ELSEVIER

Discrete Applied Mathematics 123 (2002) 487–512

DISCRETE
APPLIED
MATHEMATICS

Models, relaxations and exact approaches for the capacitated vehicle routing problem

Paolo Toth, Daniele Vigo*

Università di Bologna, Dipartimento di Elettronica, Informatica e Sistemistica, Viale Risorgimento 2, 40136 Bologna, Italy

Received 28 September 1999; accepted 2 October 2000

Abstract

In this paper we review the exact algorithms based on the branch and bound approach proposed in the last years for the solution of the basic version of the vehicle routing problem (VRP), where only the vehicle capacity constraints are considered. These algorithms have considerably increased the size of VRPs that can be solved with respect to earlier approaches. Moreover, at least for the case in which the cost matrix is asymmetric, branch and bound algorithms still represent the state-of-the-art with respect to the exact solution. Computational results comparing the performance of different relaxations and algorithms on a set of benchmark instances are presented. We conclude by examining possible future directions of research in this field. © 2002 Elsevier Science B.V. All rights reserved.

Keywords: Vehicle routing problem; Exact algorithms; Branch and bound; Relaxations

1. Introduction

The vehicle routing problem (VRP) is one of the most studied among the combinatorial optimization problems, due both to its practical relevance and to its considerable difficulty.

The VRP is concerned with the determination of the optimal routes used by a fleet of vehicles, based at one or more depots, to serve a set of customers. Many additional requirements and operational constraints are imposed on the route construction in practical applications of the VRP. For example, the service may involve both deliveries and collections, the load along each route must not exceed the given capacity of the

* Corresponding author.
E-mail address: dvigo@deis.unibo.it (D. Vigo).

PII: S0166-218X(01)00351-1

vehicles, the total length of each route must not be greater than a prescribed limit, the service of the customers must occur within given time windows, the fleet may contain heterogeneous vehicles, precedence relations may exist between the customers, the customer demands may not be completely known in advance, the service of a customer may be split among different vehicles, and some problem characteristics, as the demands or the travel times, may vary dynamically.

We consider the static and deterministic basic version of the problem, known as the *capacitated* VRP (CVRP). In the CVRP all the customers correspond to deliveries, the demands are deterministic, known in advance and may not be split, the vehicles are identical and are based at a single central depot, only the capacity restrictions for the vehicles are imposed, and the objective is to minimize the total cost (i.e., the number of routes and/or their length or travel time) needed to serve all the customers. Generally, the travel cost between each pair of customer locations is the same in both directions, i.e., the resulting cost matrix is *symmetric*, whereas in some applications, as the distribution in urban areas with one-way directions imposed on the roads, the cost matrix is *asymmetric*.

The CVRP has been extensively studied since the early sixties and in the last years many new heuristic and exact approaches were presented. The largest problems which can be consistently solved by the most effective exact algorithms proposed so far contain about 50 customers, whereas larger instances may be solved only in particular cases. So instances with hundreds of customers, as those arising in practical applications, may only be tackled with heuristic methods.

The CVRP extends the well-known *Traveling Salesman Problem* (TSP), calling for the determination of the circuit with associated minimum cost, visiting exactly once a given set of points. Therefore, many exact approaches for the CVRP were inherited from the huge and successful work done for the exact solution of the TSP.

Laporte and Nobert [32] presented an extensive survey which was entirely devoted to exact methods for the VRP and gave a complete and detailed analysis of the state of the art up to the late eighties. The aim of the present work is to provide an update of that survey, describing the algorithms recently proposed for the exact solution of CVRP both for the case with symmetric and asymmetric cost matrices. Up to the end of the last decade the most effective exact approaches for the CVRP were mainly branch and bound algorithms using basic relaxations, as the assignment problem and the shortest spanning tree. Recently, more sophisticated bounds were proposed, as those based on Lagrangian relaxations or on the additive approach, which increased the size of the problems that can be solved to optimality by branch and bound. Moreover, following the success obtained by branch and cut methods for the TSP, encouraging results were obtained by using these algorithms for the CVRP.

In this work we treat separately problems with symmetric and asymmetric cost matrices. In fact, although the symmetric problems are special cases of the asymmetric ones, the latter were much less studied in the literature and the exact methods developed for them have in general a poor performance when applied to symmetric instances. Analogously, not all the approaches proposed for symmetric problems may be directly adapted to solve also asymmetric ones. In the following, we will denote with SCVRP and ACVRP the symmetric and asymmetric CVRP, respectively. Moreover, when the explicit distinction between the two versions is not needed, we simply use CVRP.

Other surveys covering exact algorithms, but often mainly devoted to heuristic methods, were presented by Christofides et al. [10], Magnanti [37], Bodin et al. [4], Christofides [7], Laporte [29], Fisher [22], Toth and Vigo [45] and Golden et al. [25]. An annotated bibliography was recently proposed by Laporte [30], whereas an extensive bibliography was presented by Laporte and Osman [34]. A book on the subject was edited by Golden and Assad [24].

The work is organized as follows. In Section 2 we give a detailed description of CVRP as a graph theoretic problem, and introduce the corresponding notation. In Section 3 we consider the more general case of ACVRP, where the cost matrix is asymmetric, illustrating the branch and bound algorithms proposed by Laporte et al. [31] and by Fischetti et al. [18]. In Section 4 we examine the exact methods proposed for the more widely studied SCVRP. In particular, we discuss the basic relaxations based on K-tree and b-matching and their strengthening in a Lagrangian fashion proposed by Fisher [20] and Miller [40], respectively. We also briefly discuss the set-partitioning based relaxations used by Hadjcostantinou et al. [26]. Computational results comparing the performance of different relaxations and algorithms on a set of benchmark instances are presented. Finally, in Section 5, we draw some conclusions and discuss future directions of research.

The information about the performance, expressed in Mflops, of the computers used for testing the algorithms presented are taken (when available) from Dongarra [15]. Moreover, all the computational results reported in this paper are performed by using well-known test instances from the literature. As proposed in Vigo [48], the instances are identified with a name whose first character denotes the problem type (A, E and S for asymmetric, Euclidean and other symmetric problems, respectively), then the name includes the number of vertices, depot included, and the number of available vehicles, and the last letter indicates the source of the instance. For example, E051-05e is the famous 50 customers problem Euclidean described in Christofides and Eilon [8].

2. Problem definition and notation

The CVRP may be defined as the following graph theoretic problem. Let $G=(V,A)$ be a complete graph where $V=\{0,\dots,n\}$ is the vertex set and A is the arc set. Vertices $j=1,\dots,n$ correspond to the customers, each with a known nonnegative *demand*, d_j, to be delivered, whereas vertex 0 corresponds to the depot (with a fictitious demand $d_0=0$). Given a customer set $S \subseteq V$, let $d(S)=\sum_{j\in S} d_j$ denote the total demand of the set.

A nonnegative *cost*, c_{ij} is associated with each arc $(i,j)\in A$ and represents the *travel cost* spent to go from vertex i to vertex j. Generally, the use of the loop arcs, (i,i), is not allowed and this is imposed by defining $c_{ii}=+\infty$ for all $i\in V$. If the cost matrix is asymmetric, A is a set of directed arcs and the corresponding problem is called *asymmetric* CVRP (ACVRP). Otherwise, i.e., when $c_{ij}=c_{ji}$ for all $i,j\in V$, the problem is called *symmetric* CVRP (SCVRP) and the arc set A is often replaced by a set of undirected edges, E. In the following, we denote the undirected edge set of graph G by A when edges are indicated by means of their endpoints $(i,j), i,j \in V$, and by E when edges are indicated through a single index e. Given a vertex set $S \subset V$, let $\delta(S)$ and $\sigma(S)$ denote the set of edges $e \in E$ (or arcs $(i,j) \in A$) which have only one or both endpoints in S, respectively. As usual, when single vertices $i \in V$ are considered, we write $\delta(i)$ rather than $\delta(\{i\})$.

Graph G is generally assumed to be *complete* (i.e., it includes the arcs connecting all the vertex pairs, possibly with the exception of loops) since this simplifies the notation. If this is not the case, a complete graph may be easily obtained by assigning an infinite cost value to nonexisting arcs.

In several practical situations the cost matrix satisfies the *triangle inequality*, $c_{ik} + c_{kj} \geqslant c_{ij}$ for all $i,j,k \in V$. In this case, it is not convenient to deviate from the direct link between two vertices i and j. The respect of the triangle inequality is sometimes required by the algorithms for CVRP. In such case, if the original instance does not satisfy the triangle inequality, and equivalent instance may be obtained in an immediate way by adding a suitably large positive quantity M to the cost of each arc. However, the drastic distortion of the metric induced by this operation may produce very bad solutions with respect to the original costs, mainly for what concerns the effectiveness of heuristic algorithms. If G is strongly connected but not complete, it is possible to obtain a complete graph where the cost of each arc (i,j) is defined as the cost of the shortest path from i to j, computed on the original graph. Note that in this case the complete graph satisfies the triangle inequality, therefore this may be seen also as a method for "triangularizing" complete graphs. Moreover, in some instances the vertices are associated with points of the plane with given coordinates and the cost c_{ij}, for all the arcs $(i,j) \in A$, is defined as the Euclidean distance between the two points corresponding to vertices i and j. In this case the cost matrix is symmetric and satisfies the triangle inequality, and the resulting problem is often called *Euclidean* CVRP. Observe that the frequently performed rounding to the nearest integer of the real-valued Euclidean arc costs may cause a violation of the triangular inequality, whereas this does not happen if the costs are rounded up.

A set of K identical vehicles, each with capacity C, is available at the depot. Each vehicle may perform at most one route, and we assume that K is not smaller than $K_{\min}$, where $K_{\min}$ is the minimum number of vehicles needed to serve all the customers. The value of $K_{\min}$ may be determined by solving the *bin packing problem* (BPP) associated with the CVRP, calling for the determination of the minimum number of bins, each with capacity C, required to load all the n items, each with nonnegative weight $d_j, j = 1,\dots,n$. In spite of the fact that BPP is NP-hard in the strong sense, instances with hundreds of items can be optimally solved very effectively (see, e.g., Martello and Toth [38]). In the following, given a set $S \subseteq V \setminus \{0\}$, we denote by $\gamma(S)$ the minimum number of vehicles needed to serve all the customers in S, i.e., the optimal solution value of the BPP with item set S. Note that $\gamma(V \setminus \{0\}) = K_{\min}$. Often $\gamma(S)$ is replaced by the so-called *continuous* lower bound for BPP: $\lceil d(S)/C \rceil$. Moreover, to ensure feasibility we assume that $d_j \leqslant C$ for each $j = 1,\dots,n$.

The CVRP consists of finding a collection of K simple *circuits* (corresponding to vehicle routes) with minimum cost, defined as the sum of the costs of the arcs belonging to the circuits, and such that:

(i) each circuit visits vertex 0, i.e., the depot vertex;
(ii) each vertex $j \in V \setminus \{0\}$ is visited by exactly one circuit;
(iii) the sum of the demand of the vertices visited by a circuit does not exceed the vehicle capacity, C.

Several variants of the basic versions of CVRP have been considered in the literature. First of all, when the number K of available vehicles is greater than $K_{\min}$, it may be possible to leave some vehicle unused, thus requiring to determine *at most* K circuits. In this case, fixed costs are often associated with the use of the vehicles. This may be included in the CVRP by adding the constant value representing the fixed cost associated with the use of a vehicle, to the cost of the arcs leaving the depot.

In practical situations the additional objective requiring the minimization of the number of used circuits (i.e., vehicles) is frequently present. Normally, the algorithms proposed in the literature do not consider this objective explicitly, however, depending on the characteristics of the algorithm used, there are different ways to take it into account. When the algorithm allows for the determination of solutions using a number of circuits smaller than K, this objective may be easily included by adding a large constant value to the cost of the arcs leaving the depot. Thus, the optimal solution first minimizes the number of arcs leaving the depot (hence the number of circuits), then the cost of the other used arcs. If, as normally happens, the algorithm determines only solutions using all the K available vehicles, there are two possibilities. The first one is to compute $K_{\min}$ by solving the BPP associated with CVRP, and then to apply the algorithm with $K = K_{\min}$. The second possibility is to define an extended instance with a complete graph $\bar{G} = (\bar{V}, \bar{A})$ obtained from G by adding $K - K_{\min}$ dummy vertices to V, each with demand $d_j = 0$. Let $W = \{n+1, \ldots, n+K-K_{\min}\}$ be the set of these dummy vertices, the cost $\bar{c}_{ij}$ of the arcs $(i,j) \in \bar{A}$ is defined as

$$\bar{c}_{ij} := \begin{cases} c_{ij} & \text{for } i,j \in V; \\ 0 & \text{for } i = 0,\ j \in W; \\ 0 & \text{for } i \in W,\ j = 0; \\ c_{0j} & \text{for } i \in W,\ j \in V \setminus \{0\}; \\ M & \text{for } i \in V \setminus \{0\},\ j \in W; \\ M & \text{for } i \in W,\ j \in W; \end{cases} \tag{1}$$

where M is a very large positive number. The optimal solution of the CVRP computed on the extended instance may contain "empty" routes made up by single dummy vertices. Note that by adding a large constant to $\bar{c}_{0j}$, $j \in W$, the number of empty routes is maximized, i.e., the number of used vehicles is minimized.

Note that, even in the case for which the triangle inequality holds, the minimization of the number of used circuits does not correspond, in general, to the minimization of the total cost of the circuits. On the other hand, solutions forced to use exactly K circuits (with $K > K_{\min}$) do not lead, in general, to the minimum total cost.

The CVRP is known to be NP-hard (in the strong sense), and generalizes the well-known Traveling Salesman Problem, arising when $C \geqslant d(V)$ and $K = K_{\min} = 1$.

Therefore, all the relaxations proposed for the TSP are valid for the CVRP. As already mentioned, the CVRP is also related to the bin packing problem.

3. The asymmetric CVRP

In this section we examine the CVRP with asymmetric cost matrix (ACVRP). Two different basic modeling approaches have been proposed for the VRP in the literature. The models of the first type, known as *vehicle flow formulations*, use integer variables, associated with each arc or edge of the graph, which count the number of times that the arc or edge is traversed by a vehicle. These are the most frequently used models for the basic versions of VRP. The linear programming relaxation of vehicle flow models can be very weak when the capacity constraints are tight.

The models of the second type have an exponential number of binary variables, each associated with a different feasible circuit. The VRP is then formulated as a *set partitioning problem* (SPP) calling for the determination of a collection of circuits with minimum cost, which serves each customer once. The corresponding linear programming relaxation is typically much tighter than in the previous models. Note, however, that these models generally require dealing with a very large number of variables. An example of SPP-based model for SCVRP is given in Section 4.4.

The integer linear programming model we describe for ACVRP is a two-index vehicle flow formulation which uses $O(n^2)$ binary variables x, to indicate if a vehicle traverses or not an arc in the optimal solution. In other words, variable x_{ij} takes value 1 if arc $(i,j) \in A$ belongs to the optimal solution, and value 0 otherwise.

$$\text{(VRP1)} \quad \min \sum_{i \in V} \sum_{j \in V} c_{ij} x_{ij} \tag{2}$$

$$\text{s.t.} \quad \sum_{i \in V} x_{ij} = 1 \quad \text{for all } j \in V \setminus \{0\} \tag{3}$$

$$\sum_{j \in V} x_{ij} = 1 \quad \text{for all } i \in V \setminus \{0\} \tag{4}$$

$$\sum_{i \in V} x_{i0} = K \tag{5}$$

$$\sum_{j \in V} x_{0j} = K \tag{6}$$

$$\sum_{i \notin S} \sum_{j \in S} x_{ij} \geqslant \gamma(S) \quad \text{for all } S \subseteq V \setminus \{0\},\ S \neq \emptyset \tag{7}$$

$$x_{ij} \in \{0,1\} \quad \text{for all } i,j \in V. \tag{8}$$

The *indegree* and *outdegree* constraints (3) and (4) impose that exactly one arc enters and leaves each vertex associated with a customer, respectively. Analogously, constraints (5) and (6) impose the degree requirements for the depot vertex. Note that one arbitrary constraint among the $2|V|$ constraints (3)–(6) is actually implied by the remaining $2|V|-1$ ones, hence it can be removed. The so-called *capacity-cut* constraints (7) impose both the connectivity of the solution and the vehicle capacity requirements. In fact, they stipulate that each cut $(V \setminus S, S)$ defined by a vertex set S is crossed by a number of arcs not smaller than $\gamma(S)$ (minimum number of vehicles needed to serve set S). The capacity-cut constraints remain valid also if $\gamma(S)$ is replaced by the continuous lower bound for BPP (see, e.g., [12]).

Observe that, when $|S|=1$ or $S=V\setminus\{0\}$ the capacity-cut constraints (7) are weakened versions of the corresponding degree constraints (3)–(6). Note also that, because of the degree constraints (3)–(6), we have

$$\sum_{i \notin S} \sum_{j \in S} x_{ij} = \sum_{i \in S} \sum_{j \notin S} x_{ij} \quad \text{for all } S \subseteq V \setminus \{0\},\ S \neq \emptyset, \tag{9}$$

in other words, each cut $(V \setminus S, S)$ is crossed in both directions the same number of times. From (9) we may also re-state (7) as

$$\sum_{i \notin S} \sum_{j \in S} x_{ij} \geqslant \gamma(V \setminus S) \quad \text{for all } S \subset V,\ \{0\} \in S. \tag{10}$$

An alternative formulation may be obtained by transforming the capacity-cut constraints (7), by means of the degree constraints (3)–(6), into the well-known *generalized subtour elimination* constraints (GSEC):

$$\sum_{i \in S} \sum_{j \in S} x_{ij} \leqslant |S| - \gamma(S) \quad \text{for all } S \subseteq V \setminus \{0\},\ S \neq \emptyset, \tag{11}$$

which impose that at least $\gamma(S)$ arcs leave each vertex set S.

Both families of constraints (7) and (11) have a cardinality growing exponentially with n. A possible way to partially overcome this drawback is to consider only a limited subset of these constraints. This can be done by relaxing them in a Lagrangian fashion as done in [20] and in [40] (see Section 4.3) or by explicitly including them in the linear programming relaxation as done in branch and cut approaches.

Alternatively, an equivalent family of constraints with polynomial cardinality may be obtained by considering the subtour elimination constraints proposed for the TSP by Miller et al. [39], and extending them to CVRP (see, e.g., [10] and [28]):

$$u_i - u_j + C x_{ij} \leqslant C - d_j \quad \text{for all } i, j \in V \setminus \{0\},\ i \neq j, \text{ s.t. } d_i + d_j \leqslant C, \tag{12}$$

$$d_i \leqslant u_i \leqslant C \quad \text{for all } i \in V \setminus \{0\}, \tag{13}$$

where u_i, $i \in V \setminus \{0\}$, is an additional continuous variable representing the load of the vehicle after visiting customer i. It is easy to see that constraints (12)–(13) impose the capacity requirements of CVRP. In fact, when $x_{ij}=0$ the constraint is not binding since $u_i \leqslant C$ and $u_j \geqslant d_j$, whereas when $x_{ij}=1$ they impose that $u_j \geqslant u_i + d_j$. These constraints may be strengthened by lifting some coefficients as illustrated by Desrochers and Laporte [13].

Two exact algorithms, both based on the branch and bound approach were proposed for ACVRP so far. The algorithm described by Laporte et al. [31] uses a lower bound based on the *Assignment Problem* (AP) relaxation of ACVRP. The algorithm proposed by Fischetti et al. [18] combines, according to the so-called *additive approach*, the AP lower bound with a lower bound based on disjunction and one based on a min-cost flow relaxation. These bounds are briefly described in this section.

Other bounds for the ACVRP may be derived by generalizing the methods proposed for the symmetric case. For example, Fisher [20] proposed a way to extend to ACVRP the bounds based on K-tree he derived for the SCVRP (described in Sections 4.1 and 4.3). In this extension the Lagrangian problem calls for the determination of an undirected K-tree on the undirected graph obtained by replacing each pair of directed arcs (i,j) and (j,i) with a single edge (i,j) with cost $c'_{ij} = \min\{c_{ij}, c_{ji}\}$. No computational testing for this bound was presented in Fisher [20]. Possibly better bounds may be obtained by explicitly considering the asymmetry of the problem, i.e., by using K-arborescences rather than K-trees and by strengthening the bound in a Lagrangian fashion as proposed by Fisher for the CVRP (see [43,44] for an application to the capacitated shortest spanning arborescence problem, and to the VRP with backhauls, respectively).

3.1. The assignment lower bound

Carpaneto and Toth [6], and Laporte et al. [31] proposed to relax model VRP1 by dropping the capacity-cut constraints (7). The resulting relaxation, i.e. (2)–(6) and (8), is a *transportation problem* (TP), calling for a min-cost collection of circuits of G visiting once all the vertices in $V \setminus \{0\}$, and K times vertex 0. This solution can be infeasible for ACVRP since:

(i) the total customer demand on a circuit can exceed the vehicle capacity;
(ii) there may exist circuits not visiting vertex 0.

The solution of TP requires $O(n^3)$ time through a transportation algorithm. In practice, it is more effective to transform the problem into an *assignment problem* (AP) defined on the extended complete digraph $G' = (V', A')$, where $V' := V \cup W'$ and $W' = \{n+1, \ldots, n+K-1\}$ contains $K-1$ additional copies of vertex 0, and the cost c'_{ij} of each arc in A' is defined as follows:

$$c'_{ij} := \begin{cases} c_{ij} & \text{for } i,j \in V \setminus \{0\}; \\ c_{i0} & \text{for } i \in V \setminus \{0\},\ j \in W'; \\ c_{0j} & \text{for } i \in W',\ j \in V \setminus \{0\}; \\ \lambda & \text{for } i,j \in W'; \end{cases} \tag{14}$$

where $\lambda = M \gg 1$. After this transformation, constraint (5) may be replaced by K constraints of type (3), one for each copy of the depot. Analogously, constraint (6) may be replaced by K constraints of type (4). This extension was originally proposed by Lenstra and Rinnooy Kan [35], to transform into an ordinary TSP the m-TSP, which calls for the determination of a collection of m circuits visiting m times a distinguished vertex (i.e., the depot) and once all the remaining vertices. Observe that by defining λ in a different way we obtain an alternative transformation, with respect to that presented in Section 2, to obtain solutions using less than K vehicles. In particular, defining $\lambda = 0$ leads to the determination of the min-cost set of *at most* K routes, whereas defining $\lambda = -M$ leads to the determination of the min-cost set of $K_{\min}$ routes.

3.2. The bounds based on arborescences

In analogy with what is done for the SCVRP (see Section 4.1) another basic relaxation is that based on the solution of degree-constrained spanning arborescences. This relaxation may be obtained from model VRP1 by:

(i) removing the outdegree constraints (4) for all the customer vertices;
(ii) weakening the capacity-cut constraints (7) so as to impose only the connectivity of the solution, i.e. by replacing the right-hand side with 1.

The resulting relaxed problem, called K-shortest spanning arborescence problem (KSSA) is defined by

$$\text{(KSSA)} \quad \min \sum_{i \in V} \sum_{j \in V} c_{ij} x_{ij} \tag{15}$$

$$\text{s.t.} \quad \sum_{i \in V} x_{ij} = 1 \quad \text{for all } j \in V \setminus \{0\}, \tag{16}$$

$$\sum_{i \in V} x_{i0} = K, \tag{17}$$

$$\sum_{j \in V} x_{0j} = K, \tag{18}$$

$$\sum_{i \notin S} \sum_{j \in S} x_{ij} \geqslant 1 \quad \text{for all } S \subseteq V \setminus \{0\},\ S \neq \emptyset, \tag{19}$$

$$x_{ij} \in \{0, 1\} \quad \text{for all } i, j \in V. \tag{20}$$

The KSSA can be effectively solved by considering two separate subproblems:

(i) the determination of a min-cost spanning arborescence with outdegree K at the depot vertex, defined by (15), (16), (18)–(20), with variables x_{ij} for $i \in V$, $j \in V \setminus \{0\}$, and
(ii) the determination of a set of K min-cost arcs entering the depot, defined by (15), (17), and (20), with variables x_{0i} for $i \in V$.

The KSSA can be determined in $O(n^2)$ since the first subproblem can be solved in $O(n^2)$ time (see, [23,43]), while the second subproblem clearly requires $O(n)$ time.

The above described lower bound was never used within branch and bound algorithms and the preliminary computational results discussed in Section 3.5 show that its quality is generally poor and inferior to that of the AP lower bound. However, it should be mentioned that for a problem closely related to the CVRP, as the VRP with backhauls, Toth and Vigo [44] successfully used a Lagrangian relaxation based on the solution of KSSAs, solving problems with up to 100 customers.

3.3. The disjunctive lower bound

The following two bounds were proposed by Fischetti et al. [18]. The first bound is based on a disjunction on infeasible arc subsets, whereas the second bound is based on a min-cost flow relaxation.

A given arc subset $B \subset A$ is called *infeasible* if no feasible solution to ACVRP can use all its arcs, i.e., when

$$\sum_{(a,b)\in B} x_{ab} \leqslant |B| - 1 \tag{21}$$

is a valid inequality for ACVRP. For any given (minimal) infeasible arc subset $B \subset A$, the following logical disjunction holds for each $x \in F$, where F is the set of all the ACVRP feasible solutions:

$$\bigvee_{(a,b)\in B} (x \in Q^{ab} := \{x \in \mathfrak{R}^A : x_{ab} = 0\}). \tag{22}$$

Then $|B|$ restricted problems are defined, each denoted as RP^{ab} and including the additional condition $x_{ab} = 0$ imposed for a different $(a,b) \in B$. For each RP^{ab}, a valid lower bound, ϑ^{ab}, is computed through the AP relaxation of the previous section (with $c_{ab} := +\infty$ to impose $x_{ab} = 0$). The disjunctive bound

$$L_{\mathrm{D}} := \min\{\vartheta^{ab} : (a,b) \in B\}, \tag{23}$$

clearly dominates the AP lower bound, L_{AP}, since $\vartheta^{ab} \geqslant L_{\mathrm{AP}}$ for all $(a,b) \in B$.

A possible way to determine infeasible arc subsets B is the following. First solve the AP relaxation with no additional constraints, and store the corresponding optimal solution $(x^*_{ij} : i,j \in V)$. If x^* is feasible for ACVRP, then clearly the lower bound L_{AP} cannot be improved. Otherwise, try to improve it by using a disjunction on a suitable infeasible arc subset B. Note that imposing $x_{ab} = 0$ for any $(a,b) \in A$ such that $x^*_{ab} = 0$ would produce $\vartheta^{ab} = L_{\mathrm{AP}}$, hence a disjunctive bound $L_{\mathrm{D}} = L_{\mathrm{AP}}$. Therefore, B is chosen as a subset of $A^* := \{(i,j) \in A : x^*_{ij} = 1\}$, if any, corresponding to one of the following cases:

(i) a circuit which is disconnected from the depot vertex,
(ii) a path such that the total demand of the associated customer vertices exceeds C,
(iii) a feasible circuit which leaves uncovered a set of customers, S, whose total demand cannot be served by the remaining $K-1$ vehicles, i.e., such that $\gamma(S) > K-1$.

Different choices of the infeasible arc subset B lead to different lower bounds. Therefore, Fischetti, Toth and Vigo [18] used an overall bounding procedure, called ADD_DISJ, based on the *additive approach* which considers, in sequence, different infeasible arc subsets so as to produce a possibly better overall lower bound.

The additive approach was proposed by Fischetti and Toth [16] and allows for the combination of different lower bounding procedures, each exploiting different substructures of the considered problem. When applied to a minimization problem, each procedure returns a lower bound ρ and a *residual cost matrix*, $\tilde{c}$, such that:

$$\tilde{c} \geqslant 0$$

$$\rho + \tilde{c}x \leqslant cx \quad \text{for all } x \in F.$$

The entries of $\tilde{c}$ represent lower bounds on the increment of the optimal solution value if the corresponding arc is imposed in the solution. The different bounding procedures are applied in sequence, and each of them uses as input costs the residual cost matrix given as output by the previous procedure (obviously, the first procedure starts with the original cost matrix). The overall additive lower bound is given by the sum of the lower bounds obtained by each procedure. It can be easily shown that if the lower bounding procedures are based on linear programming relaxations, as those previously described for ACVRP, the reduced costs are valid residual costs. For further details see [18,17].

Procedure ADD_DISJ starts by solving the AP relaxation with no additional constraints, and defines the initial lower bound LB as the optimal AP solution value, and the arc set A^* as the arcs used in the optimal AP solution. Then iteratively an infeasible subset B, if any, is chosen from A^* and used for the computation of the disjunctive lower bound, returning a lower bound L_D and the corresponding residual cost matrix. The current LB is increased by L_D, the set A^* is updated by removing from it all the arcs whose corresponding variables are not equal to 1 in the current optimal solution of the disjunctive bound. The process is iterated until A^* does not contain further infeasible arc subsets. Procedure ADD_DISJ can be implemented, through parametric techniques, so as to have an overall time complexity equal to $O(n^4)$.

3.4. The lower bound based on min-cost flow

Let $\{S_1, \ldots, S_m\}$ be a given partition of V with $0 \in S_1$, and define

$$A_1 := \bigcup_{h=1}^{m} \{(i,j) \in A: i,j \in S_h\}$$

$$A_2 := A \setminus A_1.$$

In other words, A is partitioned into $\{A_1, A_2\}$, where A_1 contains the arcs "internal" to the subsets S_h, and A_2 those connecting vertices belonging to different S_h's.

In the following, a lower bound L_P based on projection is described. The bound is given by $L_P := \vartheta_1 + \vartheta_2$, where $\vartheta_t, t = 1, 2$, is a lower bound on $\sum (c_{ij}: (i,j) \in A^* \cap A_t)$ for every (optimal) ACVRP solution $A^* \subset A$.

The contribution to L_P of the arcs internal to the given subsets S_h is initially neglected, i.e., ϑ_1 is set equal to 0. The rationale of this choice is clarified later. As to ϑ_2, this is computed by solving the following linear programming relaxation, called R1, obtained from model VRP1 by

(i) weakening degree equations (3)–(6) into inequalities, to take into account the removal of the arcs in A_1;

(ii) imposing the capacity-cut constraints (7) and (10) only for the m subsets S_h's .
The model of R1 is

$$(\text{R1}) \quad \vartheta_2 = \min \sum_{(i,j)\in A_2} c_{ij}x_{ij} \tag{24}$$

$$\text{s.t.} \quad \sum_{i\in V:(i,j)\in A_2} x_{ij} \leqslant \begin{cases} 1 & \text{for all } j\in V\setminus\{0\}; \\ K & \text{for } j=0; \end{cases} \tag{25}$$

$$\sum_{j\in V:(i,j)\in A_2} x_{ij} \leqslant \begin{cases} 1 & \text{for all } i\in V\setminus\{0\}; \\ K & \text{for } i=0; \end{cases} \tag{26}$$

$$\sum_{i\notin S_h}\sum_{j\in S_h} x_{ij} = \sum_{i\in S_h}\sum_{j\notin S_h} x_{ij} \geqslant \begin{cases} \gamma(V\setminus S_h) & \text{for } h=1; \\ \gamma(S_h) & \text{for } h=2,\ldots,m; \end{cases} \tag{27}$$

$$x_{ij}\in\{0,1\} \quad \text{for all } (i,j)\in A_2. \tag{28}$$

This model can be solved efficiently, since it can be viewed as an instance of a min-cost flow problem on an auxiliary layered network. The network contains $2(n+m+2)$ vertices, namely:

- two vertices, say i^+ and i^-, for all $i\in V$;
- two vertices, say a_h and b_h, for all $h=1,\ldots,m$;
- a source vertex, s, and a sink vertex, t.

The arcs in the network, and the associated capacities and costs, are:

- for all $(i,j)\in A_2$: arc (i^+,j^-) with cost c_{ij} and capacity 1;
- for all $h=1,\ldots,m$: arcs (a_h,i^+) and (i^-,b_h) for all $i\in S_h$, with cost 0 and capacity 1 (if $i\neq 0$) or K (if $i=0$);
- for all $h=1,\ldots,m$: arc (a_h,b_h) with cost 0 and capacity $|S_h|-\gamma(S_h)$ (if $h\neq 1$) or $|S_1|+K-1-\gamma(V\setminus S_1)$ (if $h=1$);
- for all $h=1,\ldots,m$: arcs (s,a_h) and (b_h,t), both with cost 0 and capacity $|S_h|$ (if $h\neq 1$) or $|S_1|+K-1$ (if $h=1$).

It can be easily seen that finding the min-cost s-t flow of value $n+K$ on this network actually solves relaxation R1. The worst-case time complexity for the computation of ϑ_2, and of the corresponding residual costs, is $O(n^3)$ by using a specialized algorithm based on successive shortest path computations.

Different choices of the vertex partition $\{S_1,\ldots,S_m\}$ lead to different lower bounds. Note that choosing $S_h=\{h\}$ for all $h\in V$, produces a relaxation R1 that coincides with the AP relaxation of Section 3.1. When, on the other hand, non-singleton S_h's are present, relaxation R1 is capable of taking into account the associated capacity-cut constraints (that are, instead, neglected by AP), while loosing a possible contribution to the lower bound of the arcs inside S_h (which belong to A_1), and weakening the degree constraints of the vertices in S_h. Fischetti et al. [18] used, in sequence, different partitions obtaining an overall additive procedure, called ADD_FLOW.

The procedure is initialized with the partition $S_h=\{h\}$ for all $h=1,\ldots,m=n$ (i.e., with the AP relaxation). At each iteration of the additive scheme, relaxation R1 is solved, the current lower bound is increased, and the current costs are reduced accordingly. Then a convenient collection of subsets $S_{h_1},\dots,S_{h_r}$ (with $r \geqslant 2$) belonging to the current partition is selected and the subsets are replaced with their union, say S^*. The choice of this collection is made so as to produce an *infeasible* set S^*, i.e., a vertex set whose associated capacity-cut constraint is violated by the solution of the current relaxation R1. This hopefully produces an increase of the additive lower bound in the next iteration. The additive scheme ends when either $m=1$, or no infeasible S^* is detected.

Procedure ADD_FLOW takes $O(n^4)$ time and the resulting additive lower bound clearly dominates the AP bound, which is used to initialize it. On the other hand no dominance relation exists between ADD_FLOW and procedure ADD_DISJ of the previous section. Therefore, Fischetti, Toth and Vigo proposed to apply ADD_DISJ and ADD_FLOW in sequence, again in an additive fashion. To reduce the average overall computing time, procedure ADD_FLOW was stopped when no increase of the current lower bound LB was observed for five consecutive iterations.

3.5. Branch and bound algorithms for the ACVRP

We now briefly describe the main ingredients of the branch and bound algorithms used for the exact solution of the ACVRP proposed by Laporte et al. [31] and by Fischetti et al. [18]. The two algorithms have the same basic structure, derived from that of the algorithm for the asymmetric TSP described in Carpaneto and Toth [6] and based on that proposed in Bellmore and Malone [3]: the first one uses as lower bound the AP relaxation of Section 3.1, whereas the second uses the two additive bounding procedures described in Sections 3.3 and 3.4.

The algorithms adopt a *best-bound-first* search strategy, i.e., branching is always executed on the pending node of the branch-decision tree with the smallest lower bound value. This rule allows for the minimization of the number of subproblems solved at the expense of larger memory usage, and computationally proved to be more effective than the *depth-first* strategy, where the branching node is selected according to a last-in-first-out rule.

The branching rules used by both algorithms are related to the *subtour elimination* scheme used for the asymmetric TSP, and handle the relaxed constraints imposing the connectivity and the capacity requirements of the feasible ACVRP solutions. At a node v of the branch-decision tree, let I_v and F_v contains the arcs imposed and forbidden in the current solution, respectively.

Given the set A^* of arcs corresponding to the optimal solution of the current relaxation, a non-imposed arc subset $B{:=}\{(a_1,b_1),(a_2,b_2),\dots,(a_h,b_h)\} \subset A^*$ on which to branch is chosen.

Fischetti et al. defined B by considering the subset of A^* with the minimum number of non-imposed arcs, defining a path or a circuit which is infeasible according to the conditions of Section 3.3. Note that since the additive bounding procedure alters the objective function of the problem, an optimal solution of the relaxed problem which is feasible for ACVRP is not necessarily optimal for it. Therefore, if A^* defines a feasible ACVRP solution, set B is chosen as the feasible circuit through vertex 0 with the minimum number of non-imposed arcs. Then $h=|B|$ descendant nodes are generated. The subproblem associated with node v_i, $i=1,\ldots,h$ is defined by excluding the ith arc of B and by imposing the arcs up to $i-1$:

$$I_{v_i}:=I_v \cup \{(a_1,b_1),\ldots,(a_{i-1},b_{i-1})\},$$

$$F_{v_i}:=F_v \cup \{(a_i,b_i)\},$$

where $I_{v_1}:=I_v$.

Laporte et al. defined B as an infeasible subtour according to the conditions of Section 3.1, and used a more complex branching rule in which at each descendant node at most r arcs of B are simultaneously excluded, where $r:=\lceil d(S)/C \rceil$ and S is the set of vertices spanned by B. In this case, since at most $\binom{|B|}{r}$ descendant nodes may be generated, the set B is chosen as the one minimizing $\binom{|B|}{r}$.

The performance of the branch and bound algorithms is enhanced by means of several additional procedures performing variable fixing, feasibility checks and dominance tests. The Fischetti et al. algorithm (FTV) at each node of the branch-decision tree uses a heuristic algorithm proposed by Vigo [47] which starts from the infeasible solution associated with the current relaxation and tries to obtain a feasible solution through an insertion procedure and a post-optimization phase based on arc exchanges.

Laporte et al. used their algorithm (LMN) to solve, on a VAX 11/780 computer (0.14 Mflops), test instances where demands d_j and costs c_{ij} were randomly generated from a uniform distribution in [0, 100], and rounded to the nearest integer. The vehicle capacity was defined as

$$C:=(1-\alpha)\max_{j\in V}\{d_j\}+\alpha d(V),$$

where α is a real parameter chosen in [0,1]. The number of available vehicles was defined as $K=K_{\min}$, and computed by using the trivial BPP lower bound. Note that larger values of α produce larger C, and hence smaller K (when $\alpha=1$, ACVRP reduces to the asymmetric TSP, since $K=1$). No monotone correlation between α and the *average percentage load* of a vehicle, defined as $100\, d(V)/(KC)$, can instead be inferred. Laporte et al. considered $\alpha=0.25$, 0.50 and 0.75, producing $K=4,2$ and 2, respectively.

For each pair (n,α), five instances were generated and algorithm LMN was run by imposing a limit on the total available memory. The LMN algorithm was able to solve instances with up to 90 vertices if $\alpha \geqslant 0.50$ (i.e., with $K \leqslant 2$), although for $n \geqslant 70$ and $\alpha < 1$, only half or less of the instances were actually solved. With $\alpha=0.25$ only the instances with 10 vertices and one of those with 20 vertices were solved. The computing times for the most difficult instances solved were almost 6000 s, whereas no statistics were reported for the non-solved instances. The algorithm was also tested on instances of the same type but with $K=K_{\min}+2$ or $K=K_{\min}+4$. These problems resulted to be much easier than the previous ones: algorithm LMN was able to solve instances with up to 260 vertices. Finally, randomly generated Euclidean instances were considered and, as expected, algorithm LMN obtained poor results, being able to solve only some of the problems with two vehicles and up to 30 vertices.

Fischetti et al. tested their algorithm FTV on the same randomly generated instances used for LMN with $K=K_{\min}$. Algorithm FTV was able to solve all the instances with

Table 1
Percentage ratios of different ACVRP lower bounds with respect to the optimal solution value on real-world instances

Problem	n	K	(%) AP	(%) KSSA	(%) ADD
A034-02v	33	2	85.8	78.7	90.1
A036-03v	35	3	90.9	75.2	93.2
A039-03v	38	3	93.8	77.6	96.7
A045-03v	44	3	93.4	75.6	95.7
A048-03v	47	3	93.6	79.0	97.2
A056-03v	55	3	88.5	75.4	94.3
A065-03v	64	3	92.6	75.6	95.5
A071-03v	70	3	91.7	79.3	94.6
			91.3	77.1	94.6

up to 300 vertices and up to 4 vehicles, within 1000 CPU seconds on a DECstation 5000/240 (5.3 Mflops). For $n=90$, LMN solved one instance (out of 5) with $\alpha=0.50$ and two instances with $\alpha = 0.75$, requiring average CPU times of 5787 and 1162 s, respectively. For the same values of n and α, FTV solved all the instances within CPU times of 15 and 1 s, respectively. On these instances the additive lower bound considerably improved the AP value.

Algorithm FTV was also tested on a class of more realistic instances where the cost matrices were obtained from those of the previous class by "triangularizing" the costs, i.e., by replacing each c_{ij} with the cost of the shortest path from i to j. The number of vehicles K and the average percentage vehicle load, say r, were fixed and the vehicle capacity was defined as $C:=\lceil 100d(V)/(rK)\rceil$. Instances of this type with up to 300 vertices, 8 vehicles and with r equal to 80 and 90 were solved, those with $n \geqslant 150$ being easier than the smaller ones. Algorithm FTV was finally used to solve eight real-world instances with up to 70 vertices and 3 vehicles, coming from pharmaceutical and herbalists' product delivery in the center of an urban area with several one-way restrictions imposed on the roads. These instances resulted to be more difficult than the randomly generated ones: the computing time and the number of nodes were higher than those required for analogous random instances. The maximum CPU time required by FTV to solve the instances was about 30 minutes. Table 1 report the percentage ratios of the different lower bounds described in Sections 3.1–3.4, with respect to the optimal solution value, when applied to these real-world instances. In particular the table contains the ratios corresponding to AP, KSSA, and the overall additive bound (ADD). The average gap, over the eight instances, of the additive bound with respect to the optimal solution value was about 5.4% (that of AP being 8.7% and that of KSSA 22.9%) whereas on random instances the gap was normally much smaller (1–2% for the additive bound and 2–5% for the AP).

We recently applied algorithm FTV to some Euclidean SCVRP instances from the literature. The results we obtained show that SCVRP instances with up to 25–30 vertices may be consistently solved by this algorithm (see Table 3). Moreover, the largest instance we solved includes 47 customers.

4. The symmetric CVRP

In this section we examine the branch and bound algorithms for the symmetric version of CVRP proposed by Fisher [20] and Miller [40]. We first give a general model for SCVRP and describe the basic relaxations based on spanning trees and on b-matching. The strengthening of these basic relaxations in a Lagrangian fashion is then discussed and the overall branch and bound algorithms are described. The exact algorithm proposed by Hadjconstantinou, Christofides and Mingozzi [26], will be also briefly presented.

In the following we assume that single-customer routes are allowed.

The model we consider is obtained, as proposed in Laporte et al. [33], by adapting to SCVRP the two-index vehicle flow formulation VRP1 of ACVRP. To this end it should be noted that in SCVRP the routes are not oriented (i.e., the customers along a route may be visited indifferently clockwise or counter-clockwise). Therefore, it is not necessary to know in which direction edges are covered by the vehicles, and for each undirected edge $e \in E$ one integer variable x_e is used to indicate how many times the edge is covered in the optimal solution. In particular, if $e \notin \delta(0)$ then $x_e \in \{0,1\}$, whereas if $e \in \delta(0)$ then $x_e \in \{0,1,2\}$. The case $x_e = 2$ indicates that the endpoint customer of edge e, say j, is contained into the single-customer route $0 \to j \to 0$. The model reads:

$$\text{(VRP2)} \quad \min \sum_{e \in E} c_e x_e \tag{29}$$

$$\text{s.t.} \quad \sum_{e \in \delta(i)} x_e = 2 \quad \text{for all } i \in V \setminus \{0\}, \tag{30}$$

$$\sum_{e \in \delta(0)} x_e = 2K, \tag{31}$$

$$\sum_{e \in \delta(S)} x_e \geqslant 2\gamma(S) \quad \text{for all } S \subseteq V \setminus \{0\}, S \neq \emptyset, \tag{32}$$

$$x_e \in \{0,1,2\} \quad \text{for all } e \in \delta(0), \tag{33}$$

$$x_e \in \{0,1\} \quad \text{for all } e \notin \delta(0). \tag{34}$$

The *degree* constraints (30) and (31) impose that exactly two arcs are incident to each vertex associated with a customer, and 2K arcs are incident to the depot vertex, respectively. The capacity-cut constraints (32), where $\gamma(S)$ may be replaced by the trivial BPP lower bound, impose both the connectivity of the solution and the vehicle capacity requirements, by forcing that a sufficient number of arcs enter each subset of vertices. Also in this case, due to (30), these constraints may be rewritten as the generalized subtour elimination constraints (GSECs):

$$\sum_{e \in \sigma(S)} x_e \leqslant |S| - \gamma(S) \quad \text{for all } S \subseteq V \setminus \{0\}, S \neq \emptyset. \tag{35}$$

In addition, subtour elimination constraints as those proposed by Miller et al. [39] for the TSP may be easily extended to SCVRP (see also Section 3).

4.1. The lower bounds based on trees

Different relaxations based on spanning trees were presented for SCVRP by extending the well-known 1-tree relaxation proposed by Held and Karp [27] for the symmetric TSP.

Christofides et al. [11] proposed a branch and bound algorithm based on the *k-degree center tree* (k-DCT) relaxation of SCVRP. Given k, with $K \leqslant k \leqslant 2K$, the k-DCT is a min-cost spanning tree on G with degree k at the depot vertex. Then, K least cost arcs not in the tree are added, $2K - k$ of which are incident to the depot, and the remaining $k - K$ are not incident to it. The bound was tightened by using Lagrangian penalties associated with the degree constraints. The branch and bound algorithm was able to solve problems from the literature with up to 25 vertices within 244 s on a CDC 7600 (2 Mflops).

Another tree-based relaxation was presented in [20], and requires the determination of a K-*tree*, defined as a min-cost set of $n + K$ edges spanning the graph. The approach used by Fisher is based on formulation VRP2 with the additional assumption that single-customer routes are not allowed (by imposing $x_e \in \{0, 1\}$ for $e \in \delta(0)$). However, as he observed, in some cases this assumption is not constraining. In fact, customer j can be served alone in a route if and only if on the remaining $K - 1$ vehicles there is enough space to load the demand of the other customers, i.e., if $\gamma(V \setminus \{j\}) \leqslant K - 1$. By replacing $\gamma(\cdot)$ with the trivial BPP lower bound we may re-state the above condition as

$$d_j \geqslant C_{\min} = d(V) - (K - 1)C. \tag{36}$$

If, given a CVRP instance, condition (36) is satisfied by no $j \in V$, then in any feasible solution no customer may be served alone in a route (hence the constraint preventing it is superfluous). We checked the above condition on 65 SCVRP instances from the literature and it was satisfied, i.e., single-customer routes cannot be used, in 29 of these instances.

Fisher modeled the SCVRP as the problem of determining a K-tree with degree equal to $2K$ at the depot vertex, with additional constraints imposing: (i) the vehicle capacity requirements, and (ii) that the degree of each customer vertex must be equal to 2. These additional constraints are relaxed in a Lagrangian fashion, thus obtaining as Lagrangian problem the determination of a K-tree with degree $2K$ at the depot, which can be computed in $O(n^3)$ time (see [21]). This degree constrained K-tree relaxation may be easily obtained by considering formulation VRP2 and:

(i) removing the degree constraints (30);
(ii) weakening the capacity-cut constraints (32) into connectivity constraints by replacing the right-hand side with 1.

It can be easily seen that a K-tree solution may be infeasible for SCVRP because some vertices have degree different than two. Moreover, the demand associated with the branches leaving the depot may exceed the vehicle capacity.

Table 2
Percentage ratios of different basic SCVRP lower bounds with respect to the best known solution value of Euclidean instances

Problem	n	K	% b-matching	% K-tree[a]	% KSSA	% AP	% ADD
E045-04f	44	4	71.4	62.6	62.2	57.4	70.3
E051-05e	50	5	87.9	84.9	79.4	80.9	87.5
E072-04f	71	4	80.9	77.7	72.0	69.8	77.9
E076-10e	75	10	76.7	76.2	69.2	71.0	76.1
E101-08e	100	8	86.4	81.5	77.5	80.7	86.1
E101-10c	100	10	70.3	77.6	72.2	66.5	69.6
E135-07f	134	7	63.4	59.2	57.5	47.5	60.3
E151-12c	150	12	80.5	78.4	73.6	68.6	77.6
E200-16c	199	16	72.4	74.1	66.4	64.6	72.2
			76.7	74.7	70.0	67.4	75.5

[a]Single-customer routes not allowed.

4.2. The lower bound based on matching

The b-matching is a natural relaxation for SCVRP and is the counterpart for the symmetric version of the assignment relaxation for ACVRP described in Section 3.1. However, only recently this relaxation came on the scene, due to the work by Miller [40], after the development of efficient codes for the b-matching problem (see, e.g. [41]).

The b-matching relaxation of CVRP may be obtained by considering model VRP2 and removing the capacity-cut constraints (32). The resulting relaxed problem requires the determination of a subset of arcs covering all the vertices and such that the degree of each customer vertex is equal to two, while the degree of the depot is equal to $2K$. It can be noted that a b-matching solution may be infeasible for SCVRP since: (i) some connected components (i.e., subtours) may be disconnected from the depot, and (ii) the demand associated with a subtour may exceed the vehicle capacity. As for the AP relaxation for ACVRP, it is possible to obtain an equivalent 2-matching relaxation for SCVRP by adding $K-1$ copies of the depot and by imposing $x_e \in \{0,1\}$ for $e \in \delta(0)$ (see, Section 3.1 for further details, and Pekny and Miller [42] for an effective 2-matching algorithm). As described in the next section, in [40] some of the GSECs (35) are relaxed in a Lagrangian fashion, obtaining as Lagrangian problem the determination of a min-cost b-matching.

4.3. The Lagrangian lower bounds

The relaxations of SCVRP presented in the previous section have in general a poor quality. Table 2 reports the average percentage ratios of the basic lower bounds corresponding to the degree constrained K-tree and the b-matching with respect to the optimal or the best known solution value, for a set of widely used Euclidean SCVRP instances from the literature. The K-tree values are those reported in [20] who used real-valued cost matrices. The best known solution values that we used to compute the ratios are those reported in Toth and Vigo [46] which were obtained by using real-valued cost matrices. The b-matching values were computed with Cplex 6.0 ILP solver. The table also reports the ratios of the AP, KSSA and of the overall additive lower bound (ADD) of Fischetti et al. [18]. All these values have been computed by using integer cost matrices where the arc cost is defined as the real cost multiplied by 10,000 and rounded to the nearest integer. The final value is then scaled down by dividing it by 10,000. It should be recalled that the problem solved by Fisher in [20] was slightly different from what we defined as CVRP, since the single-customer routes were not allowed. As a consequence the K-tree values computed by Fisher may by slightly larger than those which could be obtained in the case in which single-customer routes are allowed.

By observing Table 2 it can be noted that none of the basic relaxations reaches a quality sufficient to solve moderate size problems. As an example, we used the Fischetti et al. code for ACVRP based on the additive bound ADD: the largest SCVRP instance which it was able to solve included 47 customers (problem E048-04y not included in the tables), and some problems with 25–30 customers were not solved to optimality.

Therefore, to obtain better bounds both Fisher [20] and Miller [40] strengthened the basic relaxations by dualizing, in a Lagrangian fashion, some of the relaxed constraints. In particular, Fisher included in the objective function the degree constraints (30) and some of the capacity-cut constraints (32), whereas Miller included some of the GSECs (35). It should be also remembered that Fisher did not allow single-customer routes. As in related problems, good values for the Lagrangian multipliers associated with the relaxed constraints are determined by using a standard subgradient optimization procedure (see, e.g., [19]).

The main difficulty associated with these Lagrangian relaxations is represented by the exponential cardinality of the set of relaxed constraints (i.e., the capacity-cuts and the GSECs) which does not allow for the explicit inclusion of all of them in the objective function. To this end, both authors proposed to include only a limited family $\mathscr{F}$ of relaxed capacity-cut or GSEC constraints and to iteratively add to the Lagrangian relaxation the constraints which are violated by the current solution of the Lagrangian problem. In particular, at each iteration of the subgradient optimization procedure, the arcs incident to the depot in the current Lagrangian solution are removed. Violated constraints (i.e., capacity-cuts or GSECs, depending on the approach), if any, are detected by examining the connected components obtained in this way. This detection routine is exact. In other words, if a constraint associated with, say, vertex set S, is violated by the current Lagrangian solution, then there is a connected component of that solution spanning all the vertices in S.

The new constraints are added to the Lagrangian problem, i.e., to $\mathscr{F}$, with an associated multiplier and the process is iterated until no violated constraint is detected (hence the Lagrangian solution is feasible) or a prefixed number of subgradient iterations has been executed. Slack constraints are periodically purged from $\mathscr{F}$.

Fisher [20] initialized $\mathscr{F}$ with an explicit set of constraints containing the customer subsets nested around $K+3$ *seed* customers. The seeds were chosen as the K customers farthest from the depot in the routes corresponding to an initial feasible solution, with the addition of the three customers maximally distant from the depot and the other

Table 3
Comparison of percentage ratios of the basic and improved lower bounds for SCVRP with respect to different test instances. Instances marked with an asterisk were solved to optimality by the corresponding branch and bound code

			K-tree[a]		b-matching[b]		HCM[c]	ADD[c]
Problem	n	K	% LB	% Lagr.	% LB	% Lagr.	% LB	% LB
S007-02a	6	2				100.0*		73.7*
S013-04d	12	4				96.8*		71.0*
E016-05m	15	5					97.6*	85.9*
E021-04m	20	4					100.0*	84.6*
E022-04g	21	4			90.1	99.7*		82.7*
E023-03g	22	3			96.5	100.0*		93.9*
E026-08m	25	8					100.0*	77.4
E030-03g	29	3			71.7	95.3*		—
S031-07w	30	7				96.0*		—
E031-09h	30	9					97.9*	72.8
E033-03n	32	3			86.5	98.9*		—
E036-11h	35	11					99.5*	77.1
E041-14h	40	14					98.9*	73.0
E045-04f	44	4	62.6	99.6*				70.3
E051-05e	50	5	84.9	96.7	92.9	96.9*	98.5*	87.5
E072-04f	71	4	77.7	98.3*				77.9
E076-10e	75	10	76.2	90.5			97.6	76.1
E101-08e	100	8	81.5	95.1			95.9	86.1
E101-10c	100	10	77.6	99.8*				69.6
E135-07f	134	7	59.2	97.4				60.3
E151-12c	150	12	78.4	90.7			97.2	77.6
E200-16b	199	16	74.1	84.7				72.2

[a]Real-valued costs and single-customer routes not allowed.
[b]Rounded integer costs.
[c]Real costs multiplied by 10 000 and rounded to the nearest integer.

seeds. For each seed, 60 sets were generated by including customers according to increasing distances from the seed. After 50 subgradient iterations, new sets were added to $\mathscr{F}$ by identifying violated capacity-cuts in the current Lagrangian solution as previously explained. The step size used in the subgradient optimization method was initially set to 2 and reduced by a factor of 0.75 if the lower bound was not improved in the last 30 iterations. The number of iterations of the subgradient optimization procedure performed at the root node of the branch and bound algorithm ranged between 2000 and 3000. The overall Lagrangian bound considerably improved the basic K-tree relaxation. Table 3 reports the percentage ratios of the K-tree and of the Lagrangian bound. We used the K-tree and Lagrangian bound values computed by Fisher [20] by using real-valued cost matrices and not allowing single-customers routes, and we compared the bounds with respect to the optimal or the best known solution values determined by using real-valued cost matrices. Over the nine instances considered by Fisher, the average ratio of the K-tree is 74.4% while that of the Lagrangian bound is 94.8%.

Miller [40] initialized $\mathscr{F}$ as the empty set and at each iteration of the subgradient procedure detected violated GSECs and additional constraints belonging to the following two classes. The first type of constraints is given by additional GSECs which were added when the current Lagrangian solution $\bar{x}$ contains two or more overloaded routes. The customer set of these new GSECs is the union of the sets $S_1,\ldots,S_k$ associated with the GSECs violated by $\bar{x}$. This increases the possibility that arcs connecting customers belonging to the overloaded routes to those in sets $S_1,\ldots,S_k$ are selected by the b-matching solution. The second type of constraints was added when $\bar{x}$ contained routes which were *underloaded*, i.e., routes whose associated load was smaller than the minimum vehicle load $C_{\min}$ defined by (36). In this case for each such set S, with $0 \in S$, a constraint of the form

$$\sum_{e\in\sigma(S)} x_e \leqslant |S| - 1, \tag{37}$$

which breaks the current underloaded route in $\bar{x}$, was added to $\mathscr{F}$. The procedure was iterated until no improvement was obtained since 50 subgradient iterations. The step size was modified in an adaptive way every five subgradient iterations to produce a slight oscillation in lower bound values during the progress of the subgradient procedure. If the lower bound was monotonically increasing, the step size was increased by 50%; if the oscillation of the lower bound value was greater than 2%, the step size was reduced by 20%, and when the oscillation was smaller than 0.5% it was increased by 10%.

As can be seen from Table 3, the final Lagrangian bound of Miller is considerably tight, being on average 98% of the optimal solution value for the eight problems with $n \leqslant 50$ solved by Miller by using integer rounded cost matrices. The author also communicated us some values of the pure b-matching relaxation which are reported in Table 3 (the corresponding ratio is on average 87.5%).

4.4. Bounds based on set partitioning formulation

Hadjcostantinou et al. [26] proposed a branch and bound algorithm where the lower bound is computed by heuristically solving the dual of the linear programming relaxation of the set partitioning formulation of the CVRP.

The *set partitioning* (SP) formulation of the VRP was originally proposed by Balinsky and Quandt [2] and uses a possibly exponential number of binary variables, each associated with the different feasible circuit of G. More specifically, let $\mathscr{H} = \{H_1,\ldots,H_M\}$ denote the collection of all the circuits of G each corresponding to a feasible route, with $M = |\mathscr{H}|$. Each circuit H_j has an associated optimal cost c_j, and let a_{ij} be a binary coefficient which takes value 1 if and only if vertex i is covered (i.e., visited) by route H_j. The binary variable $x_j, j = 1,\ldots,M$, is equal to 1 if and only if circuit H_j is selected in the optimal solution. The model is:

$$\text{(VRP3)} \quad \min \sum_{j=1}^{M} c_j x_j \tag{38}$$

$$\text{s.t.} \quad \sum_{j=1}^{M} a_{ij} x_j = 1 \quad i \in V \setminus \{0\} \tag{39}$$

$$\sum_{j=1}^{M} x_j = K \tag{40}$$

$$x_j \in \{0,1\} \quad j = 1,\ldots,M. \tag{41}$$

Constraints (39) impose that each customer i is covered by exactly one of the selected circuits, and (40) requires that K circuits are selected.

This is a very general model which may easily take into account several constraints as, for example, time windows, since route feasibility is implicitly considered in the definition of set $\mathscr{H}$. Agarwal et al. [1] proposed an exact algorithm for CVRP based on set partitioning approach, whereas several successful applications of this technique to tightly constrained VRPs are reported in Desrosiers et al. [14]. Moreover, the linear programming relaxation of this formulation is typically very tight (see also Bramel and Simchi-Levi [5] which give a detailed probabilistic analysis of the quality of the linear programming relaxation of the set partitioning formulation).

Hadjcostantinou et al. [26] proposed to obtain a valid lower bound to SCVRP by considering the dual of the linear relaxation of model VRP3:

$$(\text{DVRP3}) \quad \max\ K\pi_0 + \sum_{i=1}^{n} \pi_i \tag{42}$$

$$\text{s.t.} \quad \pi_0 + \sum_{i \in H_j} \pi_i \leqslant c_j \quad j = 1,\ldots,M, \tag{43}$$

$$\pi_i \quad \text{unrestricted } i = 0,\ldots,n. \tag{44}$$

Where π_i, $i = 1,\ldots,n$ are the dual variables associated with the partitioning constraints (39) and π_0 is that associated with constraint (40). It is clear that any feasible solution to problem DVRP3 provides a valid lower bound to SCVRP. Hadjcostantinou et al. [26] determined the heuristic dual solutions by combining two relaxations of the original problem: the q-path relaxation proposed in Christofides et al. [11], and the k-shortest path relaxation proposed in Christofides and Mingozzi [9]. The proposed approach was able to solve randomly generated Euclidean instances with up to 30 vertices and instances proposed in the literature with up to 50 vertices, within a time limit of 12 h on a Silicon Grapics Indigo R4000 (12 Mflops). The percentage ratio of the overall bound (HCM) are reported in Table 3.

4.5. Branching schemes and overall algorithms

Many branching schemes were used for SCVRP and almost all are extensions of those used for the TSP.

The first scheme we consider, proposed in [11], is known as *branching on arcs* and proceeds by extending partial paths, starting from the depot and finishing at a given vertex. At each node of the branch-decision tree an arc (i,j) is selected to extend the current partial path and two descendant nodes are generated: the first node is associated with the inclusion of the selected arc in the solution (i.e., $x_{ij} = 1$), while in the second node the arc is excluded (i.e., $x_{ij} = 0$).

Miller [40] used this branching scheme, where the arc to branch with is selected by examining the solution obtained by the Lagrangian relaxation based on b-matching described in Sections 4.3. When a partial path is present in the current subproblem ending, say, with vertex v, the arc (v, h) belonging to the current Lagrangian solution is selected. If the current subproblem does not contain a partially fixed path, e.g., at the root node or when a route has been closed by the last imposed arc, the arc connecting the depot with the unrouted customer j with the largest demand is selected for branching. In this case a third descendant node is also created, by imposing $x_{0j} = 2$, i.e., by considering, if feasible, the route containing only customer j. The resulting branch and bound algorithm was applied to Euclidean SCVRP instances from the literature, where the edge costs are computed as the Euclidean distances between the customers and rounded to the nearest integer. The algorithm was able to solve problems with up to 50 customers within 15,000 s on a Sun Sparc 2 (4 Mflops).

Fisher [20] used a mixed scheme where branching on arcs is used when no partial path is present in the current subproblem. In this case the currently unserved customer i with the largest demand is chosen and the arc (i, j) is used for branching, where j is the unserved customer closest to i. At the node where arc (i, j) is excluded from the solution, branching on arcs is again used, whereas at the second node the scheme known as *branching on customers* is used. One of the two ending customers, say v, of the currently imposed sequence of customers is chosen, and branching is performed by enumerating the customers which may be appended to that end of the sequence. A subset T of currently unserved customers is selected, e.g., that including the unserved customers closest to v, and $|T| + 1$ nodes are generated. Each of the first $|T|$ nodes corresponds to the inclusion in the solution of a different customer $j \in T$, while in the last node all the arcs $(v, j), j \in T$ are excluded.

The performance of this branching scheme may be enhanced by means of a dominance test proposed by Christofides et al. [11]. A node of the branch-decision tree where a partial sequence of customers $v, \ldots, w$ is fixed, can be fathomed if there exists a lower cost ordering of the customers in the sequence starting with v and ending with w. The improved ordering may be heuristically determined, e.g., by means of exchange procedures as those proposed in Lin and Kernighan [36].

The mixed branching scheme with the described dominance rule was used by Fisher to attempt the solution of Euclidean SCVRP instances with real distances and about 100 customers, but proved unsuccessful. In fact, Fisher observed that in instances where many small clusters of close customers exist (as in the case of several instances from the literature) any solution in which these customers are served contiguously in the same route have almost the same cost. Thus, when the sequence of these customers have to be determined through branching, unless an extremely tight bound is used, it would be very difficult to fathom many of the resulting nodes. Therefore, in [20] an alternative branching scheme is proposed, aiming at exploiting macro-properties of the optimal solution whose violation would have a large impact on the cost, thus allowing the fathoming of the corresponding nodes. To this end a subset T of currently unserved customers is selected and two descendant nodes are created: at the first node the additional constraint $\sum_{e \in \delta(T)} x_e = 2\lceil d(T)/C \rceil$ is added to the current problem, while at the second node the constraint $\sum_{e\in\delta(T)} x_e \geqslant 2\lceil d(T)/C\rceil+2$ is imposed. Some ways of identifying suitable subsets as well as additional dominance rules are described in [20]. This second branch and bound algorithm was successfully applied to some Euclidean SCVRP instances with real distances and with no single customer route allowed. The largest solved instance included 100 customers and was solved within less than 60,000 s on a small Apollo Domain 3000 computer (0.071 Mflops). Note, however, that several Euclidean instances from the literature were not solved to optimality.

5. Conclusions

In this paper we reviewed the most important branch and bound algorithms proposed during the last decade for the capacitated vehicle routing problem with either symmetric or asymmetric cost matrix. The progress made with these algorithms with respect to those of the previous generation is considerable: the dimension of the largest instances solved has been increased from about 25 to more than 100 customers. However, the CVRP is still far from being a closed chapter in the combinatorial optimization book. In fact some Euclidean problems from the literature with 75 customers are still unsolved and, in our opinion, the size of the problems which may be actually solved in a systematic way by the present approaches is limited to few tenths of customers.

Several possible directions of research are still almost uncovered, e.g., Dantzig–Wolfe decomposition based approaches (also known as *branch and price* approaches), but also a more deep investigation and understanding of the capabilities of the available techniques is strongly needed. As an example, we may mention that a direct computational evaluation and comparison of the effectiveness of the algorithms presented in this paper for the symmetric case is not possible. In fact, as illustrated in Table 3, each author either considered a slightly different problem (e.g., in [20] single customers routes were not allowed, whereas Miller [40] allowed them) or solved a completely different set of instances. The only instance which has been tackled by almost all the authors we considered is the 50 customers Euclidean problem described in Christofides and Eilon [8]. However, for this instance Fisher [20] used a real-valued cost matrix with Euclidean distances, Miller [40] used an integer cost matrix with Euclidean distances rounded to the nearest integer. As to Hadjconstantinou et al. [26], they used a hybrid solution where the integer cost of each arc is defined as the Euclidean distance between its endpoints multiplied by 10^4 and then rounded to nearest integer. Another research issue which may lead to interesting results is represented by the adaptation to the symmetric CVRP of the exact approaches developed for the asymmetric case, and vice versa.

Acknowledgements

This work has been supported by Ministero dell'Università e della Ricerca Scientifica e Tecnologica (MURST), and by Consiglio Nazionale delle Ricerche (CNR), Italy.

The computational experiments have been executed at the Laboratory of Operations Research of the University of Bologna (Lab.O.R.).

References

[1] Y. Agarwal, K. Mathur, H.M. Salkin, A set-partitioning-based exact algorithm for the vehicle routing problem, Networks 19 (1989) 731–749.
[2] M. Balinski, R. Quandt, On an integer program for a delivery problem, Oper. Res. 12 (1964) 300–304.
[3] M. Bellmore, J.C. Malone, Pathology of travelling salesman subtour-elimination algorithms, Oper. Res. 19 (1971) 278–307.
[4] L. Bodin, B.L. Golden, A.A. Assad, M.O. Ball, Routing and scheduling of vehicles and crews, the state of the art, Comput. Oper. Res. 10 (2) (1983) 63–212.
[5] J. Bramel, D. Simchi-Levi, On the effectiveness of the set partitioning formulation for the vehicle routing problem, Oper. Res. 45 (1997) 295–301.
[6] G.Carpaneto, P. Toth, Some new branching and bounding criteria for the asymmetric traveling salesman problem, Manage. Sci. 26 (1980) 736–743.
[7] N. Christofides, Vehicle routing, in: E.L. Lawler, J.K. Lenstra, A.H.G. Rinnooy Kan, D.B. Shmoys (Eds.), The Traveling Salesman Problem, Wiley, Chichester, 1985, pp. 431–448.
[8] N. Christofides, S. Eilon, An algorithm for the vehicle dispatching problem, Oper. Res. Quart. 20 (1969) 309–318.
[9] N. Christofides, A. Mingozzi, Vehicle routing: practical and algorithmic aspects, in: C.F.H. van Rijn (Ed.), Logistics: Where Ends Have to Meet, Pergamon Press, Oxford, 1989, pp. 30–48.
[10] N. Christofides, A. Mingozzi, P. Toth, The vehicle routing problem, in: N. Christofides, A. Mingozzi, P. Toth, C. Sandi (Eds.), Combinatorial Optimization, Wiley, Chichester, 1979, pp. 315–338.
[11] N. Christofides, A. Mingozzi, P. Toth, Exact algorithms for the vehicle routing problem based on the spanning tree and shortest path relaxations, Math. Programming 20 (1981) 255–282.
[12] G. Cornuéjols, F. Harche, Polyhedral study of the capacitated vehicle routing problem, Math. Programming 60 (1) (1993) 21–52.
[13] M. Desrochers, G. Laporte, Improvements and extensions to the Miller-Tucker-Zemlin subtour elimination constraints, Oper. Res. Lett. 10 (1991) 27–36.
[14] J. Desrosiers, Y. Dumas, M.M. Solomon, F. Soumis, Time constrained routing and scheduling, in: 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, 1995, pp. 35–139.
[15] J.J. Dongarra, Performance of various computers using standard linear equations software, Technical Report CS-89-85, University of Tennessee, Knoxville, 1996.
[16] M. Fischetti, P. Toth, An additive bounding procedure for combinatorial optimization problems, Oper. Res. 37 (1989) 319–328.
[17] M. Fischetti, P. Toth, An additive bounding procedure for the asymmetric travelling salesman problem, Math. Programming 53 (1992) 173–197.
[18] M. Fischetti, P. Toth, D. Vigo, A branch-and-bound algorithm for the capacitated vehicle routing problem on directed graphs, Oper. Res. 42 (5) (1994) 846–859.
[19] M.L. Fisher, An application oriented guide to Lagrangian relaxation, Interfaces 15 (1985) 10–21.
[20] M.L. Fisher, Optimal solution of vehicle routing problems using minimum k-trees, Oper. Res. 42 (1994) 626–642.
[21] M.L. Fisher, A polynomial algorithm for the degree constrained k-tree problem, Oper. Res. 42 (4) (1994) 776–780.
[22] M.L. Fisher, Vehicle routing, in: 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, 1995, pp. 1–33.
[23] H.N. Gabow, R.E. Tarjan, Efficient algorithms for a family of matroid intersection problems, J. Algorithms 5 (1984) 80–131.
[24] B.L. Golden, A.A. Assad, Vehicle Routing: Methods and Studies, North-Holland, Amsterdam, 1988.

[25] B.L. Golden, E.A. Wasil, J.P. Kelly, I. Chao, The impact of metaheuristics on solving the vehicle routing problem: algorithms, problem sets, and computational results, in: T.G. Crainic, G. Laporte (Eds.), Fleet Management and Logistic, Kluwer Academic Publisher, Boston (MA), 1998, pp. 33–56.
[26] E. Hadjiconstantinou, N. Christofides, A. Mingozzi, A new exact algorithm for the vehicle routing problem based on q-Paths and k-Shortest paths relaxations, Ann. Oper. Res. 61 (1995) 21–43.
[27] M. Held, R.M. Karp, The traveling salesman problem and minimum spanning tress: Part II, Math. Programming 1 (1971) 6–25.
[28] R.V. Kulkarni, P.V. Bhave, Integer programming formulations of vehicle routing problems, European J. Oper. Res. 20 (1985) 58–67.
[29] G. Laporte, The vehicle routing problem: An overview of exact and approximate algorithms, European J. Oper. Res. 59 (1992) 345–358.
[30] G. Laporte, Vehicle routing, in: M. Dell'Amico, F. Maffioli, S. Martello (Eds.), Annotated Bibliographies in Combinatorial Optimization, Wiley, Chichester, 1997.
[31] G. Laporte, H. Mercure, Y. Nobert, An exact algorithm for the asymmetrical capacitated vehicle routing problem, Networks 16 (1986) 33–46.
[32] G. Laporte, Y. Nobert, Exact algorithms for the vehicle routing problem, Ann. Discrete Math. 31 (1987) 147–184.
[33] G. Laporte, Y. Nobert, M. Desrochers, Optimal routing under capacity and distance restrictions, Oper. Res. 33 (1985) 1050–1073.
[34] G. Laporte, I.H. Osman, Routing problems: a bibliography, Ann. Oper. Res. 61 (1995) 227–262.
[35] J.K. Lenstra, A.H.G. Rinnooy Kan, Some simple applications of the traveling salesman problem, Oper. Res. Quart. 26 (1975) 717–734.
[36] S. Lin, B.W. Kernighan, An effective heuristic algorithm for the travelling salesman problem, Oper. Res. 21 (1973) 498–516.
[37] T.L. Magnanti, Combinatorial optimization and vehicle fleet planning: Perspectives and prospects, Networks 11 (1981) 179–214.
[38] S. Martello, P. Toth, Knapsack Problems: Algorithms and Computer Implementations, Wiley, Chichester, 1990.
[39] C.E. Miller, A.W. Tucker, R.A. Zemlin, Integer programming formulations and traveling salesman problems, ACM 7 (1960) 326–329.
[40] D.L. Miller, A matching based exact algorithm for capacitated vehicle routing problems, ORSA J. Comput. 7 (1) (1995) 1–9.
[41] D.L. Miller, J.F. Pekny, A staged primal-dual algorithm for perfect b-matching with edge capacities, ORSA J. Comput. 7 (1995) 298–320.
[42] J.F. Pekny, D.L. Miller, A staged primal-dual algorithm for finding a minimum cost perfect two-matching in an undirected graph, ORSA J. Comput. 6 (1994) 68–81.
[43] P. Toth, D. Vigo, An exact algorithm for the capacitated shortest spanning arborescence, Ann. Oper. Res. 61 (1995) 121–142.
[44] P. Toth, D. Vigo, An exact algorithm for the vehicle routing problem with backhauls, Transport. Sci. 31 (1997) 372–385.
[45] P. Toth, D. Vigo, Exact solution of the vehicle routing problem, in: T.G. Crainic, G. Laporte (Eds.), Fleet Management and Logistic, Kluwer Academic Publisher, Boston (MA), 1998, pp. 1–31.
[46] P. Toth, D. Vigo, The granular tabu search (and its application to the vehicle routing problem). Technical Report OR/98/9, D.E.I.S. - Università di Bologna, 1998.
[47] D. Vigo, A heuristic algorithm for the asymmetric capacitated vehicle routing problem, European J. Oper. Res. 89 (1996) 108–126.
[48] D. Vigo, VRPLIB: a vehicle routing problem library, Technical Report OR/99/9, D.E.I.S., Universitá di Bologna, 1999.