ARC ROUTING PROBLEMS, PART II: THE RURAL POSTMAN PROBLEM

H. A. EISELT

University of New Brunswick, Fredericton, New Brunswick, Canada

MICHEL GENDREAU and GILBERT LAPORTE

Centre de recherche sur les transports, Université de Montréal, Montréal, Québec, Canada

(Received January 1994; revision received March 1994; accepted June 1994)

This is the second half of a two-part survey on arc routing problems. The first part appeared in the March–April 1995 issue of this journal. Here, the rural postman problem (**RPP**) is reviewed. The paper is organized as follows: applications, the undirected **RPP**, the directed **RPP**, the stacker crane problem, and the capacitated arc routing problem.

This is the second half of a two-part survey on arc routing problems (**ARPs**). This class of problems is defined on a graph or on a multigraph $G = (V, A)$, where V is the vertex set, A is the arc set, and a nonnegative cost matrix $C = (c_{ij})$ is associated with A. The arcs can be directed, undirected, or both. In the Chinese postman problem (**CPP**) reviewed in the first part of the survey (Eiselt, Gendreau and Laporte 1995), one seeks a minimum-cost closed walk on all arcs of A. When it is required to traverse only a subset $R \subseteq A$ of arcs, the problem becomes a rural postman problem (**RPP**). There are few practical contexts where it is necessary to service *all* arcs of a network. Hence, most real-life arc routing applications are usually modeled as **RPPs**. In addition, a constrained version of the **CPP**, called the capacitated arc routing problem (**CARP**), can be viewed as an **RPP**. Our objective is to review the main applications models and algorithms for the **RPP**.

We saw in the **CPP** survey that the pure undirected or directed version of the **CCP** can be solved in polynomial time, whereas the mixed **CPP** and the windy postman problem are both NP-hard. Unfortunately, both the undirected and the directed **RPP** are NP-hard (Lenstra and Rinnooy Kan 1976), except when $R = A$ and the problem becomes a **CPP**. The mixed version of the **RPP**, where R is a set of directed arcs and $A \backslash R$ is a set of edges or undirected arcs is known as the stacker crane problem (**SCP**) and is also NP-hard (Frederickson, Hecht and Kim 1978). Another problem closely related to the **RPP** is the **CARP** introduced by Golden and Wong (1981). In this problem, each arc (v_i, v_j) has a nonnegative weight q_{ij}, and all arcs with positive weight must be traversed by a fleet of identical vehicles of capacity W and based at the depot. The **CARP** is also NP-hard. In fact, the **RPP** is a special case of the **CARP** with $W = |R|$, $q_{ij} = 1$ if $(v_i, v_j) \in R$ and $q_{ij} = 0$ if $(v_i, v_j) \notin R$. As most practical applications contain capacity restrictions, the **CARP** is probably the most important problem in the area of arc routing. We devote an important section to this problem.

As in the case of the **CPP**, a standard algorithmic strategy for **RPPs** is to first determine an augmentation of the graph to make it unicursal, and then obtain in polynomial time an Eulerian cycle or circuit on the augmented graph. As **RPPs** are NP-hard, heuristics can be used for the first phase. These heuristics often embed matching algorithms or shortest spanning tree algorithms to generate an augmented graph that satisfies the unicursality conditions. Exact algorithms for the generation of Eulerian graphs use two of the techniques commonly employed for the traveling salesman problems (**TSP**). In the case of undirected graphs, the problem is formulated as an integer linear program typically containing a large constraint set. This program is solved in a branch-and-cut fashion: A continuous relaxation of the problem is first solved and valid cuts are generated when found to be violated; branching is used to regain integrality and cuts may again be added to the subproblems. This approach is commonly applied to the symmetric **TSP** (Padberg and Rinaldi 1991; Grötschel and Holland 1991). To solve directed problems, some authors solve the spanning arborescence relaxation of the problem, coupled with branch and bound. This again follows a technique used for the asymmetric **TSP** (Fischetti and Toth

Subject classification: Transportation, vehicle routing: rural postman problem, survey.
Area of review: SURVEY, EXPOSITORY AND TUTORIAL.

0030-364X/95/4303-0399 $01.25

1992). However, as we will see, the **RPP** remains difficult to solve to optimality.

The remainder of this paper is organized as follows. Section 1 is devoted to **RPP** applications. Sections 2 and 3 investigate, respectively, the undirected and the directed **RPP**. The **SCP** is analyzed in Section 4, and the **CARP** in Section 5. The conclusion follows in Section 6.

1. APPLICATIONS

The **RPP** underlies several applications in contexts where streets or roads have to be traversed for maintenance, garbage collection, milk or mail delivery, school bus transportation, parking meter collection, electric meter reading, electrical lines and gas mains inspection, etc. Billions of dollars are spent each year on such operations and the potential for savings through optimization is enormous. Most of these applications cannot be modeled as pure **RPPs** as they involve additional characteristics as well. Solution methodologies are often tailored to the situation at hand and require modifications when applied to new contexts. Whether problems are defined on directed, undirected, or mixed graphs depends on the road or street network topology and on the operating policies involved. As a rule, one-way streets are represented by directed arcs, and two-way streets by edges. If the two sides of the street can be serviced at the same time, as is sometimes the case in garbage collection, this mixed graph representation can be appropriate. In several applications, however, the two street sides must be serviced separately, which may even be the case of one-way streets, so that arcs may have to be duplicated, and edges are replaced by two directed arcs, one in each direction. The resulting graph is then directed. Most problems belonging to this category are **RPPs** because usually not all street segments require service. In fact, servicing an arc is generally more costly than just traversing it. Traversing an arc without servicing it is called *deadheading*.

In most applications, several vehicles are involved, and there may be a number of restrictions due to capacity, distance, or travel time, etc. These problems are therefore constrained and often large scale m-vehicle **RRPs** that can rarely be tackled directly by means of a known exact algorithm. A common strategy is to first decompose these problems by breaking the original graph into smaller graphs (see, e.g., Chapleau, Ferland and Rousseau 1985, Levy and Bodin 1989, and Bodin and Levy 1991). Traffic rules such as left-turn, right-turn, or U-turn prohibitions often come into play (see Bodin and Kursh 1979 and McBride 1982). Also, penalties can be imposed for crossing streets, such as in the case of the Canadian postal application treated by Roy and Rousseau (1989). One way to handle such restrictions or penalties is to replicate vertices and impose appropriate penalties between all pairs of replicas (see Roy and Rousseau and Figure 1).

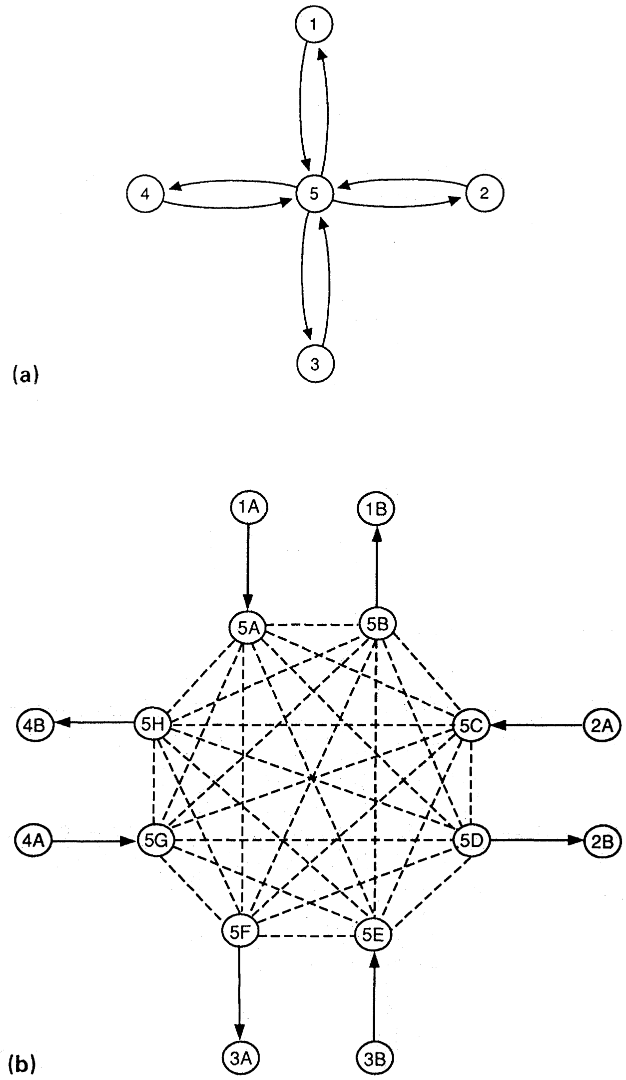


Figure 1. Replicating vertices to handle traffic restrictions or penalties. (a) Original graph. (b) Graph with replicated vertices.

Several researchers and entrepreneurs have responded to the growing demand in the arc routing area by producing generic or customized software. These programs are now developed on microcomputers and make heavy use of geocoding and graphic technologies. While most systems use heuristics, we are aware of some software based on an optimization algorithm for the mixed **CPP**. We now briefly review some representative **ARP** applications documented in the operations research literature.

1.1. Street Sweeping

Bodin and Kursh (1978, 1979) describe a computer-assisted system for the routing and scheduling of street sweepers, together with computational experiences derived from two pilot studies in New York City and Washington, D.C. In addition to the algorithm, the authors describe important preprocessing steps, such as data preparation and network verification. An important aspect of street sweeping is that particular streets can only be swept at certain times because of parking regulations. Thus, the problem involves selecting which streets to include in each route so that workloads are balanced and streets are serviced at suitable frequencies. Eglese and Murdock (1991) describe the results of a street sweeping project undertaken for the Lancashire County Council in England. They point out some important differences between their network and the type of urban network considered in Bodin and Kursh (1978). First, all streets can be regarded as two-way streets because even "in the few cases where one-way streets exist, they are normally swept in one direction by a sweeper going against the traffic taking suitable precautions." Treating all roads as two-way streets means that less data storage is required. Second, contrary to the Bodin and Kursh application, parking regulations do not have a major impact on the problem and all roads are practically always available for sweeping.

1.2. Snow Plowing

Haslam and Wright (1991) describe an algorithm used for the design of highway snow and ice control in Indiana. State roads are classified in several categories with varying priority levels for snow clearance. Lemieux and Campagna (1984) deal with a similar situation. Thus, their problem may be viewed as a hierarchical **RPP** (see Dror, Stern and Trudeau 1987) although precedence relations between the various road classes are not always strictly enforced. In addition to determining the composition and size of the fleet, the authors address a number of safety considerations, the need to replenish vehicles with sand and chemicals during operations, contingencies to be dealt with in case of storm intensification, the best allocation of vehicles to roads, the number, size and location of service centers, etc. Eglese and Li (1992) conducted a study on gritting operations, again for the Lancashire Country Council. The show how the efficiency of the optimal solution is affected by the configuration of the road network. Other studies related to snow removal include those of Alprin (1975), Cook and Alprin (1976), Tucker and Clohan (1979), England (1982), and Gélinas (1992).

1.3. Garbage Collection

Waste collection is probably the most common **ARP** application. One of the first authors to study this problem within the context of arc routing was Stricker (1970). He developed a computerized arc routing algorithm for the urban waste collection problem. This method was tested on real data from the city of Cambridge, Massachusetts. Beltrami and Bodin (1974), Bodin et al. (1989), and Negreiros (1990) also describe computer systems for garbage collection planning. The Beltrami and Bodin study was implemented in New York City and, as a result, substantial benefits were achieved. The second project was undertaken for the town of Oyster Bay, New York. The Negreiros study has served as a basis for a major project in Rio de Janeiro. One important question associated with the design of garbage collection routes is that of determining which dump site should be used by a vehicle that has become saturated. As in street sweeping, the determination of garbage collection days for each street is closely related to route design. Additional studies on waste collection are those of Wyskida and Gupta (1973), Turner and Hougland (1975), Clark and Gillean (1975), Clark and Lee (1976), Male and Liebman (1978), Minazzato (1988), Gelders and Cattrysse (1991), and Alvarez-Valdés et al. (1993).

1.4. Mail Delivery

Major studies related to postal deliveries have been undertaken by Levy (1987) and Levy and Bodin (1988, 1989) in the United States, and by Roy and Rousseau (1989) in Canada. In the U.S. situation, the problem is to partition the arcs into clusters, to locate depots used as starting points for the postal carriers' route, and to determine tours in each cluster. Clusters must be balanced and correspond to the maximum mail volume that may be carried at the same time. Each day, a postal carrier drives a car from the post office to a parking location called a *depot*. The corresponding driving time is assumed to be negligible. The carrier then delivers mail to one of the clusters adjacent to the depot, returns to his car, delivers to another adjacent cluster adjacent to the same depot, and so on, until all clusters associated with that depot have been covered. The carrier then drives to another depot, and follows the same procedure. This is repeated until a full day's work has been completed. Apart from the actual arc routing, a major concern in this application is to suitably determine the number and locations for the depots. This problem is a location-routing problem (Laporte 1988) in an arc routing context. Roy and Rousseau also view their problem as a location-routing problem, except that this time there is a different depot associated with the start of each postal carrier's route. The cost of locating a depot is equal to the cost of travel between it and the post office. All routes including this deadheading time are limited by the maximum duration of a working day, but not by capacity because relay boxes are conveniently located along each route. For more details on this subject, see Bouliane and Laporte (1992).

1.5. School Bus Routing

Few arc routing applications are as intricate as school bus routing. Here, the two objectives of minimizing the number of buses and mileage are often optimized in a hierarchical fashion as fixed costs tend to predominate. In addition to the usual traffic restrictions, several constraints must be satisfied, such as time windows, minimum and maximum number of passengers in the bus at any given time, student mix, and maximum time spent by any student on the bus. The school bus routing problem was addressed by Angel et al. (1972), Bennett and Gazis (1972), Bodin and Berman (1979), and Swersey and Ballard (1984). More recent implementations are those of Desrosiers et al. (1986) who used a column generation approach to schedule buses for some 60 schools and 20,000 students, and the system designed by Braca, Bramel et al. (1993) for New York City.

1.6. Meter Reading

Stern and Dror (1979) describe an m-vehicle version of the **RPP** associated with electric meter reading. The objective is to design m open routes of limited duration covering a given area, and starting at the depot. The objective is to minimize m. Using data from the city of Beersheva in Israel, the authors report that savings of up to 40% over the cost of the existing system could be achieved. A similar problem is treated by Wunderlich et al. (1992). These authors state that the use of a computerized system to organize meter reader routes for the Southern California Gas Company results in projected tangible annual benefits in the region of $875,000.

2. THE UNDIRECTED RURAL POSTMAN PROBLEM

We now turn our attention to particular classes of the **RPP**. In the undirected **RPP**, all arcs are undirected. An easy case occurs when the graph $\bar{G} = (V, R)$ is connected. The problem can then be solved by computing shortest chains (in G) between odd-degree vertices (some of these chains may contains edges in $A \backslash R$), and then proceeding as for the undirected **CPP** (Eiselt, Gendreau and Laporte). In general, the **RPP** is solved on a modified graph $G' = (V', A')$ defined as follows: $V' = \{v_i \in V: (v_i, v_j) \in R \text{ for some } v_j \in V\}$, and A' is obtained by first adding to A an edge (v_i, v_j) for each v_i, $v_j \in V'$ whose cost c_{ij} is equal to the length of a shortest chain between v_i and v_j; then delete all edges $(i, j) \in A \backslash R$ for which $c_{ij} = c_{ik} + c_{kj}$ for some k, as well as one of two parallel edges if they have the same cost. To illustrate, consider the two graphs shown in Figure 2, where the edges of R are shown by bold lines and the numbers correspond to edge costs. In the transformed graph G' the set R induces p connected components $G_1, \ldots, G_p$ with respective vertex sets $V_1, \ldots, V_p$ forming a partition of V'. If (c_{ij}) satisfies the triangle inequality, a heuristic for the undirected **RPP**, with a worst-case ratio of 3/2 can be constructed along the lines of the Christofides (1976) heuristic for symmetrical **TSPs**, as suggested by Frederickson (1979).

Heuristic for the Undirected RPP

STEP 1. (**Shortest spanning tree**). Construct the shortest spanning tree T connecting $G_1, \ldots, G_p$ and let $\ell(T)$ be its length. Denote by $\ell(R)$ the sum of lengths of all edges of R and by z^* the value of an optimal **RPP** solution; then $\ell(T) + \ell(R) \leqslant z^*$.

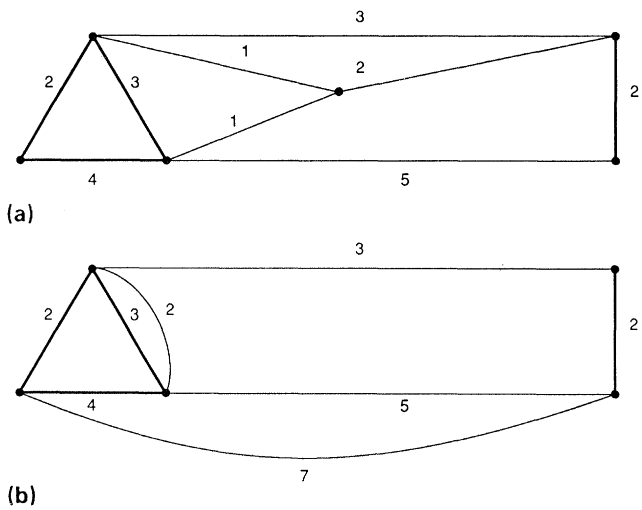


Figure 2. Graph transformation for the undirected **RPP**. (a) Original graph G. (b) Transformed graph G'.

STEP 2. (**Minimum cost matching**). Determine a minimum cost matching of all odd-degree vertices of the graph induced by $R \cup T$. Let M be the set of edges included in the optimal matching, and let $\ell(M)$ be their total length.

STEP 3. (**Eulerian cycle**). An **RPP** solution is given by an Eulerian cycle on the graph induced by $R \cup T \cup M$. If (c_{ij}) satisfies the triangle inequality, it can be shown, using the same argument as in Christofides (1976), that $\ell(M) \leqslant z^*/2$. Hence, the length $\ell(T) + \ell(R) + \ell(M)$ of the Euler cycle does not exceed $3z^*/2$.

Two integer linear programming formulations have been proposed for the **RPP**. In the first, suggested by Christofides et al. (1981), x_{ij} variables are defined as follows: if $(v_i, v_j) \in R$, x_{ij} is the number of times edge (v_i, v_j) is replicated in the optimal **RPP** solution, i.e., edge (v_i, v_j) will be traversed $1 + x_{ij}$ times; if $(v_i, v_j) \in A' \backslash R$, x_{ij} is the number of times edge (v_i, v_j) is traversed. The formulation is then as follows.

Problem URPP1

Minimize

$$\sum_{(v_i,v_j)\in R} c_{ij}(1 + x_{ij}) + \sum_{(v_i,v_j)\in A'\backslash R} c_{ij}x_{ij} \tag{1}$$

subject to

$$\sum_{\substack{(v_i,v_j)\in R\\ j>1}} (1 + x_{ij}) + \sum_{\substack{(v_i,v_j)\in R\\ j<i}} (1 + x_{ji}) + \sum_{\substack{(v_i,v_j)\in A'\backslash R\\ j>i}} x_{ij}$$

$$+ \sum_{\substack{(v_i,v_j)\in A'\backslash R\\ j<i}} x_{ji} = 0 \pmod 2 \quad (v_i \in V') \tag{2}$$

$$\sum_{v_i \in S, v_j \in \bar{S}} x_{ij} \geq 1 \left(S = \bigcup_{k \in P} V_k,\ \bar{S} = \left(\bigcup_{k=1}^{p} V_k \right) \backslash S, \right.$$

$$\left. \cdot P \subset \{1, \ldots, p\} \right) \tag{3}$$

$$x_{ij} \geq 0 \quad \text{and integer} \quad ((v_i, v_j) \in A'). \tag{4}$$

In this formulation, constraints (2) mean that in the solution, the degree of every vertex is even, while constraints (3) ensure that in the optimal cycle all connected components will be linked together. It is relatively easy to show that x_{ij} can be bounded above by 1 if $(v_i, v_j) \in R$, and by 2 if $(v_i, v_j) \in A' \backslash R$. Also (2) can be replaced by (2′), obtained by making their right-hand side equal to $2z_i$, with

$$z_i \geq 0 \quad \text{and integer} \quad (v_i \in V'). \tag{5}$$

As in the Held and Karp (1971) algorithm for the **TSP**, the authors derive a lower bound on the problem by incorporating constraints (2′) in a Lagrangian fashion in the objective function, and by observing that the subproblem defined by the x_{ij} variables associated with $A' \backslash R$ is a shortest spanning tree (**SST**) problem over a graph whose vertices correspond to the connected components of G'. Also, using the **SST**, the authors derive an upper bound for the **RPP** by matching odd-degree vertices. These bounding procedures were embedded within a branch-and-bound scheme. Twenty-four randomly generated problems were solved to optimality. The characteristics of these problems were: $9 \leq |V| \leq 84$, $13 \leq |A| \leq 184$, $4 \leq |R| \leq 78$, and $2 \leq p \leq 8$.

A related, but different, formulation was later proposed by Corberán and Sanchis (1991a). Define $A_i = \{(v_i, v_j) \in A'\}$, the set of all edges incident to v_i in G'. A vertex is called R-even (respectively, R-odd) if it is incident to an even (respectively, odd) number of edges of R. Variables x_{ij} are defined as in **URPP1**. Then the problem is as follows.

Problem URPP2

Minimize

$$\sum_{(v_i, v_j) \in R} c_{ij}(1 + x_{ij}) + \sum_{(v_i, v_j) \in A' \backslash R} c_{ij} x_{ij} \tag{6}$$

subject to

$$\sum_{(v_i, v_j) \in A_i} x_{ij} = 0 \pmod 2 \quad (v_i \in V',\ v_i \text{ is } R\text{-even}) \tag{7}$$

$$\sum_{(v_i, v_j) \in A_i} x_{ij} = 1 \pmod 1 \quad (v_i \in V',\ v_i \text{ is } R\text{-odd}) \tag{8}$$

$$\sum_{\substack{v_i \in S, v_j \in \bar{S} \\ \text{or } v_i \in \bar{S}, v_j \in S}} x_{ij} \geq 2 \left(S = \bigcup_{k \in P} V_k,\ \bar{S} = \left(\bigcup_{k=1}^{p} V_k \right) \backslash S, \right.$$

$$\left. \cdot P \subset \{1, \ldots, p\} \right) \tag{9}$$

$$x_{ij} \geq 0 \quad \text{and integer} \quad ((v_i, v_j) \in A'). \tag{10}$$

Sanchis (1990) and Corberán and Sanchis (1991a) identified a number of families of facets of the polytope of the convex hull of feasible solutions defined by (7)–(10). In a different report (1991b), these authors show that all the facet-inducing inequalities for the so-called Road-**TSP** (see Cornuéjols, Fonlupt and Naddef 1985 and Fleischmann 1988) are also facets for the undirected **RPP**. Following the work of Padberg and Grötschel (1985), Corberán and Sanchis (1991b) incorporated a number of facet generation routines within a branch-and-bound scheme that they applied to the 24 test problems of Christofides et al. (1981). Twenty-three of these problems were solved to optimality at the root node of the search tree, using only cutting planes. Two additional problems derived from the street network of Albaida (Valencia) were also solved without branching.

3. THE DIRECTED RURAL POSTMAN PROBLEM

The directed **RPP** is defined on a graph $G = (V, A)$ where A is now a set of directed arcs. The problem reduces to a directed **CPP** whenever $\bar{G} = (V, R)$ is connected. Also, in the general case, the problem is solved on a modified graph $G' = (V', A')$ constructed as in the undirected case, except that the arcs (v_i, v_j) that are introduced are now directed and have a length c_{ij} equal to the length of a shortest path from v_i to v_j. The connected components $G_1, \ldots, G_p$ with respective vertex sets are defined as above. Christofides et al. (1986) propose the following heuristic for the directed **RPP**.

Heuristic for the Directed RPP

STEP 1. (**Shortest spanning arborescence** (**SSA**)). Construct the shortest spanning arborescence (see Edmonds 1967) rooted at an arbitrary vertex, and connecting $G_1, \ldots, G_p$. Let $\tilde{G}$ be the resulting graph.

STEP 2. (**Transportation problem**). As for the directed **CPP** (see Eiselt, Gendreau and Laporte), derive an Eulerian graph from $\tilde{G}$ by adding arcs in a least-cost manner so that the number of incoming arcs of each vertex is equal to the number of outgoing arcs.

STEP 3. (**Eulerian circuit**). Determine an Eulerian circuit on the augmented graph.

The same procedure can be repeated by considering in turn all vertices as the root of the **SSA**, and then selecting the best solution. Using this heuristic, the authors solved twenty-three instances within 1.3% of optimality on the average; the largest recorded deviation was 5% and ten of the solutions were optimal.

A mathematical programming formulation and an exact algorithm were also proposed by Christofides et al. (1986). We present here a slightly simplified formulation. If $(v_i, v_j) \in R$, x_{ij} is the number of times arc (v_i, v_j) is replicated in the optimal **RPP** solution; if $(v_i, v_j) \in A' \backslash R$, then x_{ij} is the number of times arc (v_i, v_j) is traversed. Also define

$$b_{ij} = \begin{cases} 1 & \text{if } (v_i, v_j) \in R,\ v_i, v_j \in V' \\ 0 & \text{otherwise,} \end{cases}$$

$$\bar{b}_{ij} = \begin{cases} 1 & \text{if } (v_i, v_j) \in A' \backslash R,\ v_i, v_j \in V' \\ 0 & \text{otherwise.} \end{cases}$$

Then the problem can be formulated with respect to a particular vertex set $\bar{V} \in \{V_1, \ldots, V_p\}$ as follows.

Problem DRPP

Minimize

$$\sum_{(v_i,v_j)\in R} (1 + x_{ij}) + \sum_{(v_i,v_j)\in A'\backslash R} x_{ij} \tag{11}$$

subject to

$$\sum_{\substack{j \\ (v_i,v_j)\in R}} (1 + x_{ij}) b_{ij} + \sum_{\substack{j \\ (v_i,v_j)\in A'\backslash R}} x_{ij} \bar{b}_{ij}$$

$$= \sum_{\substack{j \\ (v_j,v_i)\in R}} (1 + x_{ji}) b_{ji} + \sum_{\substack{j \\ (v_j,v_i)\in A'\backslash R}} x_{ij} \quad (v_i \in V') \tag{12}$$

$$\sum_{v_i \in S, v_j \in \bar{S}} x_{ij} \geq 1 \Big(S = \bigcup_{k\in P} V_k,\ \bar{S} = \Big(\bigcup_{k=1}^{p} V_k\Big) \backslash S,$$

$$\cdot P \subset \{1, \ldots, p\},\ \bar{V} \subseteq S \Big) \tag{13}$$

$$x_{ij} \geq 0 \quad \text{and integer} \quad ((v_i, v_j) \in A'). \tag{14}$$

Constraints (12) stipulate that the in-degree of every vertex is equal to its out-degree. Constraints (13) ensure that the solution forms a spanning arborescence rooted at the component induced by $\bar{V}$, over all connected components $G_1, \ldots, G_p$. Similar to the undirected case, the problem is solved within a branch-and-bound process by incorporating constraints (12) in the objective function in a Lagrangian fashion, and solving the remaining **SST** problem over the connected components. Twenty-three out of twenty-four instances with $13 \leq |V| \leq 80$, $24 \leq |A| \leq 180$, $7 \leq |R| \leq 74$, and $2 \leq p \leq 8$ were solved to optimality using this approach.

4. THE STACKER CRANE PROBLEM

The stacker crane problem (**SCP**) is defined on a mixed graph $G = (V, A \cup E)$, where A is a set of directed arcs, and E is a set of undirected edges. The problem is to determine a shortest circuit including each arc of A at least once. The arcs of A can be viewed as movements to be performed by a crane, each exactly once, in a specified direction. The arcs can also be interpreted as deliveries to be made by a vehicle: the **SCP** then amounts to minimizing the sum of interdelivery costs. If the cost c_{ij} of every arc (v_i, v_j) of A is zero, the **SCP** is equivalent to a **TSP**, and thus the **SCP** is NP-hard.

Frederickson, Hecht and Kim proposed two heuristics that require G to satisfy the following properties: 1) each vertex is incident to at least one arc of A, and 2) the c_{ij}'s on edges satisfy the triangle inequality. If G does not satisfy these two properties, it can be transformed into an equivalent graph G' that satisfies them. The problem is then solved on G' and the solution can be interpreted in terms of the original graph G. In what follows, we assume that G satisfies 1 and 2. The two heuristics are called **LARGEARCS** and **SMALLARCS**. The first performs better when the total cost of the arcs is large compared with the cost of the optimum tour, and vice versa.

Algorithm LARGEARCS

STEP 1. (**Bipartite graph**). Extract from G the set of directed arcs (Figure 3a). Construct a bipartite graph G' in which the first set of vertices corresponds to the tails of the arcs of A, and the second set to the heads. Create edges between all heads and tails not corresponding to the same arc, with the appropriate travel costs corresponding to shortest path lengths (Figure 3b).

STEP 2. (**Matching**). Solve a minimum cost bipartite matching problem on G'. Construct a directed graph G'' having an arc set consisting of A, and all edges of the matching solution directed from head to tail. In the example, we assume the matching solution is {(1, 4), (2, 3), (5, 8), (6, 7)} (see Figure 3c).

STEP 3. (**Shortest spanning tree**). The graph G'' consists of a number of disjoint connected components. Determine a shortest tree spanning these components, using the original edge costs (the cost of an edge between two components being equal to the cost of the cheapest connection between these components) (see Figure 3d).

STEP 4. (**Euler tour**). Construct a directed graph $\bar{G}$ having an arc set consisting of all arcs of G'', and two arcs for each edge of the shortest spanning tree, one in each direction (Figure 3e). This graph is Eulerian and the required **SCP** solution is given by solving a directed **CPP** on $\bar{G}$.

It is easy to verify that the time complexity of the **LARGEARCS** algorithm is $O(\max\{|V|^3, |A|^3\})$. Furthermore, if the cost matrix satisfies the triangle inequality, the procedure produces a tour T of length $\ell_L(T)$ satisfying

$$\ell_L(T) \leq 3z^* - 2\ell(A), \tag{15}$$

where z^* is the value of the optimal **SCP** solution, and $\ell(A)$ is the sum of lengths of all arcs of A (Frederickson, Hecht and Kim).

The second algorithm, **SMALLARCS**, constructs a solution that satisfies sufficiency conditions for a mixed graph to be Eulerian: It should be even, symmetric, and balanced.

Algorithm SMALLARCS

STEP 1. (**Contracted graph**). Construct from G a contracted undirected graph G^* as follows. Define a vertex for every arc of A, and add edges between these vertices. The cost of an edge between vertices corresponding two arcs (v_i, v_j) and (v_k, v_ℓ) is equal to $\min\{c_{ik}, c_{i\ell}, c_{jk}, c_{j\ell}\}$ (Figure 4b). Determine all shortest chains between pairs of vertices of G^* and replace the edge by the corresponding shortest chain thus obtaining graph G^{**} (Figure 4c).

STEP 2. **(Shortest spanning tree).** Determine a shortest spanning tree on G^{**} (Figure 4d).

STEP 3. **(Minimum cost matching).** Perform a minimum cost matching of the odd-degree nodes in the spanning tree (Figure 4e).

STEP 4. **(Re-expanded graph).** Replace all vertices by the corresponding arc of A, and replace each spanning and matching edge by the corresponding shortest chain in Step 1, thus obtaining a mixed graph $\hat{G}$ (Figure 4f). Note that if (v_i, v_j) is an arc of G, then both v_i and v_j have the same degree parity. If v_i and v_j both have an odd degree, add edge (v_i, v_j), and temporarily direct if from v_j to v_i. Let $G^\#$ be the resulting graph (Figure 4g).

STEP 5. **(First Eulerian tour).** Find an Eulerian tour in $\bar{G}$, ignoring even arc directions. If the length of the even arcs which are traversed backwards exceeds $\ell(\bar{A})/2$, reverse the direction of the tour.

STEP 6. **(Added arcs).** Associate with each edge the direction of the tour. If an even arc (v_i, v_j) is traversed in the wrong direction, add two arcs (v_j, v_i). Let $\tilde{G}$ be the resulting directed graph (Figure 4h).

STEP 7. **(Final Eulerian tour).** Determine an Eulerian circuit on $\tilde{G}$.

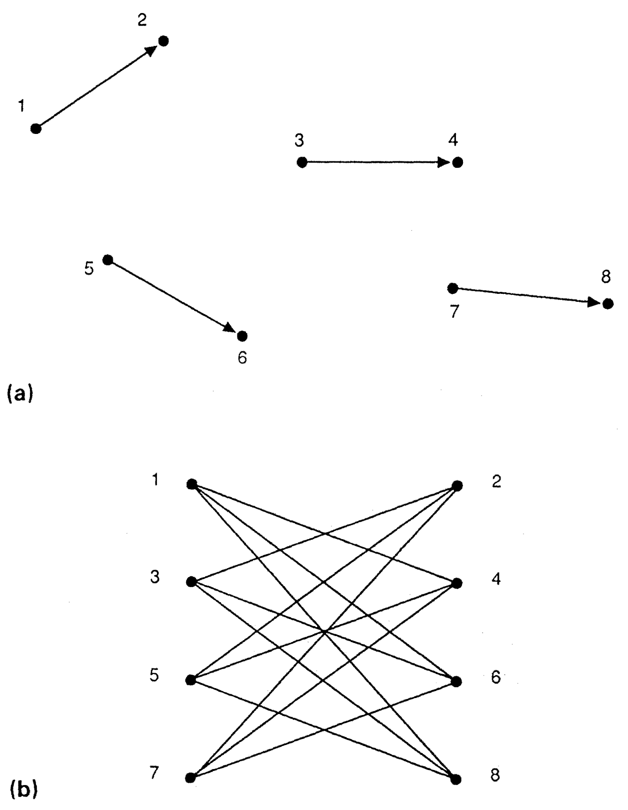
The **SMALLARCS** algorithm also has a time complexity of $O(\max\{|V|^3, |A|^3\})$. If the cost matrix satisfies the triangle inequality, the length $\ell_S(T)$ of the tour satisfies

$$\ell_S(T) \leq \tfrac{1}{2}\,(3z^* + \ell(A)). \tag{16}$$

An algorithm with better worst-case performance is obtained by applying both **LARGEARCS** and **SMALLARCS**, and taking the best solution. If $\ell(T) = \min\{\ell_L(T), \ell_S(T)\}$, then

$$\ell(T) \leq 9z^*/5 \tag{17}$$

if the triangle inequality is satisfied. The validity of this result rests on the fact that $\ell_S(T) \leq \ell_L(T)$ whenever $\ell(A) \leq 3z^*/5$. Thus, if $\ell(A) \geq 3z^*/5$, (15) can be rewritten as $\ell_L(T) \leq 9z^*/5$; if $\ell(A) < 3z^*/5$, (16) becomes $\ell_S(T) \leq 9z^*/5$.



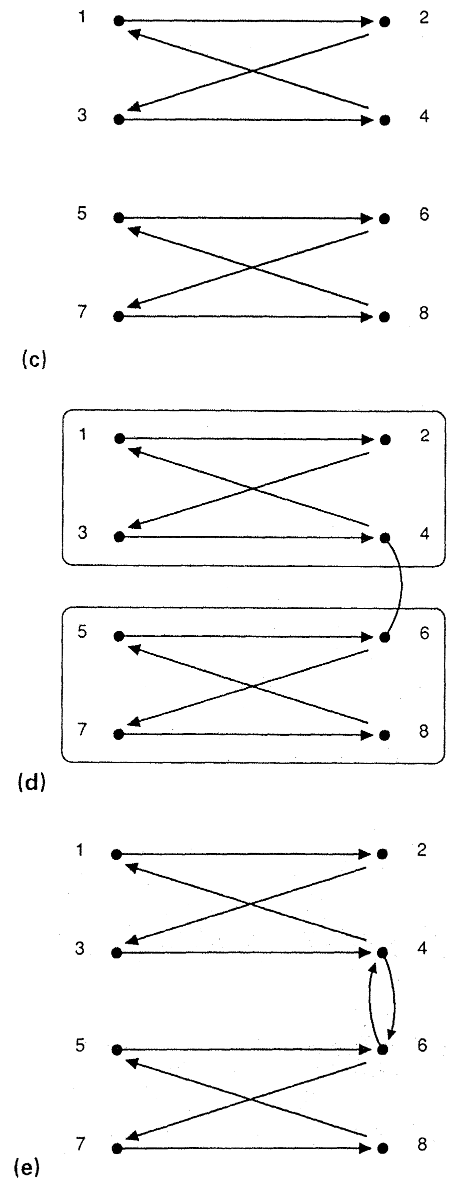


Figure 3. Illustration of the **LARGEARCS** algorithm. (a) Original arc set A. (b) Bipartite graph G'. (c) Graph G''. (d) Shortest spanning tree with two connected components. (e) Directed graph $\bar{G}$.

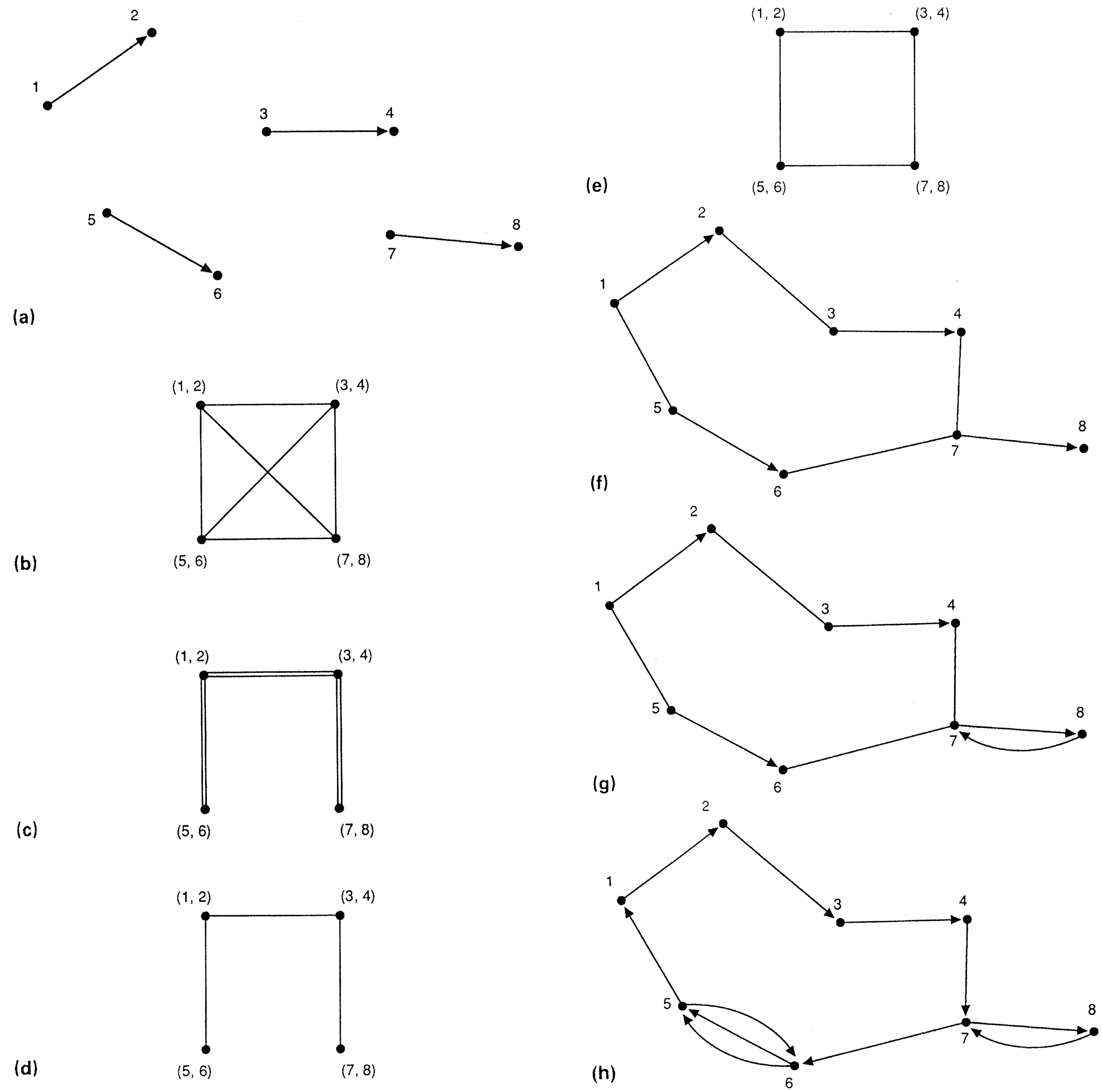


Figure 4. Illustration of the **SMALLARCS** algorithm. (a) Original arc set A. (b) Undirected graph G^*. (c) Directed graph G^{**}. (d) Shortest spanning tree on G^{**}. (e) Matching odd-degree vertices. (f) Mixed graph $\hat{G}$. (g) Mixed graph $G^{\#}$. (h) Directed graph $\tilde{G}$.

An arc insertion heuristic, adapted from the **TSP** insertion heuristic of Rosenkrantz, Stearns and Lewis (1977) has been developed by Lukka and Salminen (1987). Its time complexity is $O(|A|^2 \log |A|)$, and its worst-case performance can be arbitrarily bad. However, on randomly generated problems, the quality of the solutions produced by the insertion algorithm falls somewhere between $\ell_L(T)$ and $\ell_S(T)$, usually close to the best of these two values. One interesting feature of the insertion heuristic lies in its flexibility, as it can easily be modified to incorporate multiple vehicles and side constraints.

Finally, note that Frederickson, Hecht and Kim also investigated an m-vehicle version of the **SCP**, where the objective is to minimize the length of the longest tour. They provide a heuristic algorithm running in $O(\max\{|V|^3, |A|^3\})$ time, and having a worst-case performance ratio of $14/5 - 1/m$.

5. THE CAPACITATED ARC ROUTING PROBLEM

Recall that in the capacitated arc routing problem (**CARP**), each arc (v_i, v_j) has a nonnegative demand or weight q_{ij}. It is assumed that a fleet of m homogeneous vehicles of capacity W are based at a depot located at vertex v_1, say. The **CARP** consists of determining a minimum cost traversal of all arcs by the vehicles so that the total demand of all arcs serviced by any vehicle does not exceed W. Note that a vehicle may traverse an arc without servicing it, but each arc must be serviced by exactly one vehicle. The **CARP** is also a generalization of the capacitated Chinese postman problem (**CCPP**) in which $q_{ij} > 0$ for all (v_i, v_j) (Christofides 1973).

Now consider the capacitated vehicle routing problem (**CVRP**) in which demands are associated with vertices. This problem consists of designing vehicle routes of total minimal length, so that each route starts and ends at the depot, each remaining vertex is visited exactly once by a vehicle, and the sum of demands on any route does not exceed W. For a recent survey, see Laporte (1992). Golden and Wong show how the **CVRP** can be transformed into a **CARP** by splitting each vertex into two vertices joined by an arc, and by assigning the original vertex demand to that arc. The reverse transformation has been proposed by Pearn, Assad and Golden (1987) for undirected graphs; the directed case is similar. These authors transform a **CARP** into a **CVRP** by replacing each arc (v_i, v_j) with positive demand q_{ij} with three vertices s_{ij}, m_{ij} and s_{ji}, each having a demand equal to $q_{ij}/3$. In this transformation, the distances between vertex pairs are defined as

$$d(s_{ij}, s_{kl}) = \begin{cases} (c_{ij} + c_{kl})/4 + p(v_i, v_k) & (v_i, v_j) \neq (v_k, v_l) \\ 0 & (v_i, v_j) = (v_k, v_l) \end{cases}$$

$$d(v_1, s_{ij}) = c_{ij}/4 + p(v_i, v_1)$$

$$d(m_{ij}, v) = \begin{cases} c_{ij}/4 & (v = s_{ij} \text{ or } s_{ji}) \\ \infty & \text{otherwise,} \end{cases}$$

where $p(v_i, v_j)$ is the length of a shortest path from v_i to v_j in G. With these distances, strictly enforcing the constraint that a customer (vertex) can only be visited by a single vehicle in the **CVRP** guarantees that, for any arc (v_i, v_j), s_{ij}, m_{ij} and s_{ji} appear consecutively on the same route in any **CVRP** solution, thus yielding the equivalence between the original and the transformed problems. The number of vertices resulting from this transformation is equal to $1 + 3|R|$, where $R = \{(v_i, v_j): q_{ij} > 0\}$. The interest of these transformations is mostly formal and their algorithmic value has yet to be demonstrated.

The α-approximate version of a problem is defined as the problem of finding a solution whose cost is at most $(1 + \alpha)$ times that of the optimal solution. Golden and Wong have shown that if $C = (c_{ij})$ satisfies the triangle inequality, then the 0.5-approximate **CCPP** is NP-complete. For this, they reduce the partition problem, known to be NP-complete, into a 0.5-approximate **CCPP**. Hence, the 0.5-approximate **CARP** and **CVRP** are also NP-complete. Pearn (1984) and Win (1987) have shown that this results holds even when G is a path. Sahni and Gonzalez (1976) have shown that when the cost matrix does not satisfy the triangle inequality, solving the α-approximate **TSP** is NP-complete for any finite α and hence, this is true for the **CVRP** and the **CARP**.

Solvable cases of the **CCPP** have been investigated by Assad, Pearn and Golden 1987 and by Busch 1991. If G is a linear graph, all vehicles have the same capacity W, and all arcs have the same demand $q_{ij} = 1$, then the **CCPP** can be solved in $O(|V|)$ time (Assad, Pearn and Golden 1987, Busch 1991). When the vehicle fleet is heterogeneous, i.e., vehicle capacities are not all identical, then the problem can be solved by means of a pseudopolynomial algorithm (Busch). However, **CARPs** defined on linear graphs, but with nonidentical demands, are NP-hard (Busch), and so are **CARPs** defined on trees (Golden and Wong 1981, Busch 1991, Labbé, Laporte and Mercure 1991). Finally, Assad, Pearn and Golden have shown that the **CCPP** with identical demands and defined on a cycle graph is solvable in $O(|V|)$ time, and the **CCPP** defined on a complete graph can be solved in polynomial time if all q_{ij}'s are no greater than $W/|V|$ when $|V|$ is odd, and no greater than $W/(|V| - 1)$ when $|V|$ is even.

Two integer linear programming formulations have been proposed for the undirected **CARP**. The first, by Golden and Wong, uses directed variables and that of Belenguer and Benavent (1991) uses undirected variables. In the directed **CARP** formulation, binary variables x_{ijk} are equal to 1 if and only if edge (v_i, v_j) is *traversed* from v_i to v_j by vehicle k, and binary variables y_{ijk} are equal to 1 if and only if (v_i, v_j) is *serviced* by vehicle k while traveling from v_i to v_j. Note that x_{ijk} is bounded above by 1 as it is never optimal for a vehicle to traverse an edge more than once in any given direction. Recall that all arcs (v_i, v_j) with $q_{ij} > 0$ must be serviced, but the remaining arcs may also traversed. Also define $E(S) = \{(v_i, v_j): v_i \in S, v_j \in V\backslash S \text{ or } v_i \in V\backslash S, v_j \in S\}$ and $E^+(S) = E(S) \cap \{(v_i, v_j) \in E: q_{ij} > 0\}$. The formulation is then as follows.

Problem DCARP

Minimize

$$\sum_{k=1}^{m} \sum_{(v_i,v_j)\in A} c_{ij}x_{ijk} \tag{18}$$

subject to

$$\sum_{(v_j,v_i)\in A} x_{jik} - \sum_{(v_i,v_j)\in A} x_{ijk} = 0 \quad (v_i \in V,\ k = 1, \ldots, m) \tag{19}$$

$$\sum_{k=1}^{m} (y_{ijk} + y_{jik}) = \begin{cases} 0 & \text{if } q_{ij} = 0 \\ 1 & \text{if } q_{ij} > 0 \end{cases} \quad ((v_i, v_j) \in A) \tag{20}$$

$$x_{ijk} \geq y_{ijk} \quad ((v_i, v_j) \in A,\ k = 1, \ldots, m) \tag{21}$$

$$\sum_{(v_i,v_j)\in A} q_{ij}y_{ijk} \leq W \quad (k = 1, \ldots, m) \tag{22}$$

$$\left.\begin{array}{l} \sum\limits_{v_i,v_j\in S} x_{ijk} \leq |S| - 1 + n^2 u_k^S \\ \sum\limits_{v_i\in S}\sum\limits_{v_j\notin S} x_{ijk} \geq 1 - w_p^S \\ u_k^S + w_k^S \leq 1 \\ u_k^S,\ w_k^S \in \{0, 1\} \end{array}\right\} \begin{array}{l}(S \subseteq V\backslash\{v_1\};\ S \neq \emptyset; \\ k = 1, \ldots, m\end{array} \tag{23}$$

$$x_{ijk}, y_{ijk} \in \{0, 1\} \quad ((v_i, v_j) \in A;\ k = 1, \ldots, m). \tag{24}$$

In this formulation, constraints (19) are flow conservation equations for each vehicle. Constraints (20) ensure that service arcs correspond to those with a positive demand. Constraints (21) state that an arc is serviced by a vehicle only if it is traversed by the same vehicle. Constraints (22) guarantee that the capacity of a vehicle is never exceeded. Constraints (23) ensure that the solution does not contain any illegal subtours. To see how these constraints operate, observe that for given k and S, only one of the two binary variables u_k^S or w_k^S can take the value 1. Thus, any cycle with vertex set S and arcs traversed by vehicle k must be connected to $V\backslash S$ (and thus to v_1) since

$$\sum_{v_i,v_j\in S} x_{ijk} > |S| - 1 \Rightarrow u_k^S = 1 \Rightarrow w_k^S$$
$$= 0 \Rightarrow \sum_{v_i\in S}\sum_{v_j\notin S} x_{ijk} \geq 1.$$

In the formulation proposed by Belenguer and Benavent for the undirected case, variables x_{ijk} and y_{ijk} are only defined for $i < j$. Moreover, x_{ijk} now represents the number of times edge (v_i, v_j) is traversed by vehicle k without being serviced by that vehicle. The formulation is as follows.

Problem UCARP

Minimize

$$\sum_{k=1}^{m}\sum_{(v_i,v_j)\in A} c_{ij}(x_{ijk} + y_{ijk}) \tag{25}$$

subject to

$$\sum_{k=1}^{m} y_{ijk} = 1 \quad ((v_i, v_j) \in A \text{ and } q_{ij} > 0) \tag{26}$$

$$\sum_{(v_i,v_j)\in A} q_{ij} y_{ijk} \leq W \quad (k = 1, \ldots, m) \tag{27}$$

$$\sum_{(v_i,v_j)\in E(S)} x_{ijk} + \sum_{(v_i,v_j)\in E^+(S)} y_{ijk} \geq 2y_{hlk} (S \subseteq V\backslash\{v_i\};\ S \neq \emptyset;\ k = 1, \ldots, m;\ v_h, v_l \in S \text{ and } q_{hl} > 0) \tag{28}$$

$$\left.\begin{array}{l}\sum\limits_{(v_i,v_j)\in E(S)} x_{ijk} + \sum\limits_{(v_i,v_j)\in E^+(S)} y_{ijk} = 2z_k^S \\ z_k^S \geq 0 \text{ and integer}\end{array}\right\} \begin{array}{l}(S \subseteq V\backslash\{v_1\}; \\ S \neq \emptyset;\ k = 1, \\ \ldots, m\end{array} \tag{29}$$

$$x_{ijk} \geq 0 \text{ and integer};$$
$$y_{ijk} \in \{0, 1\}((v_i, v_j) \in A;\ k = 1, \ldots, m). \tag{30}$$

Here, constraints (26) state that every edge with a positive demand is serviced exactly once by a vehicle. Constraints (27) are capacity constraints. Constraints (28) play a role similar to that of (23): They ensure that if S contains an edge (v_h, v_l) serviced by vehicle k, then S must be connected to its complement by the same vehicle. Constraints (29) stipulate that any nonempty vertex set S not containing the depot must be connected to its complement an even number of times by any vehicle. There is no known way of expressing these constraints in terms of the x_{ijk} and y_{ijk} variables alone. However, Belenguer and Benavent show that the following valid constraints are often but not always sufficient to ensure evenness:

$$-\sum_{(v_i,v_j)\in E(S)} x_{ijk} + \sum_{(v_i,v_j)\in T} y_{ijk} - \sum_{(v_i,v_j)\in E^+(S)\backslash T} y_{ijk}$$
$$\leq |T| - 1 (S \subseteq V\backslash\{v_1\};\ S \neq \emptyset;\ T \subseteq E^+(S);\ |T| \text{ odd}). \tag{29'}$$

These constraints are valid because $\sum_{(v_i,v_j)\in T} y_{ijk} \leq |T|$ and, if the equality holds,

$$\sum_{(v_i,v_j)\in E(S)} x_{ijk} + \sum_{(v_i,v_j)\in E^+(S)\backslash T} y_{ijk}$$

must be at least 1 in order for the evenness condition to be satisfied because $|T|$ is odd. Belenguer (1990) and Belenguer and Benavent have studied the convex hull of feasible solutions associated with **UCARP** and have derived a number of valid inequalities. Then, using a nonfully automated branch-and-cut procedure, they have solved to optimality two instances of sizes $|V| = 16$, $|A| = 26$ and $|V| = 24$, $|A| = 34$ on an IBM PC/AT.

In addition to obvious relaxations of the above ILP formulation, lower bounds for the undirected **CARP** have been developed by Golden and Wong (1981), Assad, Pearn and Golden (1987), Win (1987), Pearn (1988), as well as Benavent et al. (1992). The latter reference includes theoretical and computational comparisons of all known bounds. All these bounds require the solution of a 1-matching problem on an undirected graph H derived from G. We first describe the Golden and Wong bound in which this idea is first introduced, followed by two of the bounds developed by Benavent et al.

To compute the Golden and Wong bound, first determine a lower bound m on the number of vehicles required in the **CARP** solution. Golden and Wong take $m = \lceil \sum_{(v_i,v_j)\in A} q_{ij}/W \rceil$, but any of the known bin packing lower bounds could be used (see, e.g., Labbé, Laporte and Mercure). Let d_i be the degree of vertex v_i. Now define

$$\lambda = \begin{cases} 2\underline{m} - d_1 & \text{if } d_1 \text{ is even} \\ 2\underline{m} - d_1 - 1 & \text{if } d_1 \text{ is odd.}\end{cases} \tag{31}$$

Then construct the following *matching graph* $H = (U, B)$. The vertex set U is equal to the union of three disjoint sets T, K and L, where $T = \{v_i \in V: d_i \text{ is odd}\}$ and $K = \{k_1, \ldots, k_\lambda\}$, $L = \{\ell_1, \ldots, \ell_\lambda\}$ are two sets of artificial vertices. If $\lambda \leq 0$, then K and L are empty. Define the arcs (u_r, u_s) of B and their associated costs d_{rs} as follows (see Figure 5): If

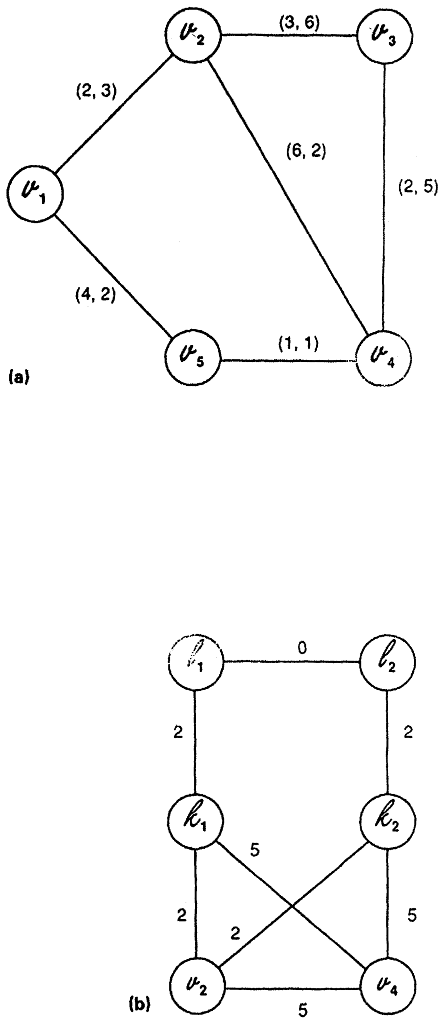


Figure 5. Construction of H from G. (a) Graph G with (c_{ij}, q_{ij}) pairs, $W = 10$, $\underline{m} = 2$. (b) Graph H with costs d_{rs}.

$u_r, u_s \in T$, d_{rs} is the length of a shortest chain in G between $v_r \equiv u_r$ and $v_s \equiv u_s$; if $u_r \in T$ and $u_s \in K$, d_{rs} is the length of a shortest chain in G between $v_r \equiv u_r$ and v_1; if $u_r = k_t \in K$ and $u_s = \ell_t \in L$ for some t, $d_{rs} = \min_{j>1}\{c_{1j}\}$; if $u_r, u_s \in L$, $d_{rs} = 0$.

Let $z(H)$ be the solution value of a minimum cost 1-matching problem on H. Golden and Wong show that

$$LB0 = z(H) + \sum_{\substack{(v_i,v_j)\in A \\ q_{ij}>0}} c_{ij}$$

is a valid lower bound on the value of an optimal **CARP** solution. Note that as in the undirected **CPP**, the matching solution corresponds to a minimum cost augmentation of G by means of artificial arcs to ensure all vertices have even degree. Assad, Pearn and Golden show that the augmented graph derived in that fashion cannot always be decomposed into vehicle tours feasible for the **CARP**. They prove that for the **CCPP**,

$$z^*/LB0 \leqslant \lceil |A| / \lfloor \alpha^{-1} \rfloor \rceil, \qquad (32)$$

where α is the smallest number satisfying $q_{ij} \leqslant \alpha W$. They also illustrate that this bound is tight, i.e., (32) can be satisfied as an equality.

Assad, Pearn and Golden introduce the so-called *node scanning lower bound* (**LB1**) for the **CCPP**. Let d_{ij} denote the length of a shortest chain between v_i and v_j in G, and let d_i again be the degree of v_i. Relabel all vertices in nondecreasing order of d_{1i}. Let $\ell = \min\{k: \sum_{i=2}^{k} d_i \geqslant \lambda\}$, where λ is defined by (32), and reset $d_\ell := \lambda - \sum_{i=2}^{\ell-1} d_i$. Then

$$LB1 = \sum_{\substack{(v_i,v_j)\in A \\ q_{ij}>0}} c_{ij} + \sum_{i=2}^{\ell} d_{1i} d_i$$

is a valid lower bound for the undirected **CCPP**. When $q_{ij} \geqslant W/2$ for all (v_i, v_j), the value of **LB1** coincides with that of the **CCPP** solution. Pearn (1988) shows that **LB0** outperforms **LB1** in most instances, but **LB1** may be expected to perform well on sparse graphs with large demands.

Pearn (1988) describes a lower bounding procedure that combines the principles underlying the computation of **LB0** and **LB1**. Win shows that the worst-case values of the ratios z^*/**LB0** and z^*/**LB1** are of the order of m, the number of vehicles required in the optimal solution, and proposes a number of alternative bounding procedures that dominates **LB0**. Benavent et al. (1992) performed a comparison of all known bounds and developed some new ones, the best one being **LB2**. This bound is theoretically superior to **LB0**, to the Pearn bound, and to the best of the Win bounds. Bound **LB2** as well as the various bounds proposed by Win improve upon **LB0** by considering not only the number of vehicles necessary to cover the whole graph, but also the number of vehicles required for certain subgraphs. First define for every proper subset S of V the cut-set $E(S) = \{(v_i, v_j) \in A: v_i \in S, v_j \in V\backslash S \text{ or } v_i \in V\backslash S, v_j \in S\}$. Also, let T be the subset of all odd vertices among v_1 and vertices of V incident to at least one edge with positive demand.

Lower Bounding Procedure LB2 for the CARP

STEP 1. (**Initialization**). Set $S := \{v_1\}$, $L_1 := L_2 := 0$, and $\bar{c} = \sum_{(v_i,v_j)\in A, q_{ij}>0} c_{ij}$.

STEP 2. (**1-matching solutions**). Set $V' = V\backslash S$ and let G' be the graph induced by V'. Identify the connected components $G'_s = (V'_s, A'_s)$ of $G'(s = 1, \ldots, t)$.

a. For $s = 1, \ldots, t$, compute:

$$q_s := |\{(v_i, v_j) \in E(V'_s): q_{ij} > 0\}|,$$

$$p_s := \left\lceil \left(\sum_{(v_i,v_j)\in A'_s \cup E(V'_s)} q_{ij} \right) \Big/ W \right\rceil,$$

$$r_s := \max\{0, 2p_s - q_s\}, \quad c'_s := \min_{(v_i,v_j)\in E(V'_s)} \{c_{ij}\}.$$

Set $T_s := T \cap V'_s$. If $T_s \neq \varnothing$ or $r_s > 0$, construct the weighted graph H_s (see below), solve a minimum cost

1-matching problem on H_s, with solution value $z(H_s)$. Set $m_s := z(H_s)$.

b. Set $L_1 := \max\{L_1, \bar{c} + L_2 + \sum_{s=1}^{t} m_s\}$ and $L_2 := L_2 + \sum_{s=1}^{t} r_s c'_s$.

STEP 3. **(Updating the working vertex set).** Set $S' := \{v_i \in V: v_i$ is adjacent to a vertex in $S\}$ and $S := S \cup S'$. If $S \neq V$, go to Step 2. Otherwise, set $LB2 := L_1$ and stop.

Graph H_s is constructed as follows. For $v_i \in V'_s$, let c_i be the minimum distance between v_i and any vertex of S, and relabel the vertices of V'_s as $v_{i_1}, v_{i_2}, \ldots$, so that $c_{i_1} \leq c_{i_2} \leq \ldots$. Let h be the least integer satisfying $\sum_{j=1}^{h} d_{i_j} \geq r_s$, where d_{i_j} is the degree of v_{i_j}. Then the vertex set of H_s is equal to $J \cup K \cup T'_s \cup L$, where J is a set of r_s artificial vertices; K contains d_{i_j} copies of vertex $v_{i_j} \in V'_s$ for $j = 1, \ldots, h$; T'_s contains a copy of each vertex of T_s, except for those vertices whose copies are already included in K; L is a set of $\max(0, |T_s| - r_s)$ artificial vertices. The arcs of H_s and their costs are defined as follows: Any arc of $K \cup T'_s$ has a cost equal to the cost of a shortest chain in G between the corresponding vertices, where the cost of an arc between two copies of the same vertex is zero; if $v_k \in K \cup T'_s$ and $v_\ell \in J \cup L$, arc (v_k, v_ℓ) has a cost equal to c_i, where v_i is the vertex of V'_s corresponding to v_k; if $v_k, v_\ell \in L$, arc (v_k, v_ℓ) has a cost of zero.

In addition to **LB2**, Benavent et al. (1992) developed **LB3**, a lower bound only applicable to the case where the number of vehicles in the optimal solution is equal to $\lceil \sum_{(v_i,v_j)\in A} q_{ij}/W \rceil$. Labbé, Laporte and Mercure study the particular case where G is a tree. Consider the subtree T_j rooted at v_j in the opposite direction from v_1 and $b(T_j)$, the optimal value of the bin packing problem associated with the arc demands of T_j. Then

$$\mathbf{LB4} = 2 \sum_{(v_i,v_j)\in A} b(T_j) c_{ij}$$

is a lower bound on the value of the optimal **CARP** solution. Since the bin packing problem is itself NP-hard, the authors replace $b(T_j)$ by a lower bound on its value.

Several heuristic procedures have been proposed for the **CARP**. These can be broadly classified into categories: 1) simple constructive methods; 2) two-phase constructive methods; and 3) improvement methods. We first present five examples of simple constructive heuristics.

Construct-Strike Algorithm

This heuristic was first proposed by Christofides (1973) and later improved by Pearn (1989). The original algorithm gradually constructs feasible cycles and removes them from the graph. When a cycle is constructed, care is taken to ensure that its removal from the graph will not create two disconnected subgraphs. When feasible cycles can no longer be found, an Euler cycle is constructed on the remaining graph, and the search for feasible cycles is repeated. Pearn (1989) modifies this algorithms by lifting the restriction that the graph obtained when removing a cycle remains connected. The complexity of the Christofides heuristic is $O(|A||V|^3)$; in the version developed by Pearn, the complexity increases to $O(|A||V|^4)$.

Path-Scanning Algorithm

In this procedure due to Golden, DeArmon and Baker (1983), feasible cycles are constructed one at a time using a greedy optimality criterion. The entire procedure is repeated with five different criteria and the best solution is selected. Pearn (1989) suggests using at each step a criterion at random, and selecting the best of several solutions generated by this process. Each execution of this procedure requires $O(|V|^3)$ operations.

Augment-Merge Algorithm

This algorithm is also due to Golden, DeArmon and Baker, and is inspired by the Clarke and Wright (1964) algorithm for the vehicle routing problem. Initially, all arcs belong to different cycles. Then, cycles are gradually merged according to a savings criterion. This algorithm has a complexity of $O(|V|^3)$.

Parallel-Insert Algorithm

This algorithm was proposed by Chapleau et al. (1984) for a school bus routing application. It is similar to the path-scanning algorithm, but constructs instead several routes in parallel. According to its authors, this procedure has a lesser tendency to produce zigzag routes that are problematic in a school bus context.

Augment-Insert Algorithm

This procedure draws from the augment-merge and the parallel-insert algorithms, and was suggested by Pearn (1991). In a first phase, all arcs are gradually inserted into feasible cycles connected to the depot, as long as this is possible. Two different criteria are proposed for constructing these cycles: one based on cost, and one based on demand. In a second phase, the remaining arcs are included in the existing cycles, by using a savings criterion. This algorithm can be executed in $O(|V|^3)$ time.

As for the vehicle routing problem (see, e.g., Gillett and Miller 1974, Beasley 1983 as well as Haimovich and Rinnooy Kan 1985), two-phase constructive algorithms for the **CARP** belong to two different categories.

Cluster-First, Route-Second Heuristics

Here the arcs are first partitioned into clusters, each having a total weight not exceeding W. For this, a greedy criterion can be applied (Win), or a generalized assignment algorithm can be used (Benavent et al. 1990). Then a vehicle route is determined for each cluster by a simple modification of a **CPP** algorithm.

Route-First, Cluster-Second Heuristics

Win describes the following procedure. First construct a giant Euler tour over all edges with positive demands.

If these edges induce a connected graph, then this tour can be obtained in polynomial time by a simple modification of a **CPP** algorithm. Otherwise, the problem is an **RPP**, and Frederickson's 1/2-approximation algorithm can be applied. The tour is then partitioned into feasible clusters, a task for which Win suggests using a next fit bin packing heuristic, and a vehicle tour is constructed for each cluster. Ulusoy (1985) constructs an auxiliary graph G^* whose vertices, except the first one, correspond to arcs on the Euler tour, and whose arcs correspond to feasible vehicle tours on G. The least cost set of vehicle tours is then obtained by solving a shortest path problem on G^*.

Post-optimization procedures can be applied to the solution obtained by means of any constructive heuristic. Examples are provided by Ulusoy and by Win. Typically, such improvement methods are inspired by edge exchange heuristics for the **TSP**, see, e.g., Lin and Kerninghan (1973). Win suggests the use of simulated annealing. Tabu search methods (see, e.g., Glover 1989, 1990) could also be applied.

Computational comparisons between the various heuristics are not easy to make. Some heuristics were designed with specific applications in mind, e.g., school bus routing (Chapleau et al. 1984) and explicitly consider a variety of side constraints or objectives. Not all heuristics were tested (e.g., the two-phase procedures of Win), and tests are not always executed on the same problems. It is also difficult to make comparisons between methods that use a single pass, and those that are repeated starting from various starting solutions. This being said, the two papers by Pearn (1989, 1991) provide an interesting comparison of the most popular heuristics. Tests indicate that on dense graphs, the modified construct-strike algorithm proposed by this author yields the best solutions on a series of test problems with $11 \leq |V| \leq 17$, and arc densities between 70% and 100%; it often produces an optimal solution. On sparse graphs with large arc demands, the augment-insert algorithm seems to fare the best. In comparison with other procedures, it produced the best solution on a problems with $13 \leq |V| \leq 27$, and arc densities between 15% and 30%.

6. CONCLUSION

We have provided a two-part survey of the main known results on arc routing problems. The main algorithmic results described in this paper are described in Table I.

Arc routing problems play an important role in distribution management and have been investigated by several researchers. By and large, however, arc routing problems have not received as much attention as node routing problems. The field of arc routing is, to some extent, disorganized and this is due, we believe, to the fact that these problems occur in such a variety of practical problems with different constraints and objectives. As a result, much effort has been devoted to the solution of special cases at the expense of a unifying theory.

A common exact solution methodology for these problems is to formulate them as integer linear programs and apply branch and cut. While this technique has proved highly effective for the **TSP**, it has not met with the same degree of success for **ARPs**. One reason may lie in the structure of the problems themselves. Variables are as a rule integer rather than binary and facets are not so easily identified. Another simple explanation is that **ARPs** have received far less attention than the **TSP**. There may be nice properties just waiting to be discovered.

Approximate algorithms for **ARPs** tend to be ad hoc affairs, often straightforward adaptations of **TSP** heuristics. In recent years, a number of powerful heuristics such as tabu search have been developed for the **VRP**

Table I
Summary of the Main Algorithmic Results for the **RPP**

Problem	Exact Algorithms	Heuristic Algorithms
Undirected **RPP**	NP-hard. Branch-and-bound ILP based algorithm: 24 instances solved ($9 \leq \lvert V\rvert \leq 84$, $13 \leq \lvert A\rvert \leq 184$, $4 \leq \lvert R\rvert \leq 78$). Corberán and Sanchis (1991b) solve 23 of these instances without branching	Heuristic with worst-case ratio of 3/2 (Frederickson 1979)
Directed **RPP**	NP-hard. Branch-and-bound ILP based algorithm (Christofides et al. 1986): 23 instances solved ($13 \leq \lvert V\rvert \leq 80$, $24 \leq \lvert A\rvert \leq 180$, $7 \leq \lvert R\rvert \leq 74$)	Christofides et al. (1986) proposed a heuristic: on 23 instances, the average deviation from optimality was 1.3% and 10 instances were solved optimally
Stacker Crane Problem	NP-hard. No exact algorithm available	The combination of **LARGEARCS** and **SMALLARCS** yields a worst-case ratio of 9/5 (Frederickson, Hecht and Kim 1978)
Capacitated Arc Routing Problem	NP-hard. Branch-and-cut algorithm developed by Belenguer and Benavent (1991). Two instances solved: $\lvert V\rvert = 16$, $\lvert A\rvert = 26$ and $\lvert V\rvert = 24$, $\lvert A\rvert = 34$. Several lower bounds have been proposed	Good heuristics are modified construct-strike for dense graphs (70%–100%) and augment—insert for sparse graphs (15%–30%) (Pearn 1989, 1991)

(Gendreau, Laporte and Potvin 1994). One interesting line of research would be the development of similar methods for **ARP**s.

ACKNOWLEDGMENT

This research was supported in part by the Canadian Natural Sciences and Engineering Research Council, under grants OGP0009160, OGP0038816, and OGP0039682. This support is gratefully acknowledged. Thanks are due to an anonymous referee for his valuable comments.

REFERENCES

ALPRIN, B. S. 1975. A Simulation Approach to Solving the Snow and Ice Removal Problem in an Urban Area. M.Sc. Dissertation, The University of Tulsa, Oklahoma.

ALVAREZ-VALDES, R., E. BENAVENT, V. CAMPOS, A. CORBERÁN, E. MOTA, J. M. TAMARIT AND V. VALLS. 1993. ARC, A Computerised System for Urban Garbage Collection. Working Paper, Departamentó de Estadística e Investigación Operativa, Universidad de Valencia, Spain.

ANGEL, R. D., W. L. CAULDE, R. NOONAN AND A. WHINSTON. 1972. Computer-Assisted School Bus Scheduling. *Mgmt. Sci.* **B18,** 279–288.

ASSAD, A. A., W.-L. PEARN AND B. L. GOLDEN. 1987. The Capacitated Chinese Postman Problem: Lower Bounds and Solvable Cases. *Am. J. Math. and Mgmt. Sci.* **7,** 63–88.

BEASLEY, J. E. 1983. Route First-Cluster Second Methods for Vehicle Routing. *Omega* **11,** 403–408.

BELENGUER, J. M. 1990. *El Poliedro del Problema de Rutas por Arcos con Capacidades*. Ph.D. Thesis, Universidad de Valencia, Spain.

BELENGUER, J. M., AND E. BENAVENT. 1991. Polyhedral Results on the Capacitated Arc Routing Problem. Working Paper, Departamento de Estadística e Investigación Operativa, Universidad de Valencia, Spain.

BELTRAMI, E. L., AND L. D. BODIN. 1974. Networks and Vehicle Routing for Municipal Waste Collection. *Networks* **4,** 65–94.

BENAVENT, E., V. CAMPOS, A. CORBERÁN AND E. MOTA. 1990. The Capacitated Arc Routing Problem. A Heuristic Algorithm. *QÜESTIIÓ* **14,** 107–122.

BENAVENT, E., V. CAMPOS, A. CORBERÁN AND E. MOTA. 1992. The Capacitated Arc Routing Problem. Lower Bounds. *Networks* **22,** 669–690.

BENNETT, B., AND D. GAZIS. 1972. School Bus Routing by Computer. *Trans. Res.* **6,** 317.

BODIN, L. D., AND L. LEVY. 1991. The Arc Partitioning Problem. *Eur. J. Opnl. Res.* **53,** 393–401.

BODIN, L. D., AND O. BERMAN. 1979. Routing and Scheduling of School Buses by Computer. *Trans. Sci.* **13,** 113–129.

BODIN, L. D., AND S. J. KURSH. 1978. A Computer-Assisted System for the Routing and Scheduling of Street Sweepers. *Opns. Res.* **26,** 525–537.

BODIN, L. D., AND S. J. KURSH. 1979. A Detailed Description of a Computer System for the Routing and Scheduling of Street Sweepers. *Comput. & Opns. Res.* **6,** 181–198.

BODIN, L. D., G. FAGIN, R. WELEBNY AND J. GREENBERG. 1989. The Design of a Computerized Sanitation Vehicle Routing and Scheduling System for the Town of Oyster Bay, New York. *Comput. & Opns. Res.* **16,** 45–54.

BOULIANE, J., AND G. LAPORTE. 1992. Locating Postal Relay Boxes Using a Set Covering Algorithm. *Am. J. Math. and Mgmt. Sci.* **12,** 65–74.

BRACA, J., J. BRAMEL, B. POSNER AND D. SIMCHI-LEVI. 1993. A Computerized Approach to the New York City School Bus Routing Project. Working Paper, Columbia University, New York.

BUSACKER, R. G., AND T. L. SAATY. 1965. *Finite Graphs and Networks*. McGraw Hill, New York.

BUSCH, I. K. 1991. Vehicle Routing on Acyclic Networks. Ph.D. Dissertation, The Johns Hopkins University, Baltimore, Md.

CHAPLEAU, L., J. A. FERLAND, G. LAPALME AND J.-M. ROUSSEAU. 1984. A Parallel Insert Method for the Capacitated Arc Routing Problem. *O. R. Lett.* **3,** 95–99.

CHAPLEAU, L., J. A. FERLAND AND J.-M. ROUSSEAU. 1985. Clustering for Routing in Densely Populated Areas. *Eur. J. Opnl. Res.* **20,** 48–57.

CHRISTOFIDES, N. 1973. The Optimum Traversal of a Graph. *Omega* **1,** 719–732.

CHRISTOFIDES, N. 1976. Worst-Case Analysis of a New Heuristic for the Travelling Salesman Problem. Report No 388, Graduate School of Industrial Administration, Carnegie Mellon University, Pittsburgh.

CHRISTOFIDES, N., V. CAMPOS, A. CORBERÁN AND E. MOTA. 1981. An Algorithm for the Rural Postman Problem. Imperial College Report IC.O.R.81.5, London.

CHRISTOFIDES, N., V. CAMPOS, A. CORBERÁN AND E. MOTA. 1986. An Algorithm for the Rural Postman Problem on a Directed Graph. *Math. Prog. Study* **26,** 155–166.

CLARKE, G., AND J. W. WRIGHT. 1964. Scheduling of Vehicles from a Central Depot to a Number of Delivery Points. *Opns. Res.* **12,** 568–581.

CLARK, R. M., AND J. I. GILLEAN. 1975. Analyses of Solid Waste Management Operations in Cleveland, Ohio. *Interfaces* **6** (1, part 2), 32–42.

CLARK, R. M., AND C. H. LEE JR. 1976. Systems Planning for Solid Waste Collection. *Comput. & Opns. Res.* **3,** 157–173.

COOK, T. M., AND B. S. ALPRIN. 1976. Snow and Ice Removal in an Urban Environment. *Mgmt. Sci.* **23,** 227–234.

CORBERÁN, A., AND J. M. SANCHIS. 1991a. A Polyhedral Approach to the Rural Postman Problem. Working Paper, Departamento de Estadística e Investigación Operativa, Universidad de Valencia, Spain.

CORBERÁN, A., AND J. M. SANCHIS. 1991b. On the General Routing Problem Polyhedron: Facets from the GTSP and RPP Polyhedra. Working Paper, Departamento de Estadística e Investigación Operativa, Universidad de Valencia, Spain.

CORNUÉJOLS, G., J. FONLUPT AND D. NADDEF. 1985. The Traveling Salesman on a Graph and Some Related Integer Polyhedra. *Math. Prog.* **33,** 1–27.

DESROSIERS, J., J. A. FERLAND, J.-M. ROUSSEAU, G. LAPALME AND L. CHAPLEAU. 1986. TRANSCOL: A

Multi-Period School Bus Routing and Scheduling System. *TIMS Studies Mgmt. Sci.* **22,** 47–71.

DROR, M., H. STERN AND P. TRUDEAU. 1987. Postman Tour on a Graph With Precedence Relation on Arcs. *Networks* **17,** 283–294.

EDMONDS, J. 1967. Optimum Branching. *J. Res. Natl. Bur. Stand.*, Section B. **71,** 233–240.

EGLESE, R. W., AND L. Y. O. LI. 1992. Efficient Routing for Winter Gritting. *JORSA* **43,** 1031–1034.

EGLESE, R. W., AND H. MURDOCK. 1991. Routing Road Sweepers in a Rural Area. *JORSA* **42,** 281–288.

EISELT, H. A., M. GENDREAU AND G. LAPORTE. 1995. Arc Routing Problems. Part I: The Chinese Postman Problem. *Opns. Res.* **43,** 231–242.

ENGLAND, R. 1982. Computer Analysis Ensures a Clean Sweep. *Surveyor* **6,** 15.

FISCHETTI, M., AND P. TOTH. 1992. An Additive Bounding Procedure for the Asymmetric Travelling Salesman Problem. *Math. Prog.* **53,** 173–197.

FLEISCHMANN, B. 1988. A New Class of Cutting Planes for the Symmetric Traveling Salesman Problem. *Math. Prog.* **40,** 225–246.

FREDERICKSON, G. N. 1979. Approximation Algorithms for Some Routing Problems. *J. Assoc. Comput. Mach.* **26,** 538–554.

FREDERICKSON, G. N., M. S. HECHT AND C. E. KIM. 1978. Approximation Algorithms for Some Postman Problems. *SIAM J. Comput.* **7,** 178–193.

GELDERS, L. F., AND D. G. CATTRYSSE. 1991. Public Waste Collection: A Case Study. *Belgian J. Opns. Res., Statist. and Comput. Sci.* **31,** 3–15.

GELINAS, É. 1992. Le problème du postier chinois avec contraintes générales de préséance. M.Sc.A. Dissertation, École Polytechnique de Montréal, Canada.

GENDREAU, M., G. LAPORTE AND J.-Y. POTVIN. 1994. Metaheuristics for the Vehicle Routing Problem. Publication #963, Centre de recherche sur les transports, Montréal, Canada.

GILLETT, B., AND L. MILLER. 1974. A Heuristic Algorithm for the Vehicle Dispatch Problem. *Opns. Res.* **22,** 340–349.

GLOVER, F. 1989. Tabu Search, Part I. *ORSA J. Comput.* **1,** 190–206.

GLOVER, F. 1990. Tabu Search, Part II. *ORSA J. Comput.* **2,** 4–32.

GOLDEN, B. L., AND R. T. WONG. 1981. Capacitated Arc Routing Problems. *Networks* **11,** 305–315.

GOLDEN, B. L., J. S. DEARMON AND E. K. BAKER. 1983. Computational Experiments With Algorithms for a Class of Routing Problems. *Comput. & Opns. Res.* **10,** 47–59.

GRÖTSCHEL, M., AND O. HOLLAND. 1991. Solution of Large-Scale Symmetric Traveling Salesman Problems. *Math. Prog.* **51,** 141–202.

HAIMOVICH, M., AND A. H. G. RINNOOY KAN. 1985. Bounds and Heuristics for Capacitated Vehicle Routing Problem. *Math. Opns. Res.* **10,** 527–542.

HASLAM, E., AND J. R. WRIGHT. 1991. Application of Routing Technologies to Rural Snow and Ice Control. *Trans. Res. Rec.* No. 1304, 202–211.

HELD, M., AND R. M. KARP. 1971. The Traveling-Salesman Problem and Minimum Spanning Trees: Part II. *Math. Prog.* **1,** 6–25.

LABBÉ, M., G. LAPORTE AND H. MERCURE. 1991. Capacitated Vehicle Routing on Trees. *Opns. Res.* **39,** 616–622.

LAPORTE, G. 1988. Location-Routing Problems. In *Vehicle Routing: Methods and Studies*, B. L. Golden, and A. A. Assad (eds.). North-Holland, Amsterdam, 163–197.

LAPORTE, G. 1992. The Vehicle Routing Problem: An Overview of Exact and Approximate Algorithms. *Eur. J. Opnl. Res.* **59,** 345–358.

LEMIEUX, P. F., AND L. CAMPAGNA. 1984. The Snow Ploughing Problem Solved by a Graph Theory Algorithm. *Civil Eng. Syst.* **1,** 337–341.

LENSTRA, J. K., AND A. H. G. RINNOOY KAN. 1976. On General Routing Problems. *Networks* **6,** 273–280.

LEVY, L. 1987. The Walking Line of Travel Problem: An Application of Arc Routing and Partitioning. Ph.D. Dissertation, University of Maryland, College Park, Md.

LEVY, L., AND L. D. BODIN. 1988. Scheduling the Postal Carriers for the United States Postal Service: An Application of Arc Partitioning and Routing. In *Vehicle Routing: Methods and Studies*, B. L. Golden, and A. A. Assad (eds.). North-Holland, Amsterdam, 359–394.

LEVY, L., AND L. D. BODIN. 1989. The Arc Oriented Location Routing Problem. *INFOR* **27,** 74–94.

LIN, S., AND B. W. KERNIGHAN. 1973. An Effective Heuristic Algorithm for the Traveling-Salesman Problem. *Opns. Res.* **21,** 498–516.

LUKKA, A., AND L. SALMINEN. 1987. Comparison of Three Heuristics for an Arc Routing Problem. Research Report 6/1987, Department of Information Technology, Lappeenranta University of Technology, Finland.

MALE, J. W., AND J. C. LIEBMAN. 1978. Districting and Routing for Solid Waste Collection. *J. Environ. Eng. Div.* **104,** EEI.

MCBRIDE, R. 1982. Controlling Left and U-Turns in the Routing of Refuse Collection Vehicles. *Comput. & Opns. Res.* **9,** 145–152.

MINAZZATO, E. 1988. La Raccolta dei Rifiuti Solidi Urbani Nella Città di Como: Modelli e Algoritmi di Ricerca Operativa. Tesi di Laurea in Scienze Statistiche ed Economiche, Università degli Studi di Padova, Italy.

NEGREIROS, M. 1974. O Problema de Planejamento e Percuso de Veículos na Coleta do Lixo Urbano Domiciliar. M.Sc. Dissertation, Sistemas/COPPE, Federal University of Rio de Janeiro, Brazil.

ORLOFF, C. S. 1974. A Fundamental Problem in Vehicle Routing. *Networks* **4,** 35–64.

PADBERG, M. W., AND M. GRÖTSCHEL. 1985. Polyhedral Computations. In *The Traveling Salesman Problem: A Guided Tour of Combinatorial Optimization*, E. L. Lawler, J. K. Lenstra, A. H. G. Rinnooy Kan, and D. B. Shmoys (eds.). John Wiley, Chichester, U.K., 307–360.

PADBERG, M. W., AND G. RINALDI. 1991. A Branch-and-Cut Algorithm for the Resolution of Large-Scale Symmetric Traveling Salesman Problems. *SIAM Rev.* **33,** 60–100.

PEARN, W.-L. 1984. The Capacitated Chinese Postman Problem. Ph.D. Thesis, University of Maryland, College Park, Md.

PEARN, W.-L. 1988. New Lower Bounds for the Capacitated Arc Routing Problem. *Comput. & Opns. Res.* **16,** 589–600.

PEARN, W.-L. 1989. Approximate Solutions for the Capacitated Arc Routing Problem. *Comput. & Opns. Res.* **16,** 589–600.

PEARN, W.-L. 1991. Augment-Insert Algorithms for the Capacitated Arc Routing Problem. *Comput. & Opns. Res.* **18,** 189–198.

PEARN, W.-L., A. A. ASSAD AND B. L. GOLDEN. 1987. Transforming Arc Routing into Node Routing Problems. *Comput. & Opns. Res.* **14,** 285–288.

ROSENKRANTZ, D. J., R. E. STEARNS AND P. M. LEWIS III. 1977. An Analysis of Several Heuristics for the Traveling Salesman Problem. *SIAM J. Comput.* **6,** 563–581.

ROY, S., AND J.-M. ROUSSEAU. 1989. The Capacitated Canadian Postman Problem. *INFOR* **27,** 58–73.

SAHNI, S. K., AND T. GONZALEZ. 1976. P-Complete Approximations Problems. *J. Assoc. Comput. Mach.* **23,** 555–565.

SANCHIS, J. M. 1990. *El Poliedro del Problema del Cartero Rural*. Ph.D. Thesis, Universidad de Valencia, Spain.

STERN, H., AND M. DROR. 1979. Routing Electric Meter Readers. *Comput. & Opns. Res.* **6,** 209–223.

STRICKER, R. 1970. Public Sector Vehicle Routing: The Chinese Postman Problem. M.Sc. Dissertation, Department of Electrical Engineering, Massachusetts Institute of Technology, Cambridge, Mass.

SWERSEY, A. J., AND W. BALLARD. 1984. Scheduling School Buses. *Mgmt. Sci.* **30,** 844–853.

TUCKER, W. B., AND G. M. CLOHAN. 1979. Computer Simulation of Urban Snow Removal. *Trans. Research Board Special Research Report No 185*. 293–302.

TURNER, W., AND E. HOUGLAND. 1975. The Optimal Routing of Solid Waste Collection. *AIIE Trans.* **7,** 427–431.

ULUSOY, G. 1985. The Fleet Size and Mix Problem for Capacitated Arc Routing. *Eur. J. Opnl. Res.* **22,** 329–337.

WIN, Z. 1987. Contributions to Routing Problems. Doctoral Dissertation, Universität Augsburg, Germany.

WUNDERLICH, J., M. COLLETTE, L. LEVY AND L. D. BODIN. 1992. Scheduling Meter Readers for Southern California Gas Company. *Interfaces* **22**(3), 22–30.

WYSKIDA, R., AND J. GUPTA. 1973. IE's Improve City's Solid Waste Collection. *Indust. Engin.* **46,** 12–15.