados computacionales.

Palabras clave: Rutas de vehículos con capacidades, grafos mixtos, heurístico.

AMS**:** 90B06, 90C35

1. INTRODUCTION

The *Capacitated General Routing Problem on Mixed Graphs* (CGRP-m) can be defined as follows:

"Let $G = (V, E \cup A)$ be a strongly connected mixed graph where: each link $(i,j) \in E \cup A$ has associated a cost $c_{ij} \geq 0$, vertex 1 represents a depot where there are k vehicles with identical capacity W, it exists a set $V_R \subseteq V$ such that each vertex $i \in V_R$ has associated a positive demand $q_i \leq W$, it exists a set $A_R \subseteq A$ such that each arc $(i,j) \in A_R$ has associated a positive demand $q_{ij} \leq W$, it exists a set $E_R \subseteq E$ such that each edge $(i,j) \in E_R$ has associated a positive demand $q_{ij} \leq W$ and the sum of all demands is not greater than kW.

Find k tours in G such that each tour passes through the depot, the demands at V_R, A_R and E_R are satisfied, each one by exactly one tour, the total load of each tour does not exceed vehicle capacity W and the sum of the cost of the k tours is minimum."

Note that in order to ensure feasibility, it is assumed without loss of generality that k is equal to the minimum number of vehicles necessary to attend all demands.

The CGRP-m generalizes many vehicle routing problems that have been studied in the last forty years and for which hundreds of papers have been written, either to give exact or heuristic procedures for their resolution or to give lower bounds. For example:

- If $A = \emptyset = V_R$ we have the Capacitated Arc Routing Problem (CARP).
- If $A = \emptyset = E_R$ we have the Capacitated Vehicle Routing Problem (CVRP).
- If $E = \emptyset = A_R$ we have the Asymmetric Capacitated Arc Routing Problem (ACVRP).

- If k = 1 we have the General Routing Problem (GRP).

- If k = 1 and $V_R = \emptyset$ we have the Mixed Rural Postman Problem (MRPP).

- etc.

With regard to the particular cases with k > 1, it is assumed that problems of realistic size must be tackled with heuristic approaches, so the majority of the studies focuses on effective heuristic procedures that either improve previous approaches or solve the problem for the first time. As the best heuristics procedures for particular cases of the CGRP-m, we must cite among others, the works of Hertz, Laporte and Mittaz (2000) for the CARP, Taillard (1993) for the CVRP, Vigo (1996) for the ACVRP, López (1998) for the GRP, and the paper of Corberán, Martí and Romero (2000) for the MRPP.

The reasons of the extensive study of these problems are due to the fact that there are many real situations in which such a set of routes is required: collection or delivery of goods, garbage collection, mail delivery, snow removal, school bus routing, pipelines inspections, etc.

Despite the fact that in a big city, its street network must be modelized as a mixed graph and, for instance, in order to collect garbage, several vehicles are needed, so the CGRP-m seems to be appropriate to solve optimally this real problem, the CGRP-m, in its general definition, has hardly been considered in the literature. In fact, as far as we know, the only paper that gives a procedure (approximate procedure) to solve this problem is by Pandit and Muralidharan (1995). This procedure follows the route-first-partition-next methodology. Basically it consists of two phases:

In the first phase they obtain a heuristic solution to the GRP in the graph *G,* that is, they suppose no capacity restrictions so only one vehicle is required to attend all the demands. Then, a Giant Tour that satisfies all the demands is obtained.

In the second phase this Giant Tour is broken into subtours in the following way: start the tour at the depot, follow the tour, break the tour when the capacity constraint is violated and return to the depot (first route). Go from the depot to the previous break point, follow the tour until the capacity constraint is violated again, return to the depot (second route), and so far.

In this paper we present an alternative heuristic procedure to the one given by Pandit and Muralidharan. Our procedure is close to the partition-first-route-next methodology, as it happens in many studies on particular cases of this problem, see for example Chapleau et al. (1984), Benavent **et al**. (1990), Pearn (1991) or Vigo (1996). This methodology seems more appropriate in a CGRP-m constructive heuristic, in fact, in only one instance from those in which we have compared both heuristics, the algorithm of Pandit and Muralidharan provided better result than ours.

The remainder of this paper is organized as follows. In Section 2 we present our heuristic for the CGRP-m. In Section 3, in order to clarify how they work, we apply this heuristic and the one of Pandit and Muralidharan to an instance. Finally, in Section 4 we show our computational results on a set of 28 instances with up to 50 vertices, 98 links and 4 vehicles.

2. HEURISTIC ALGORITHM FOR THE CGRP-m

The heuristic we present here constructs the routes one after the other. Firstly, for each required arc, edge or isolated vertex it finds a minimum cost initial route joining the required element with the depot. To start a vehicle route it selects the remaining largest initial route and it inserts, with minimum cost increment of the route, demands in this route according with the following preferences. First the nearest required elements to the one that defines the initial route, without violate the capacity constraint. Then it inserts demands corresponding to required elements traversed by the route but with demands not inserted yet, and finally, if the loaded demand is not greater than 0.9W, it tries to insert demands corresponding to required elements that are very close to the route. Exceptionally, our procedure may provide more than k routes, as it occurs in all the heuristics procedures to solve capacitated routing problems.

Once we have the solution given by our heuristic, as it implies that we have assigned required elements to each vehicle, we may consider that we have k (or more) General Routing Problems on mixed graphs, so in order to improve each one of the obtained routes, we apply a GRP heuristic to the set of demands satisfied

by each vehicle. This heuristic, based on Monte Carlo techniques is due to López (1998) and it is the more recently one that we have found for this problem.

In what follows in this section, each step of our heuristic is described in more detail.

Notation used by the heuristic:

- W: Vehicle capacity.
- Q: Total demand in the graph.
- CA: Accumulated cost of the assigned routes.
- DA: Accumulated demand of the assigned routes.
- Required element: Each required isolated vertex (non incident with required arcs or edges) and each required arc or edge together with its two incident vertices. We suppose that the sum of the (at most three) demands of a required element is not greater than W. A required element is denoted by (i,j). If i = j we understand that this element is a (isolated) vertex (we suppose that G has not loops).
- B: the set of required elements not inserted yet in any route at a given moment.
- R(i, j): A route containing the depot and the required element (i, j).
- C(i, j): Total cost of the route R(i, j).
- D(i, j): Total demand of the route R(i, j).
- N: Number of assigned routes.
- SP(u,v): Shortest path from vertex u to vertex v.
- CSP(u,v): Cost of SP(u,v) (CSP(u,u)=0).
- q_0: Minimal demand corresponding to the required vertices (all), arcs and edges not assigned yet to a route at a given moment.
- DF: Demand needed at a given moment to complete the maximal capacity W of a route.
- M_0: Statistical median of the costs of all edges and arcs of the graph.
- KI(u,v): Insertion path from vertex u to vertex v. If $u \neq v$, KI(u,v) is a segment of a route between vertices corresponding to two consecutive required elements in the route, including the depot, both inserted in that route, and such that this segment does not contain any other required element inserted in the route. Necessarily KI(u,v) = SP(u,v). If u = v, KI(u,u) = {u}.

HEURISTIC

Step 0: (Previous calculations and initialization)

B = The set of all required elements in G (as they where defined above).

- Find the initial route for each $(i,j) \in B$ in the following way:

  - For each edge $(i,j) \in B$ its initial route is the one of least cost between $R(i,j) = SP(1,i) \cup (i,j) \cup SP(j,1)$ and $R(j,i) = SP(1,j) \cup (j, i) \cup SP(i,1)$.
  - For each arc $(i,j) \in B$ its initial route is $R(i,j) = SP(1,i) \cup (i,j) \cup SP(j,1)$.
  - For each isolated vertex $(i,i) \in B$ (i = (i,i)) its initial route is $R(i,i) = SP(1,i) \cup SP(i,1)$.

- For each pair (u_1,u_2), (v_1,v_2) of required elements of the graph, including the depot (remember that a vertex u is denoted by (u,u)), calculate the distance between them, defined as $\min\{CSP(u_1,v_1), CSP(u_1,v_2), CSP(u_2,v_1), CSP(u_2,v_2), CSP(v_1,u_1), CSP(v_1,u_2), CSP(v_2,u_1), CSP(v_2, u_2)\}$.

- Calculate M_0.

- Do DA = 0, CA = 0 and N = 0.

Step 1: Do N = N+1, I = 0.

Select the initial route R(i,j) with largest cost corresponding to a required element (i,j) $\in$ B. Let R(i,j) = SP(1,i) $\cup$ (i,j) $\cup$ SP(j,1) be this route, consider the insertion paths: KI(1,i), KI(j,1), KI(1,1), KI(i,i) and KI(j,j), being (i,j) the first inserted required element in route N.

Do $D(i, j) = q_i + q_{ij} + q_j$ and DF = W – D(i,j).

Step 2: Put in order the demands of the remainder elements of B according to the instance of the required element to (i,j) in increasing order of distance. Given this order, calculate the maximal accumulated demand F_s (s implies that F_s corresponds to the first s elements) such that $F_s \leq DF$.

Note that the demand corresponding to a vertex belonging to several required elements is assigned to the first of these required element according to the given order.

Let A_s be the set containing the first s required elements. If $A_s=\emptyset$ (s=0) go to Step 5.

Do $D(i,j) = D(i,j) + F_s$ and DF = W – D(i,j).

Step 3: For each element of A_s not inserted yet in the route N, calculate the cost increment due to its insertion in each one of the new insertion paths $KI(u_i,v_i)$. This cost is defined in the following way:

- For each non inserted edge $(u,v) \in A_s$, $\min\{CSP(u_i,u) + c_{uv}+CSP(v,v_i) – CSP(u_i, v_i), CSP(u_i,v) + c_{uv} + CSP(u, v_i) – CSP(u_i,v_i)\}$.
- For each non inserted arc $(u,v) \in A_s$, $CSP(u_i,u) + c_{uv} + CSP(v,v_i) – CSP(u_i, v_i)$.
- For each non inserted isolated vertex $u \in A_s$, $CSP(u_i,u) + CSP(u,v_i) – CSP(u_i,v_i)$.

Step 4: Select the insertion path $KI(u_i,v_i)$ and the required element $(u,v)\in A_s$ for which the minimum cost increment is reached in Step 3.

Insert this required element in the route N by replacing $KI(u_i,v_i)$ with $SP(u_i, u)\cup(u,v)\cup SP(v,v_i)$. R(i,j) is then updated:

$$R(i,j) = \{R(i,j)–KI(u_i,v_i)\}\cup SP(u_i,u)\cup(u,v)\cup SP(v, v_i)$$

Remove $KI(u_i,v_i)$ from the list of insertion paths and add to this list the new insertion paths: $KI(u_i,u)$, $KI(v,v_i)$, KI(u,u) and KI(v,v) (note that occasionally $u_i = v_i$ and/or u = v).

Do I = I + 1. If I < s go to Step 3.

If $DF < q_0$ go to Step 10.

Step 5: Consider the demands not inserted yet in any route and located at vertices and links traversed by the route R(i,j). Put in order these demands according to their distance to the depot in the route, going to or coming from the depot, in increasing order of distance. To do this, consider that a demand corresponding to an arc or edge is located in the center of the link. Given this order, calculate the maximal accumulated demand F_m (m implies that F_m corresponds to the first m elements) such that $F_m \leq DF$ and let F_n be the total non inserted demand located in R(i,j).

If $F_m = 0$ (m = 0) go to Step 6.

$$D(i,j) = D(i,j) + F_m$$

$$DF = DF- F_m.$$

If $DF< q_0$ go to Step 10.

If $F_m = F_n$ go to Step 7.

Step 6: Put in increasing order the remaining demands not inserted yet in any route and located at vertices and links traversed by the route R(i,j). Given this order, calculate the maximal accumulated demand F_p such that $F_p \leq DF$.

If $F_p = 0$ go to Step 7.

Do $D(i,j) = D(i,j) + F_p$ and $DF = DF - F_p$.

If $DF < q_0$ or $DF \leq 0.10 \cdot W$ go to Step 10.

Step 7: From among the required elements $(u,v) \in B$ not present in R(i,j), not labeled and such that at least one of its demands q_u, q_{uv}, q_v, is different from zero and little or equal than DF, select the one that minimizes $\min\{CSP(u_i,u)+CSP(v,u_i)$: u_i is a vertex belonging to a required element inserted in R(i,j), including the depot$\}$ if this minimum is little or equal than $2M_0$.

If such required element (u,v) does not exist go to Step 10.

Step 8: Let $d_{uv} = q_u + q_{uv} + q_v$ be the demand of (u,v).

If $d_{uv} \leq DF$ do $d_0 = d_{uv}$ and go to Step 9.

From among the six elements q_u, q_v, q_{uv}, $q_u + q_v$, $q_u + q_{uv}$, $q_v + q_{uv}$, select the largest one little or equal than DF. Let d_0 be this element.

Step 9: Do $D(i,j) = D(i,j) + d_0$, $DF = DF - d_0$ and:

- $R(i,j) = R(i,j) \cup SP(u_i,u) \cup (u,v) \cup SP(v,u_i)$ if d_0 contains q_{uv}.
- $R(i,j) = R(i,j) \cup SP(u_i,u) \cup SP(u,u_i)$ if $d_0 = q_u$.
- $R(i,j) = R(i,j) \cup SP(u_i,v) \cup SP(v,u_i)$ if $d_0 = q_v$.
- $R(i,j) = R(i,j) \cup SP(u_i,u) \cup (u,v) \cup SP(v,u_i)$ if $d_0 = q_u + q_v$ and $CSP(u_i,u) + c_{uv} + CSP(v,u_i) \leq$ $\leq CSP(u_i,u) + CSP(u,u_i) + CSP(u_i,v) + CSP(v,u_i)$.
- $R(i,j) = R(i,j) \cup SP(u_i,u) \cup SP(u,u_i) \cup SP(u_i,v) \cup SP(v,u_i)$ otherwise.

If $DF > 0.10 \cdot W$ and $DF \geq q_0$, label (u,v) and go to Step 7.

Step 10: Consider route R(i, j) as a definitive route.

Do $DA = DA + D(i,j)$ and $CA = CA + C(i,j)$.

If DA = Q **STOP**.

Update B, update the demands of its elements and for each required vertex become isolated, find its initial route and calculate its distances to the remaining required elements. Remove the label to the required elements labeled in Step 9.

Go to Step 1.

The above heuristic has polynomial complexity. In fact, the steps with largest number of operations are: Step 0 with complexity $O(|V|^3)$ (in the worst case, the set of its operations is dominated by the computation of the shortest path between each ordered pair of vertices in G, which are stored in such a way that we do not need to compute them in the following steps); Step 2 with complexity $O(r^2)$ being $r = |A_R \cup E_R| + |V_R|$ (in the worst case, the set of its operations is dominated by the ordering of the demands of all required elements in G); Step 3 which, after all the times that the heuristic passes through it, has complexity $O(r \cdot |V|^2)$ ($r \cdot |V|^2$ is an upper bound on the number of combinations between a required element and an insertion path) and Step 7 for which it is easy to see that its complexity is $O(r \cdot |V|)$. Then, we may conclude that our heuristic has polynomial complexity upper-bounded by $O(s^3)$ being $s = \max\{|V|,r\}$ (note that $k \leq r$).

3. EXAMPLE OF APPLICATION OF THE HEURISTIC

In this section we apply our heuristic for the CGRP-m to the mixed graph given in Figure 1, with 20 vertices, 27 edges, 8 arcs, 12 required vertices (the shady vertices), one of them isolated, 4 required edges (in bold

lines) and 3 required arcs (in bold lines). Numbers in brackets represent demands and numbers without brackets represent link costs. The total demand of the graph is Q = 110 and the vehicle capacity is W = 40, then, the minimum number of vehicles to satisfy the demands is k = 4. The statistical median of the link costs is M_0 = 5 and the minimum individual demand is q_0 = 2.

Once we have the routes given by our heuristic, in order to improve their cost, we will apply to each one the GRP heuristic given by López (1998) which, in accordance with computational results, produces optimal or nearly optimal solutions in small/medium-sized instances.

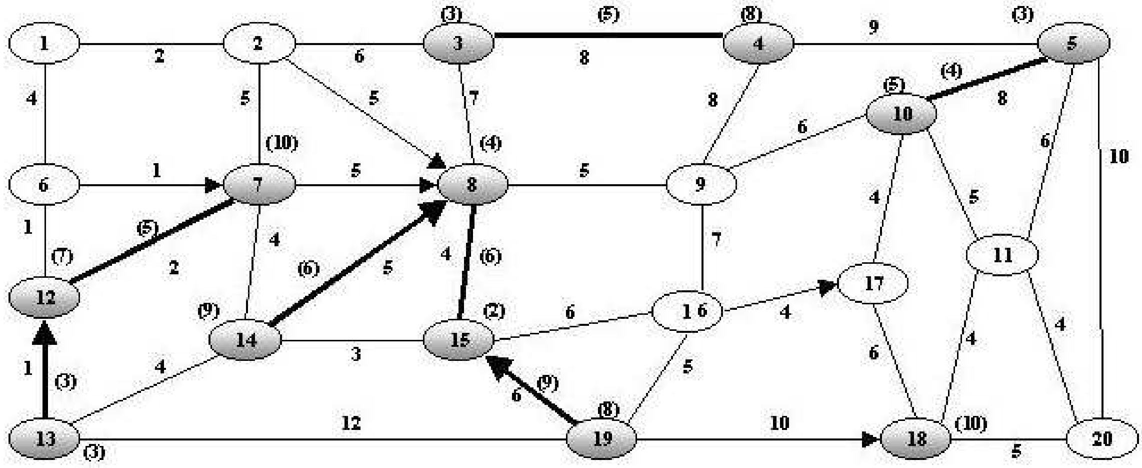

Figure 1. Mixed graph where we are going to apply our heuristic.

Finally, to compare both heuristics, we apply to the same graph the heuristic given by Pandit and Muralidharan, obtaining its Giant Tour with the heuristic of López and, in order to guaranty the same conditions to both heuristics, we also apply the GRP heuristic to each one of the routes given by the heuristic of Pandit and Muralidharan.

To be brief, we will only show how our heuristic constructs the first route step by step:

Step 1 selects as largest initial route R(18,18) = SP(1,18) $\cup$ SP(18,1), with C(0,18) = 62.

R(18,18) = (1,2)(2,8)(8,9)(9,10)(10,11)(11,18)(18,11)(11,10)(10,9)(9,8)(8,3)(3,2)(2,1). N = 1, I = 0, the new insertion paths are KI(1,18), KI(18,1) and KI(18,18). The first required element inserted in the route is vertex 18, D(18,18)=q_{18}=10 and DF =40–10=30.

Is **Step 2**, the required elements ordered according to their distance to vertex 18 are {(10,5), (19,15), (15,8), (3,4), (14,8), (13,12), (12,7)}, with respective demands {12,19,10,16,15,13,15}. The accumulated demands are then {12,31,41,57,72,85,100}.

The maximal accumulated demand little or equal than DF is F_1 = 12 (s = 1).

$$D(18,18) = 10 + 12 = 22.$$

$$A_s = \{(10,5)\}$$

From **Steps 3 and 4** we obtain that the insertion path KI(18,1) is substituted for SP(18,10) $\cup$ (10,5) $\cup$ SP(5,1), I = I + 1 = 1 = s, the new insertion paths are KI(18,10), KI(5,1), KI(10,10) and KI(5,5).

R(18,18) is updated:

$$R(18,18) = (1,2)(2,8)(8,9)(9,10)(10,11)(11,18)(18,11)(11,10)(10,5)(5,4)(4,3)(3,2)(2,1)$$

$$DF = 18 > q_0 = 2.$$

In **Step 5** the demands present in R(18,18) but not inserted yet are $\{q_4, q_{43}, q_3, q_8\}$, ordered according to the largest distance to the depot (in the route). The accumulated demands are then {8,13,16,20}, the maximal accumulated demand little or equal than DF is F_3 = 16, $F_n = F_4$ = 20 ($F_3 < F_4$), D(18,18) = 22+ 16 = 38 and DF = 2 $\geq q_0$.

In **Step 6**, the only demand to be considered is q_8 = 4, greater than DF, so it is not inserted.

In **Step 7** we have that there are nonrequired elements complying with the conditions given in that step so the heuristic goes to **Step 10**, being the first definitive route R(18,18 = (1,2)(2,8)(8,9)(9,10)(10,11) (11,18)(18,11)(11,10)(10,5)(5,4)(4,3)(3,2)(2,1), with D(18,18) = 38, C(18,18) = 69, DA = 0 + 38 = 38, CA = 0 + 69 = 69 and Q – DA > 0, so the heuristic goes to **Step 1** again.

The second time that the heuristic arrives at Step 10, it obtains the second definitive route R(19,15) = (1,2)(2,8)(8,15)(15,16)(16,19)(19,15)(15,14)(14,13)(13,12)(12,6)(6,1), with D(19,15) = 38 and C(19,15) = 41, and the third time it arrives at Step 10 it obtains the last definitive route R(14,8) = (1,6)(6,7)(7,14)(14,8) (8,15)(15,14)(14,13)(13,12)(12,7)(7,2)(2,1), with D(14,8) = 34 and C(14,8) = 35. The total cost of the three routes is CA = 145. In Figure 2 we show the three routes in the graph (they are represent by dotted arcs).

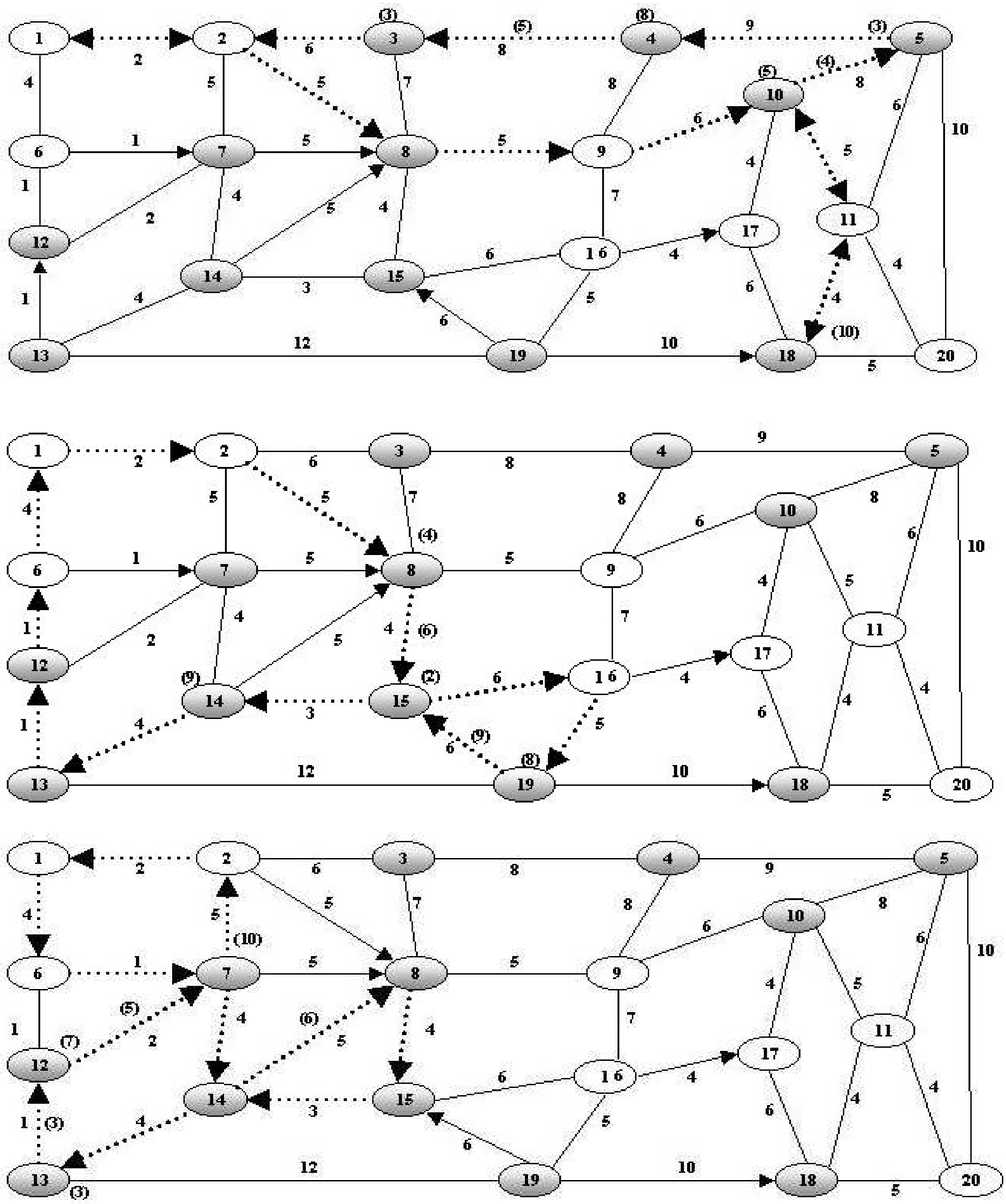

Figure 2. Routes given by our heuristic.

The GRP heuristic only improves the third route, with a saving of 2 units of cost. The new third route is given in Figure 3. It is R(14,8) = (1,6)(6,12)(12,7)(7,14)(14,8)(8,15)(15,14) (14,13)(13,12)(12,6)(6,1).

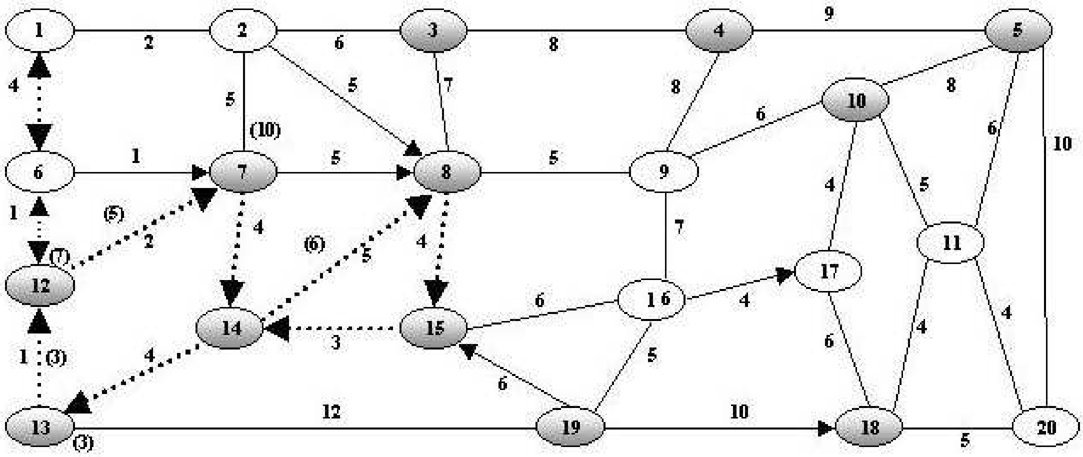

Figure 3. Improvement of route 3 by means of the GRP heuristic.

With regard to the heuristic of Pandit and Muralidharan, Figure 4 shows the Giant Tour obtained by the GRP heuristic, whereas Figure 5 shows the three routes obtained by this method with cost 77, 51 and 28 respectively and with load 38, 38 and 34 respectively. Its total cost is then 156. They are:

R_1 = (1,2)(2,3)(3,4)(4,5)(5,10)(10,11)(11,18)(18,11)(11,10)(10,9)(9,8)(8,3)(3,2)(2,1).

R_2 = (1,2)(2,8)(8,15)(15,16)(16,19)(19,15)(15,14)(14,8)(8,3)(3,2)(2,1).

R_3 = (1,2)(2,8)(8,15)(15,14)(14,13)(13,12)(12,7)(7,12)(12,6)(6,1).

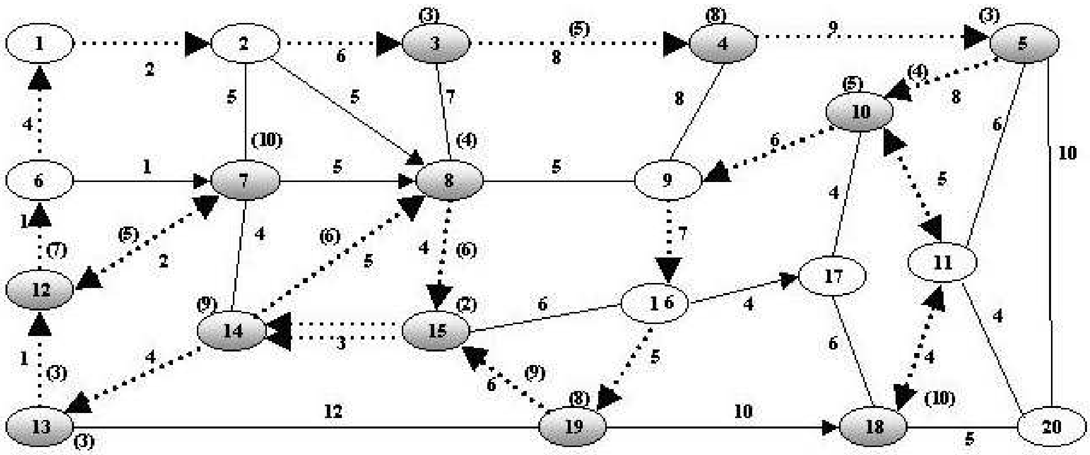

Figure 4. Giant Tour in G obtained by the GRP heuristic.

After applying the GRP heuristic to each one of the three routes obtained by the heuristic of Pandit and Muralidharan, we have that this heuristic improves routes 1 and 2 with new costs 69 and 48 respectively. The total new cost is then 145 and it represents a saving of 11 units. The first new route is exactly the first one given by our heuristic, so we do not reproduce it. The second new route (see Figure 6) is (1,6)(6,7)(7,14)(14,8)(8,15)(15,16) (16,19)(19,15)(15,14)(14,13) (13,12)(12,6)(6,1).

Then we have that without applying the GRP heuristic to any route obtained by both heuristics, the solution given by Pandit and Muralidharan represents a cost increment of 7.58 % with regard to the solution produced by our heuristic whereas after applying the GRP heuristic to all routes obtained by both heuristics, the solution given by Pandit and Muralidharan represents a cost increment of 1.39 % with regard to the solution produced by our heuristic.

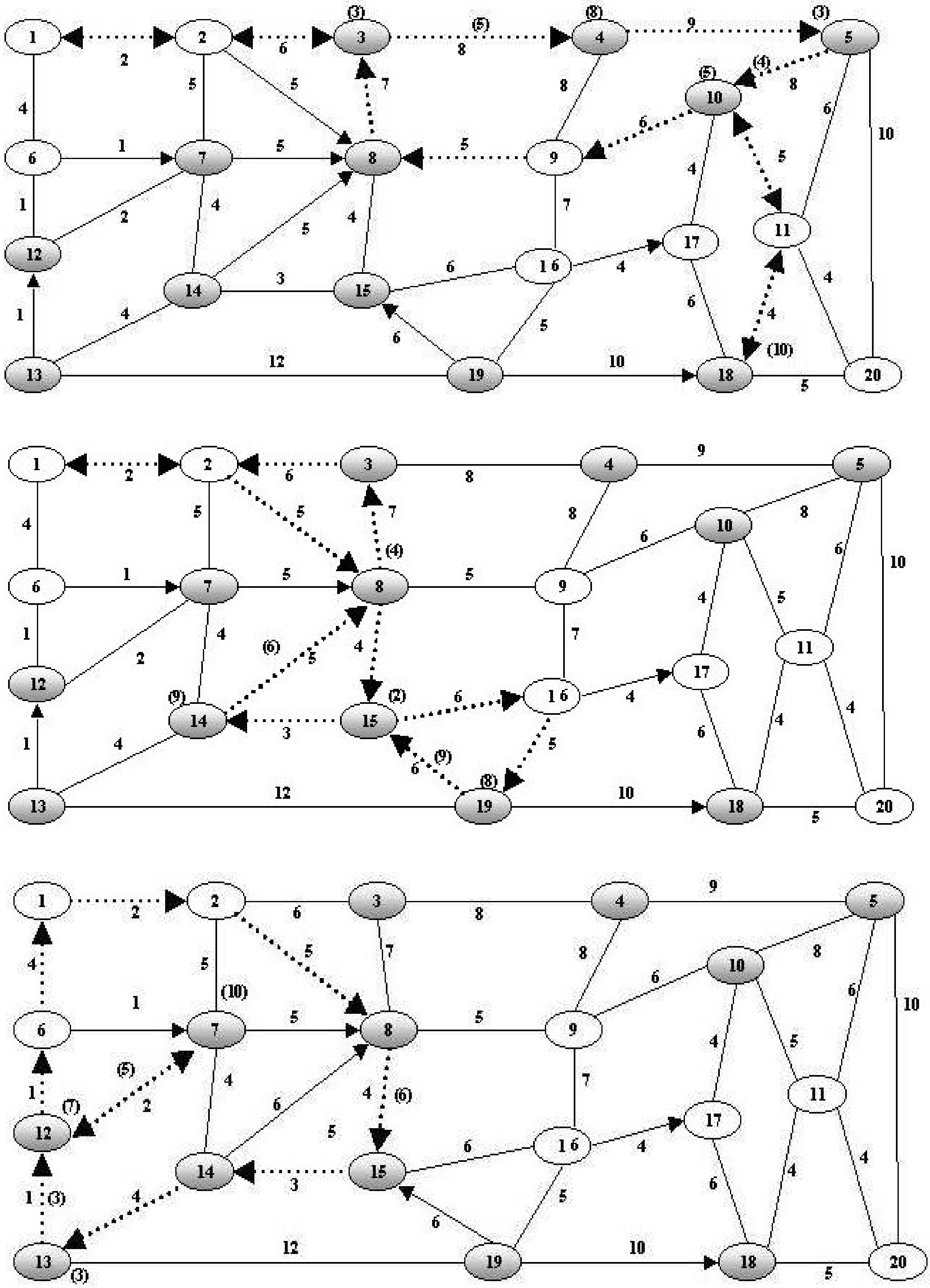

Figura 5. The three routes obtained from the Giant Tour.

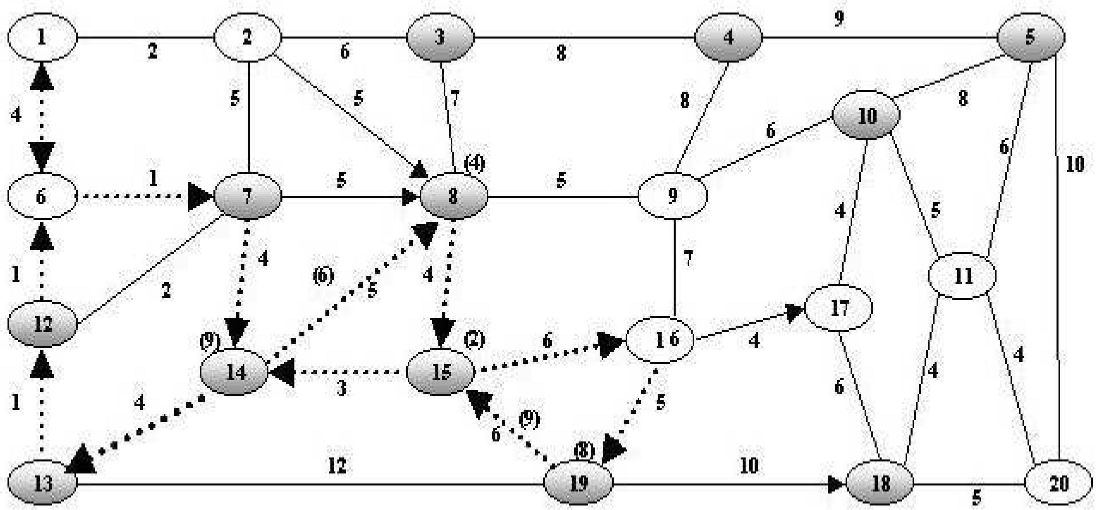

Figure 6. Improvement of route 2 of the heuristic of Pandit and Muralidharan, by means of the GRP heuristic.

4. COMPUTATIONAL RESULTS

In this section we compare the behavior of our heuristic and the one of Pandit and Muralidharan in a set of 28 instances with $20 \leq |V| \leq 50$, $25 \leq |E| \leq 97$, $0 \leq |A| \leq 16$ and $2 \leq k \leq 4$. These instances where manually generated trying to cover different situations (demands located in clusters, non-directed graphs, total demand very close to kW, etc.) with the restrictions imposed by the code of the GRP heuristic. Our heuristic has been partially coded in C++ whereas the Giant Tour obtained by the GRP heuristic of López has been manually broken into subtours, with the help of some subroutines coded in C++.

In Table 1 we show the relevant data corresponding to each instance, whereas in Table 2 we show the costs given by: our heuristic (ASH), our heuristic plus the application of the GRP heuristic to each obtained route (ASH+MC), the heuristic of Pandit and Muralidharan (PM) and the heuristic of Pandit and Muralidharan plus the application of the GRP heuristic to each obtained route (PM + MC). We also present in Table 2 the deviations of the results given by PM with respect to ASH (100(PM/ASH - 1), and the deviations of the results given by PM + MC with respect to ASH and ASH + MC (100((PM + MC)/ASH-1) and 100 ((PM + MC)/(ASH + MC) - 1) respectively).

Number	Q	W	k	$\|V\|$	$\|E\|$	$\|A\|$	$\|V_R\|$	$\|V_{RI}\|$	$\|E_R\|$	$\|A_R\|$
1	110	40	3	20	27	8	12	2	4	3
2	771	300	3	20	25	7	12	3	4	3
3	462	200	3	20	25	7	7	2	3	2
4	425	150	3	20	28	9	11	5	3	2
5	300	110	3	20	25	5	11	2	6	3
6	460	160	3	20	31	5	17	2	15	3
7	771	200	4	20	21	6	12	3	4	4
8	470	170	3	25	35	7	20	2	15	4
9	1020	500	3	25	35	7	11	3	6	2
10	130	50	3	25	37	3	12	3	5	0
11	250	75	4	25	33	5	8	3	5	2
12	240	90	3	25	40	5	8	3	2	3
13	190	80	3	30	52	5	14	2	5	5
14	230	90	3	30	45	5	11	2	3	2
15	255	100	3	30	47	1	9	6	1	1
16	186	80	3	30	35	9	9	2	1	4
17	322	110	3	30	38	7	14	1	7	4
18	256	100	3	35	54	8	11	3	5	1
19	252	110	3	35	54	10	10	1	4	4
20	305	110	3	35	45	11	11	4	1	5
21	428	230	2	40	70	16	13	3	6	5
22	564	300	2	40	54	12	19	3	11	4
23	645	250	3	40	66	9	12	1	6	3
24	330	150	3	45	82	0	12	3	9	0
25	370	130	3	45	59	11	16	2	7	3
26	1343	500	3	50	74	14	21	4	9	3
27	320	120	3	50	76	11	14	3	3	3
28	350	130	3	50	97	1	11	8	1	1

Table 1. Relevant data of the 28 instances (Q = total demand and $|V_{RI}|$ = number of isolated required vertices including the depot).

From these computational results we obtain that:

- On average, the heuristic of Pandit and Muralidharan (PM) produces a cost increment in the solution of 11.05 % with respect to our heuristic and it only produces better solution than ours in one instance, the instance number 8.
- After applying the GRP heuristic to the routes obtained by both CGRP-m heuristics, we can see that our solutions have been slightly improved, in fact, 19 costs where not modified (that means that once demands are assigned to a vehicle, our heuristic produces for this vehicle an optimal or nearly optimal tour solution to satisfy these demands), whereas the improvements produced in the solutions given by heuristic PM where more significant. Anyway, the average cost increment of the second procedure (PM + MC) with respect to the first one (ASH + MC) is still significant; 8.41 % and it only produces the best solution in 4 instances (instances 3, 8, 21 and 28).
- Even if we compare PM+MC with our heuristic without improving the routes of the second one, we obtain an average cost increment in the solution of 7.17 %, being 4 the number of instances in which PM + MC produces better solution than ours (instances 3, 8, 21 and 28).
- In 2 instances from among 28, our heuristic gave a solution with k + 1 vehicles (instances number 5 and 6), whereas the heuristic of Pandit and Muralidharan produced a solution with k + 1 vehicles in 6 instances (instances 4, 5, 6, 7, 12 and 20).

We conclude that, according to these computational results, the CGRP-m heuristic that we have presented here improves substantially the results obtained applying the route-first-partition-next procedure proposed by Pandit and Muralidharan, which is the only method that we have found in the Operational Research literature in order to solve the CGRP-m.

Table 2. Computational results: costs and deviations. AHS = our heuristic; AHS + MC = AHS plus the application of the GRP heuristic to each obtained route; PM = heuristic of Pandit and Muralidharan; PM + MC = PM plus the application of the GRP heuristic to each obtained route.

Number	ASH	PM	$100\left(\frac{PM}{AHS}-1\right)$	ASH + MC	PM + MC	$100\left(\frac{PM+MC}{ASH+MC}-1\right)$	$100\left(\frac{PM+MC}{ASH}-1\right)$
1	145	156	7.58	143	145	1.39	.00
2	397	480	20.90	397	478	20.40	20.40
3	323	327	1.23	321	313	- 2.49	- 3.09
4	113	164	45.13	113	155	37.16	37.16
5	152	170	11.84	152	164	7.89	7.89
6	169	200	18.34	169	185	9.46	9.46
7	495	561	13.33	495	541	9.29	9.29
8	237	207	- 12.65	211	203	- 3.79	- 14.34
9	114	118	3.50	114	118	3.50	3.50
10	130	132	1.53	127	132	3.93	1.53
11	209	232	11.00	209	230	10.04	10.04
12	122	130	6.55	116	123	6.03	.81
13	139	156	12.23	139	143	2.87	2.87
14	128	138	7.81	128	136	6.25	6.25
15	92	93	1.08	92	92	.00	.00
16	153	162	5.88	153	153	.00	.00
17	180	210	16.66	180	210	16.66	16.66
18	136	152	11.76	136	152	11.76	11.76
19	124	148	19.35	124	148	19.35	19.35
20	156	181	16.02	156	181	16.02	16.02
21	161	162	.62	156	153	- 1.92	- 4.96
22	218	235	7.79	218	235	7.79	7.79
23	193	222	15.02	193	202	4.66	4.66
24	223	255	14.34	211	255	20.85	14.34
25	248	283	14.11	244	281	15.16	13.30
26	1395	1695	21.50	1352	1423	5.25	2.00
27	179	198	10.61	179	196	9.49	9.49
28	199	212	6.53	199	196	- 1.50	- 1.50
Average			**11.05**			**8.41**	**7.17**

ACKNOWLEDGEMENTS

Authors would like to thank José María Sanchís for providing us the code of the GRP heuristic corresponding to the work of López (1998), directed by him, and for helping us in its use.

REFERENCES

BENAVENT, E.; V. CAMPOS; A. CORBERAN and E. MOTA (1990): "The capacitated arc routing problem. A heuristic algorithm", **Qüestiió** 14, 107-122.

CHAPLEAU, L.; J.A. FERLAND; G. LAPALME and J.M. ROUSSEAU (1984): "A parallel insert method for the capacitated arc routing problem", **Operations Research Letters** 3, 2, 95-99.

CORBERÁN, A.; R. MARTI and A. ROMERO (2000): "Heuristics for the mixed rural postman problem", **Computers & Operations Research** 27, 183-203.

HERTZ, A.; G. LAPORTE and M. MITTAZ (2000): "A tabu search heuristic for the capacitated arc routing problem", **Operations Research** 48, 1, 129-135.

LOPEZ, M. (1998): "Optimización mediante técnicas de simulación Monte Carlo del recorrido del servicio de recogida de residuos en el municipio de Aldaya (Valencia): caso de trazado urbano con alto número de calles con sentido de circulación prohibido", **Proyecto Final de Carrera**, Escuela Técnica Superior de Ingenieros Industriales de la Universidad Politécnica de Valencia.

PANDIT, R. and B. MURALIDHARAN (1995): "A capacitated general routing problem on mixed networks", **Computers & Operations Research** 22, 5, 465-478.

PEARN, W.L. (1991): "Augment-insert algorithms for the capacitated arc routing problem", **Computers & Operations Research** 14, 4, 285-288.

TAILLARD, E. (1993): "Parallel iterative search methods for vehicle routing problems", **Networks**, 23, 661-673.

VIGO, D. (1996): "A heuristic algorithm for the asymmetric capacitated vehicle routing problem", **European Journal of Operational Research** 89, 108-126.