Les Cahiers du GERAD ISSN: 0711–2440

A Branch-and-Cut Algorithm for the Undirected Capacitated Arc Routing Problem

G. Ghiani, D. Laganà,
G. Laporte, r. Musmanno

G–2007–39

Mai 2007

A Branch-and-Cut Algorithm for the Undirected Capacitated Arc Routing Problem

Gianpaolo Ghiani
Dipartimento di Ingegneria dell'Innovazione
Università di Lecce, Strada per Arnesano, 73100 Lecce, Italy
gianpaolo.ghiani@unile.it

Demetrio Laganà
Dipartimento di Elettronica, Informatica e Sistemistica
Università della Calabria, 87036 Rende (CS), Italy
lagana@hpcc.unical.it

Gilbert Laporte
GERAD and Canada Research Chair in Distribution Management
HEC Montréal, Montréal, Canada H3T 2A7
gilbert@crt.umontreal.ca

Roberto Musmanno
Dipartimento di Elettronica, Informatica e Sistemistica
Università della Calabria, 87036 Rende (CS), Italy
musmanno@unical.it

Mai 2007

Les Cahiers du GERAD

G–2007–39

Abstract

The *Capacitated Arc Routing Problem* (CARP) consists of determining a set of least cost capacitated vehicle routes servicing a set of arcs. In this paper the undirected CARP is formulated as a pure binary linear integer problem. Valid inequalities are generated and the problem is solved by branch-and-cut. All the benchmark instances proposed by DeArmon and Benavent et al. can be solved optimally without any branching, for the first time ever.

Keywords: Capacitated arc routing problem, integer programming, branch-and-cut.

Résumé

Le problème non orienté de tournées sur les arcs avec capacités (PTAC) consiste à déterminer un ensemble de tournées de véhicules de moindre coût desservant un ensemble d'arcs. Dans cet article, le PTAC est formulé comme un programme linéaire binaire. On génère des inégalités valides et on résout le problème par séparation et coupes. Tous les problèmes tests de DeArmon et de Benavent et al. sont résolus à l'optimalité pour la première fois.

Mots clés : problème de tournées sur les arcs avec capacités, programmation en nombres entiers, séparation et coupes.

Acknowledgments: This research was partially supported by the Ministero dell'Istruzione, dell'Università e della Ricerca Scientifica (MIUR) and by the Center of Excellence on High-Performance Computing, University of Calabria, Italy, and by the Canadian Natural Sciences and Engineering Research council under grant 39682-05. This support is gratefully acknowledged.

1 Introduction

The purpose of this article is to present a new formulation and a branch-and-cut algorithm for the undirected *Capacitated Arc Routing Problem* (CARP) defined as follows. Let $G(V, E)$ be an undirected graph, where $V = \{0, 1, \ldots, n\}$ is the vertex set and E is the edge set. Vertex 0 represents the depot at which are based m identical vehicles of capacity Q. A subset $R \subseteq E$ of edges are *required*, i.e., they must be *serviced* by a vehicle, but any edge of E can be *traversed or deadheaded* any number of times. Each edge has a non-negative cost (or length) c_e. In addition, each required edge has a nonnegative weight (or demand) d_e. The CARP is to design a set of least cost vehicle routes such that each route starts and ends at the depot, each required edge is serviced by exactly one vehicle, and the total demand serviced by any vehicle does not exceed Q. The CARP was introduced by Golden and Wong (1981). It is NP-hard since it includes as a special case the *Rural Postman Problem* (RPP), shown to be NP-hard by Lenstra and Rinnooy Kan (1976). Applications of the CARP arise in garbage collection, snow removal, street sweeping and gritting, mail delivery, meter reading, school bus routing, etc.

While the CARP is a central problem in arc routing and has been known for a long time, it is still almost exclusively tackled by means of heuristics. As far as we are aware, the only available exact method for the CARP is a parallel branch-and-bound algorithm proposed by Hirabayashi et al. (1992a) and Kiuchi et al. (1995) in which a lower bound is computed at each node through a node duplication lower bounding procedure (Hirabayashi et al., 1992b). This enumerative algorithm was capable of instances ranging from 15 edges, with an average of 35.2 nodes in the search tree, to 50 edges, with an average of 124.5 nodes. Transformations of the CARP into an equivalent vertex routing problem (namely the *Capacitated Vehicle Routing Problem*, CVRP) have been proposed by Pearn, Assad and Golden (1987), Longo, Poggi de Aragão and Uchoa (2006), and Baldacci and Maniezzo (2006). The first two procedures require three CVRP vertices for each required arc of the CARP, while the third method needs only two CVRP vertices. By using a state-of-the-art CVRP algorithm, Baldacci and Maniezzo (2006) were able to improve the lower bounds obtained for a number of classical test problems, and could solve some instances involving up to 98 required edges. Various lower bounds have been developed by Benavent et al. (1992), Amberg and Voß (2002), Wøhlk (2003), and Belenguer and Benavent (1998, 2003). In addition, Welz (1994) has proposed valid inequalities and separation procedures for the directed version of the CARP. In recent years, the advent of metaheuristics has given rise to a new generation of powerful algorithms. These include those of Hertz, Laporte and Mittaz (2000), Hertz and Mittaz (2001), Beullens et al. (2003), Lacomme, Prins and Ramdane-Cherif (2004), Doerner et al. (2004), and Brandão and Eglese (2006). The best methods are based on tabu search, variable neighbourhood search, and memetic search. We refer the reader to Eiselt, Gendreau and Laporte (1995), to Assad and Golden (1995), and to the book of Dror (2000), for a detailed description of early lower bounds and heuristics. Polyhedral studies of the CARP and other arc routing problems are reviewed in Eglese and Letchford (2000) and in Benavent Corberán and Sanchis (2000).

In this paper we formulate the undirected CARP as a pure binary linear integer program, and we introduce four families of new valid inequalities, which are embedded within a branch-and-cut algorithm. The remainder of this article is organized as follow. In Section 2, we present the new formulation and the valid inequalities. We then describe the branch-and-cut algorithm in Section 3, and computational results in Section 4. Conclusions follow in Section 5.

2 Formulation

Our linear formulation is inspired from a non-linear model described by Belenguer and Benavent (1998). Let x_e^k be a binary decision variable equal to 1 if and only if vehicle k services the required edge $e \in R$, and let y_e^k be an integer variable equal to the number of times vehicle k deadheads edge $e \in E$. Given a subset of vertices $S \subseteq V \backslash \{0\}$, let $E(S)$ be the subset of edges with both endpoints in S, $E_R(S) = E(S) \cap R$ the subset of required edges with both endpoints in $S, \delta(S) = \{(i,j) : i \in S, j \notin S \text{ or } i \notin S, j \in S\}$ the cutset associated with S, and $\delta_R(S) = \delta(S) \cap R$ the cutset of required edges associated with S. In addition, for any subset of edges $E' \subseteq E$ and for every subset of required edges $R' \subseteq R$, let $x^k(R') = \sum_{e \in R'} x_e^k$ and $y^k(E') = \sum_{e \in E'} y_e^k$ for a given vehicle k. The Belenguer and Benavent (1998) formulation is as follows:

$$\text{Minimize } \sum_{k=1}^{m} \sum_{e \in E} c_e x_e^k + \sum_{k=1}^{m} \sum_{e \in R} c_e y_e^k \tag{1}$$

subject to

$$\sum_{k=1}^{m} x_e^k = 1 \qquad (e \in R) \tag{2}$$

$$\sum_{e \in R} d_e x_e^k \leq Q \qquad (k = 1, \ldots, m) \tag{3}$$

$$x^k(\delta_R(S)) + y^k(\delta(S)) \geq 2x_f^k \qquad (S \subseteq V \backslash \{0\}, f \in E_R(S), k = 1, \ldots, m) \tag{4}$$

$$x^k(\delta_R(S)) + y^k(\delta(S)) = 0 \mod (2) \qquad (S \subseteq V \backslash \{0\}, k = 1, \ldots, m) \tag{5}$$

$$x_e^k \in \{0, 1\} \qquad (e \in R, k = 1, \ldots, m) \tag{6}$$

$$y_e^k \geq 0 \text{ and integer} \qquad (e \in E, k = 1, \ldots, m), \tag{7}$$

where the objective function (1) is the total traversal cost, constraints (2) state that every required edge must be serviced, constraints (3) impose the vehicle capacity is never exceeded and constraints (4) ensure that the solution is connected. Finally, parity constraints (5) stipulate that each route induces an Eulerian subgraph. Unfortunately, these constraints are non-linear. Belenguer and Benavent (1998) propose a *relaxation* of their formulation in which constraints (5) are substituted with the following valid inequalities:

$$x^k(\delta_R(S)\backslash H) + y^k(\delta(S)) \geq x^k(H) - |H| + 1 \quad (S \subseteq V, H \subseteq \delta_R(S), |H| \text{ odd }). \tag{8}$$

2.1 CARP reformulation

We introduce a linear formulation for the undirected CARP in terms of the x_e^k and y_e^k variables. Our model exploits the fact that in a CARP optimal solution each vehicle route can be seen as an optimal RPP route spanning the required edges of the route. Hence the RPP dominance relations described in Christofides et al. (1981) can be reformulated as:

$$y_e^k \leq 2 \qquad (e \in E\backslash R, k = 1, \ldots, m), \tag{9}$$

$$y_e^k + x_e^k \leq 2 \qquad (e \in R, k = 1, \ldots, m). \tag{10}$$

Indeed, on the basis of the results reported in Ghiani and Laporte (2000), only a small number of the inequalities (9)–(10) are satisfied as equalities in an optimal CARP solution. These dominance relations allow us to replace each integer valued y_e^k variables with a pair of binary variables $y_e^{'k}$ and $y_e^{''k}$, such that:

$$y_e^k = y_e^{'k} + y_e^{''k} \qquad (e \in E, k = 1, \ldots, m), \tag{11}$$

$$y_e^{'k} \in \{0, 1\} \qquad (e \in E, k = 1, \ldots, m), \tag{12}$$

$$y_e^{''k} \in \{0, 1\} \qquad (e \in E, k = 1, \ldots, m). \tag{13}$$

Using these variables, we can now express the parity constraints (5) as *cocircuit inequalities* (Barahona and Grötschel, 1986; Ghiani and Laporte, 2000):

$$\sum_{e \in \delta_R(S)\backslash F} x_e^k + \sum_{e \in \delta(S)\backslash F'} y_e^{'k} + \sum_{e \in \delta(S)\backslash F''} y_e^{''k} \geq \sum_{e \in F} x_e^k + \sum_{e \in F'} y_e^{'k} + \sum_{e \in F''} y_e^{''k} - |F| - |F'| - |F''| + 1$$

$$(S \subseteq V, F \subseteq \delta_R(S), F' \subseteq \delta(S), F'' \subseteq \delta(S), |F| + |F'| + |F''| \text{ odd}, k = 1, \ldots, m). \tag{14}$$

Constraints (14) state that if an odd subset $F \cup F' \cup F''$ of serviced and deadheaded edges are incident to a set of vertices S, then at least another edge of the cutset has to be serviced or traversed.

2.2 Equivalent solutions

The new CARP formulation (1)–(6), (11)–(14) yields a large number of equivalent solutions. First, since all vehicles have the same capacity, for any given solution $m!$ equivalent solutions can be obtained through a permutation of the vehicle indices. We introduce additional constraints to remove this redundancy. Let σ be a permutation of the set I_R of the indices corresponding to the required edges $e \in R$, and let $\sigma(i) = j \in I_R$. Define $i(k) = \min\{i \in N : x^k_{e_{\sigma(i)}} = 1\}$ as the smallest index of the required edges serviced by vehicle k. To impose the condition

$$i(1) \leq \ldots \leq i(m), \tag{15}$$

we use the following set of *fixing constraints*:

$$x^1_{e_{\sigma(1)}} = 1, \tag{16}$$

$$x^k_{e_{\sigma(i)}} \leq \sum_{j=1,\ldots,i-1} x^{k-1}_{e_{\sigma(j)}}, \qquad (k = 3, \ldots, m, i \geq 2), \tag{17}$$

$$x^k_{e_{\sigma(i)}} = 0, \quad i = 1, \ldots, m-1, \qquad (k = i+1, \ldots, m). \tag{18}$$

Constraints (16) state that vehicle with index 1 must serve edge $e_{\sigma(1)}$. Constraints (17) stipulate that if a required edge $e_{\sigma(i)}$ is serviced by vehicle k ($k = 3, \ldots, m$) then at least one edge $e_{\sigma(j)}$, $j = 1, \ldots, i-1$, must be serviced by the vehicle with the index immediately preceding k. Finally, constraints (18) state that edges $e_{\sigma(i)}$ ($i = 1, \ldots, m-1$), cannot be serviced by vehicles with indices larger than $k = i+1, \ldots, m$.

To illustrate, let $m = 3$, $I_R = \{3, 5, 10, 21, 7, 35, 46\}$, $\{\sigma(1) = 46, \sigma(2) = 10, \sigma(3) = 3, \sigma(4) = 7, \sigma(5) = 5, \sigma(6) = 21, \sigma(7) = 35\}$ and consider the three routes ($e^1_{\sigma(2)}$, $e^1_{\sigma(7)}$, $e^1_{\sigma(1)}$), ($e^2_{\sigma(3)}, e^2_{\sigma(6)}$) and ($e^3_{\sigma(5)}, e^3_{\sigma(4)}$). Then $i(1) = \min\{2, 7, 1\} = 1$, $i(2) = \min\{3, 6\} = 3$ and $i(3) = \min\{5, 4\} = 4$. Therefore $i(1) \leq i(2) \leq i(3)$ and inequalities (16)–(18) become:

$$\begin{aligned}
x^1_{e_{46}} &= 1;\\
x^3_{e_{10}} &\leq x^2_{e_{46}};\\
x^3_{e_3} &\leq x^2_{e_{46}} + x^2_{e_{10}};\\
x^3_{e_7} &\leq x^2_{e_{46}} + x^2_{e_{10}} + x^2_{e_3};\\
x^3_{e_5} &\leq x^2_{e_{46}} + x^2_{e_{10}} + x^2_{e_3} + x^2_{e_7};\\
x^3_{e_{21}} &\leq x^2_{e_{46}} + x^2_{e_{10}} + x^2_{e_3} + x^2_{e_7} + x^2_{e_5};\\
x^3_{e_{35}} &\leq x^2_{e_{46}} + x^2_{e_{10}} + x^2_{e_3} + x^2_{e_7} + x^2_{e_5} + x^2_{e_{21}}.
\end{aligned}$$

Moreover, $x^k_{e_{\sigma(i)}} = 0$ for $i = 1, 2$ and $k = 2, 3$.

2.3 Surrogate valid inequalities

Our branch-and-cut algorithm uses two classes of constraints, introduced by Belenguer and Benavent (1998), which impose conditions on the aggregated variables $z_e = \sum_{k=1}^{m} (y'^k_e + y''^k_e)$,

$e \in E$. Given a set $S \subseteq V \backslash \{0\}$, define $D(S) = \sum_{e \in E_R(S) \cup \delta_R(S)} d_e$, and observe that at least $\lceil D(S)/Q \rceil$ vehicles are required to service the demand $D(S)$, and any vehicle that services some edge in the set $E_R(S) \cup \delta_R(S)$ will have to cross $\delta(S)$ at least twice. Therefore, any feasible solution contains at least $2\lceil D(S)/Q \rceil - |\delta_R(S)|$ deadheading edges in the cutset $\delta(S)$. This condition is expressed by the following *surrogate capacity constraints*:

$$\sum_{k=1}^{m} \sum_{e \in \delta(S)} (y_e^{\prime k} + y_e^{\prime\prime k}) \geq 2\lceil D(S)/Q \rceil - |\delta_R(S)| \qquad \big(S \subseteq V \backslash \{0\}\big). \tag{19}$$

We now introduce new valid inequalities based on a basic property of Eulerian graphs, namely that such graphs must be *even*, i.e., all their vertices must have an even degree. The graph associated with any feasible solution for the undirected CARP must be an even and therefore, any edge cutset must contain an even number of edges. In particular, if any edge cutset has an odd number of required edges, then at least one edge in the cutset must be deadheaded. This condition is expressed by the following *odd edge cutset constraints*:

$$\sum_{k=1}^{m} \sum_{e \in \delta(S)} (y_e^{\prime k} + y_e^{\prime\prime k}) \geq 1 \quad (S \subseteq V, |\delta_R(S)| \text{ odd }). \tag{20}$$

3 Branch-and-cut algorithm

In this section we describe a branch-and-cut algorithm based on the CARP formulation defined by (1)–(6), and (11)–(19).

Step 1 (First node of the search tree)
Let $\overline{z}$ be an upper bound on the optimal solution value z^*. Define a first subproblem as a linear program containing the objective function (1), constraints (2) and (3), the fixing constraints (16)–(18) generated for the permutation σ, with $\sigma(i) \leq \sigma(j)$ for each $i \leq j$, a connectivity constraint for each single component, a cocircuit constraint with $F \cup F' \cup F'' = \emptyset$ for each R-odd vertex. Insert this subproblem in a list.

Step 2 (Termination test)
If the list is empty, stop. Otherwise select a subproblem from the list according to the smallest lower bound strategy.

Step 3 (Subproblem solution)
Solve the subproblem and let z be the solution value. If $z \geq \bar{z}$, remove the current subproblem from the list and go to Step 2. If the solution is feasible for the CARP and $z < \bar{z}$, set $\bar{z} := z$, remove the current subproblem from the list, and go to Step 2. Otherwise, go to Step 4.

Step 4 (Cut generation)
Identify as many violated inequalities as possible. If no inequality is generated, go to Step 5, otherwise add the violated inequalities to the current subproblem, and go to Step 3.

Step 5 (Branching)
Branch on the fractional variable x_e^k, $y_e^{'k}$ or $y_e^{''k}$ nearest to 0.5 and generate the corresponding subproblems. Insert the subproblems in the list and go to Step 2.

In Step 1, we use the TSA upper bound of Brandão and Eglese (2006), which empirically outperforms all upper bounds obtained by means of recent heuristics.

For Step 4 we have developed the following separation procedures. Let $(\bar{x}, \bar{y}', \bar{y}'')$ be the current solution of the subproblem selected in Step 3, with $\bar{x} \in \Re^{m|R|}$, $\bar{y}' \in \Re^{m|E|}$ and $\bar{y}'' \in \Re^{m|E|}$, the separation problems for the constraints (4), (14), (19), and (20) solved as follows.

Connectivity inequalities (4)**.** While the separation problem is solvable in $O(|V|^3)$ time, we use a modification of the heuristic proposed by Fischetti, Salazar and Toth (1997). For each vehicle k, we construct a maximum spanning tree on an auxiliary graph $\bar{G}^k$ where each connected component C_h^k of R^k is represented by a vertex h'^k, and edges $e'^k = (h'^k, t^k)$ have a cost equal to the sum of variables x_e^k, $y_e^{'k}$ and $y_e^{''k}$ corresponding to edges $e^k = (i^k, j^k)$ such that $i^k \in V_h^k$ and $j^k \in V_t^k$. At any stage of the construction of this tree, let S^k be the set of connected components of R^k corresponding to vertices of the partial tree. If S^k yields a violated connectivity constraint, this constraint is generated. Once the spanning tree is complete, another check for violated connectivity constraints is made by removing in turn each edge of the tree.

Cocircuit inequalities (14)**.** This separation problem can be solved in a polynomial time by using the heuristic procedure proposed by Ghiani and Laporte (2000) for the undirected RPP. The slack of (14) can be expressed as:

$$\sum_{e\in\delta_R(S)\setminus F} \bar{x}_e^k + \sum_{e\in\delta(S)\setminus F'} \bar{y}_e^{'k} + \sum_{e\in\delta(S)\setminus F''} \bar{y}_e^{''k} + \sum_{e\in F}(1-\bar{x}_e^k) + \sum_{e\in F'}(1-\bar{y}_e^{'k}) + \sum_{e\in F''}(1-\bar{y}_e^{''k}) - 1.$$

To minimize this quantity, include in F every edge e such that $\bar{x}_e^k \geq 0.5$, in F' every edge e such that $\bar{y}_e^{'k} \geq 0.5$ and in F'' every edge e such that $\bar{y}_e^{''k} \geq 0.5$. If $|F| + |F'| + |F''|$ is odd and the associated slack is negative, then constraint (14) is violated by the current solution. Otherwise, no constraint (14) is violated.

Surrogate capacity inequalities (19)**.** This separation problem can be solved in polynomial time whenever $\sum_{k=1}^{m} \sum_{e\in\delta(S)} (y_e^{'k} + y_e^{''k}) \geq 2D(S)/Q - |\delta_R(S)|$, by applying a modification of the procedure proposed by Belenguer and Benavent (2003). We solve a maximum flow-minimum cut problem on an auxiliary graph $\tilde{G}$ constructed from G by adding a dummy vertex $n+1$, connected to every vertex $i \in V$. The capacity u_e of every edge e of $\tilde{G}$ is defined as follows:

$$u_e = \begin{cases} \bar{z}_e = \sum_{k=1}^{m} \left(\bar{y}_e^{\prime k} + \bar{y}_e^{\prime\prime k} \right) & \text{if } E \backslash R. \\ \bar{z}_e + 1 - d_e/Q & \text{if } e \in R. \\ \sum_{f \in \delta_R(\{v_i\})} d_f/Q & \text{if } e = (i,j), i \in \{1,\ldots,n\} \text{ and } j = n+1. \end{cases}$$

Using a reasoning similar to that of Belenguer and Benavent (2003), we observe that a minimum cut separating the depot and vertex $n+1$ corresponds to a cut edgeset which induces a partition of $V \cup \{n+1\}$ into $\{\bar{V}, V \cup \{n+1\} \backslash \bar{V}\}$ such that $0 \in \bar{V}$ and $n+1 \in V \cup \{n+1\} \backslash \bar{V}$. Also, given a set $S \subseteq V \backslash \{0\}$, the capacity of the cutset $\delta(S \cup \{n+1\})$, minus the constant $2\sum_{e \in R} d_e/Q$, is equal to the slack of constraint (19). It follows that if the maximum flow from 1 to $n+1$ minus $2\sum_{e \in R} d_e/Q$ is negative, then constraint (19) corresponding to a minimum cutset $\delta(S^*)$ is violated by the current solution. Otherwise, no constraint (19) is violated.

Odd edge cutset inequalities (20)**.** We have applied the $O(|V|^4)$ exact procedure of Padberg and Rao (1982). Given surrogate variables $\bar{z}_e = \sum_{k \in I}(\bar{y}_e^{\prime k} + \bar{y}_e^{\prime\prime k})$, let $G(\bar{z})$ be the graph induced by edges $e \in E$ such that $\bar{z}_e > 0$. We define $\bar{z}_e$ as the capacity of edge $e \in E$, and label as odd the vertices incident with an odd number of required edges of G. An *odd cutset* is defined as an edge cutset $\delta(S)$ such that S contains an odd number of odd vertices. Using the algorithm of Padberg and Rao (1982) we determine the minimum capacity odd cut in $G(\bar{z})$. If the capacity of this cut is less than 1 then constraint (20) is violated by the current solution. Otherwise, no constraint (20) is violated.

4 Computational results

The algorithm was coded in C using the Microsoft Visual Studio C++ Environment and the CPLEX library. It was executed on a PC with a Pentium IV processor clocked at 2 GHz. We used CPLEX 9.0 to solve the linear programs. The algorithm was tested on two sets of the CARP benchmark instances introduced by DeArmon (1981) and Benavent et al. (1992). Computational results are provided in Tables 1 and 2. The column headings are defined as follows:

$\|V\|$	:	number of vertices of the graphs;
$\|E\|$	:	number of edges (all required);
$\overline{z}$	:	initial upper bound;
$\underline{z}$	:	lower bound at the root of the search tree;
CONNECT	:	number of generalized connectivity inequalities;
COCIRCUIT	:	number of cocircuit inequalities;
AGGRCAP	:	number of aggregate capacity inequalities;
ODDEDGE	:	number of odd edge cutset inequalities;
z^*	:	optimal solution value;
$\underline{z}/z^*$	:	lower bound at the root node of the search tree, divided by the optimal solution value;
Nodes	:	number of nodes in the search tree;
Seconds	:	CPU time in seconds.

Our computation times do not include one time required to compute the upper bound $\overline{z}$. According to Brandão and Eglese (2006), this time is on average 2.5 seconds for the DeArmon instances, and 20.2 seconds for the Benavent et al. (1992) instances, on a Pentium Mobile (1.4 GHz).

Our results indicate that the proposed algorithm can solve, for the first time and without any branching, all the DeArmon (1981) and Benavent (1992) CARP instances. The lower bound $\underline{z}$ generated at the root node of the search tree is always equal to the optimum z^*. It is also never less than the best of the available lower bounds: CPA (Belenguer and Benavent, 1998), LB2 (Benavent et al., 1992), NDLB (Kinchi et al., 1992), and DWMLB (Longo et al., 2006). The best of these bounds is often equal to $\overline{z}$ so that there is then no need to execute our algorithm. However, we decided not to use these bounds in order to test and illustrate the strength of our lower bounding procedure. It can be seen that all four types of valid inequalities played a role in the solution process, but the connectivity and cocircuit inequalities were the most useful. The instances for which optimal solution were identified for the first time are boldmarked. Solutions are available at: `www.deis.unical.it/deis1.0/portale/home/musmanno`.

5 Conclusions

The undirected CARP is a hard combinatorial optimization problem with several applications in the field of distribution management. We have described a new linear formulation that provides a full description of the CARP in terms of binary variables. A fully automated branch-and-cut algorithm was then developed and implemented. This algorithm is capable of solving to optimality all instances of two benchmark sets, for the first time ever and without any branching.

Table 1: Computational results for the DeArmon (1981) instances

Instances	$\|V\|$	$\|E\|$	$\overline{z}$	$\underline{z}$	**CONNECT**	**COCIRCUIT**	**AGGRCAP**	**ODDEDGE**	z^*	$\underline{z}/z^*$	**Nodes**	**Seconds**
1	12	22	316	316	58	169	1	1	316	1.000	1	5.73
2	12	26	339	339	69	279	1	1	339	1.000	1	10.28
3	12	22	275	275	71	97	0	0	275	1.000	1	2.86
4	11	19	287	287	17	78	1	2	287	1.000	1	1.58
5	13	26	377	377	70	196	1	2	377	1.000	1	10.12
6	12	22	298	298	52	145	1	4	298	1.000	1	3.25
7	12	22	325	325	60	158	0	5	325	1.000	1	3.98
10	27	46	**348**	**348**	771	919	11	34	**348**	1.000	1	472.80
11	27	51	303	303	1194	1028	6	37	303	1.000	1	230.66
12	12	25	275	275	43	137	1	4	275	1.000	1	3.14
13	22	45	395	395	201	323	0	4	395	1.000	1	22.64
14	13	23	**458**	**458**	122	318	2	11	**458**	1.000	1	16.41
15	10	28	536	536	58	215	0	4	536	1.000	1	6.06
16	7	21	100	100	85	218	1	24	100	1.000	1	283.26
17	7	21	58	58	63	182	0	16	58	1.000	1	54.68
18	8	28	127	127	41	98	1	8	127	1.000	1	4.76
19	8	28	91	91	52	301	1	10	91	1.000	1	6.72
20	9	36	164	164	93	204	1	0	164	1.000	1	206.52
21	11	11	55	55	20	7	0	0	55	1.000	1	0.33
22	11	22	121	121	7	0	1	7	121	1.000	1	0.11
23	11	33	156	156	52	267	1	4	156	1.000	1	9.03
24	11	44	200	200	113	534	1	7	200	1.000	1	49.11
25	11	55	233	233	506	68	1	27	233	1.000	1	35.48

Table 2: Computational results for the Benavent et al. (1992) instances

Instances	$\|V\|$	$\|E\|$	$\overline{z}$	$\underline{z}$	**CONNECT**	**COCIRCUIT**	**AGGRCAP**	**ODDEDGE**	z^*	$\underline{z}/z^*$	**Nodes**	**Seconds**
1.A	24	39	173	173	66	99	0	27	173	1.000	1	2.67
1.B	24	39	173	173	142	265	1	23	173	1.000	1	12.06
1.C	24	39	**245**	**245**	497	469	0	21	**245**	1.000	1	55.95
2.A	24	34	227	227	62	46	3	15	227	1.000	1	1.75
2.B	24	34	259	259	99	202	3	21	259	1.000	1	8.34
2.C	24	34	457	457	540	684	4	16	457	1.000	1	140.81
3.A	24	35	81	81	93	98	2	26	81	1.000	1	3.20
3.B	24	35	87	87	153	269	4	24	87	1.000	1	17.05
3.C	24	35	138	138	388	294	3	13	138	1.000	1	78.1
4.A	41	69	400	400	380	544	4	29	400	1.000	1	82.22
4.B	41	69	412	412	432	708	3	25	412	1.000	1	166.25
4.C	41	69	428	428	56	553	2	19	428	1.000	1	138.89
4.D	41	69	**530**	**530**	1396	975	1	4	**530**	1.000	1	402.83
5.A	34	65	423	423	249	588	3	31	423	1.000	1	58.20
5.B	34	65	446	446	381	723	2	26	446	1.000	1	135.53
5.C	34	65	**474**	**474**	485	730	2	23	**474**	1.000	1	202.40
5.D	34	65	**577**	**577**	1208	1278	0	32	**577**	1.000	1	756.31
6.A	31	50	223	223	195	250	2	31	223	1.000	1	29.25
6.B	31	50	**233**	**233**	316	408	3	22	**233**	1.000	1	32.30
6.C	31	50	**317**	**317**	467	145	0	3	**317**	1.000	1	222.94
7.A	40	66	279	279	341	415	0	25	279	1.000	1	51.61
7.B	40	66	283	283	433	431	2	16	283	1.000	1	79.14
7.C	40	66	**334**	**334**	2157	1513	5	53	**334**	1.000	1	4169.44
8.A	30	63	386	386	209	507	1	21	386	1.000	1	45.45
8.B	30	63	395	395	303	555	3	12	395	1.000	1	86.02
8.C	30	63	**521**	**521**	529	440	0	2	**521**	1.000	1	556.01
9.A	50	92	323	323	509	817	1	51	323	1.000	1	190.06
9.B	50	92	326	326	580	698	1	28	326	1.000	1	229.26
9.C	50	92	332	332	724	940	0	10	332	1.000	1	597.42
9.D	50	92	**391**	**391**	2583	5007	19	13	**391**	1.000	1	495.64
10.A	50	97	428	428	157	233	2	18	428	1.000	1	648.92
10.B	50	97	436	436	482	543	5	11	436	1.000	1	776.83
10.C	50	97	446	446	876	769	3	35	446	1.000	1	1025.58
10.D	50	97	**526**	**526**	2148	1099	2	33	**526**	1.000	1	4865.07

References

1. Amberg, A., S. Voß. 2002. A hierarchical relaxation lower bound for the capacitated arc routing problem. *Proceedings of the 35th Annual Hawaii International Conference on System Sciences*.

2. Assad, A. A., B. L. Golden. 1995. Arc routing methods and applications. M.O. Ball, T.L. Magnanti, C.L. Monma, G.L. Nemhauser, eds. *Network Routing*, Handbooks in Operations Research and Management Science. North-Holland, Amsterdam, 375–483.

3. Baldacci, R., V. Maniezzo. 2006. Exact methods based on node-routing formulations for undirected arc-routing problems. *Networks* **47** 52–60.

4. Barahona, F., M. Grötschel. 1986. On the cycle polytope of a binary matroid. *Journal of Combinatorial Theory* **40** 40–62.

5. Belenguer, J. M., E. Benavent. 1998. The capacitated arc routing problem: Valid inequalities and facets. *Computational Optimization and Applications* **10** 165–187.

6. Belenguer, J. M., E. Benavent. 2003. A cutting plane algorithm for the capacitated arc routing problem. *Computers & Operations Research* 705–728.

7. Benavent, E., V. Campos, A. Corberán, E. Mota. 1992. The capacitated arc routing problem: Lower bounds. *Networks* **22** 669–690.

8. Benavent E., A. Corberán, J. M. Sanchis. 2000. Linear programming based methods for solving arc routing problems. M. Dror, ed. *Arc routing. Theory, Solutions and Applications.* Kluwer, Boston, 231–75.

9. Beullens, P., L. Muyldermans, D. Cattrysse, D. Van Oudheusden. 2003. A guided local search heuristic for the capacitated arc routing problem. *European Journal of Operational Research* **147** 629–643.

10. Brandão, J., R. Eglese. 2006. A deterministic tabu search algorithm for the capacitated arc routing problem. *Computers & Operations Research*, forthcoming.

11. Christofides, N., V. Campos, A. Corberán, E. Mota. 1981. An Algorithm for the Rural Postman Problem. Imperial College Report IC.O.R. 81.5.

12. DeArmon, J. S. 1981. A Comparison of Heuristics for the Capacitated Chinese Postman Problem. Master's Thesis, University of Maryland, College Park, MD.

13. Doerner, K.F., R.F. Hartl, V. Maniezzo, M. Reimann. 2004. Applying ant colony optimization to the capacitated arc routing problem. M. Dorigo, M. Birattari, C. Blum, L. Gambardella, F. Mondada, eds. *Ant Colony Optimization and Swarm Intelligence*. Springer, Berlin, Lecture Notes in Computer Science **3172** 420–421.

14. Dror, M., editor. 2000. *Arc routing. Theory, Solutions and Applications.* Kluwer, Boston.

15. Eglese, R. W., A. Letchford. 2000. Polyhedral theory for arc routing problems. M. Dror, ed. *Arc routing. Theory, Solutions and Applications.* Kluwer, Boston, 199–230.

16. Eiselt, H. A., M. Gendreau, G. Laporte. 1995. Arc routing problems, part II: The rural postman problem. *Operations Research* **43** 399–414.

17. Fischetti, M., J. J. Salazar, P. Toth. 1997. A branch-and-cut algorithm for the symmetric generalized traveling salesman problem. *Operations Research* **45** 378–394.

18. Ghiani, G., G. Laporte. 2000. A branch-and-cut algorithm for the Undirected Rural Postman Problem. *Mathematical Programming* **87** 467–481.

19. Golden, B. L., R. T. Wong. 1981. Capacitated arc routing problems. *Networks* **11** 305–315.

20. Golden, B. L., J. S. DeArmon, E. K. Baker. 1983. Computational experiments with algorithms for a class of routing problems. *Computers & Operations Research* **10** 47–59.

21. Hertz, A., G. Laporte, M. Mittaz. 2000. A tabu search heuristic for the capacitated arc routing problem. *Operations Research* **48** 129–135.

22. Hertz, A., M. Mittaz. 2001. A variable neighbourhood descent algorithm for the undirected capacitated arc routing problem. *Transportation Science* **35** 425–434.

23. Hirabayashi, R., Y. Saruwatari, N. Nishida. 1992a. Tour construction algorithm for the capacitated arc routing problem. *Asia–Pacific Journal of Operational Research* **9** 155–175.

24. Hirabayashi, R., Y. Surawatari, N. Nishida. 1992b. Node Duplication Lower Bounds for the Capacitated Arc Routing Problem. *Journal of the Operations Research Society of Japan* **35** 119–133.

25. Kiuchi, M., Y. Shinano, R. Hirabayashi, Y. Saruwatari. 1995. An exact algorithm for the capacitated arc routing problem using parallel branch and bound method. Abstract of the 1995 Spring National Conference of the Operational Research Society of Japan, 28–29.

26. Lacomme, P., C. Prins, W. Ramdane-Cherif. 2004. Competitive memetic algorithms for arc routing problems. *Annals of Operations Research* **131** 159–185.

27. Lenstra, J. K., A. H. G. Rinnooy Kan. 1976. On general routing problems. *Networks* **6** 273–280.

28. Longo, H., M. Poggi de Aragão, E. Uchoa. 2006. Solving capacitated arc routing problems using a transformation to the CVRP. *Computers & Operations Research* **33** 1823–1837.

29. Padberg, M.W., M. R. Rao. 1982. Odd minimum cut-sets and b-matchings. *Mathematics of Operations Research* **7** 67–80.

30. Pearn W. L., A. Assad, B. L. Golden. 1987. Transforming arc routing into node routing problems. *Computers & Operations Research* **14** 285–288.

31. Welz, S. A. 1994. Optimal solutions for the capacitated arc routing problem using integer programming. Ph.D. Dissertation, Department of Quantitative Analysis and Operations Management, University of Cincinnati.

32. Wøhlk, S. 2006. New lower bounds for the capacitated arc routing problem. *Computers & Operations Research* **33** 3458–3472.