Chapter 2

Classical Exact Algorithms for the Capacitated Vehicle Routing Problem

Frédéric Semet
Paolo Toth
Daniele Vigo

2.1 ▪ Introduction

In this chapter we present an overview of the early exact methods used for the solution of the *Capacitated Vehicle Routing Problem* (CVRP). The CVRP is an extension of the well-known *Traveling Salesman Problem* (TSP), calling for the determination of a Hamiltonian circuit with minimum cost visiting exactly once a given set of points. Therefore, the foundation of many exact approaches for the CVRP were derived from the extensive and successful work done for the exact solution of the TSP. However, even if tremendous progress has been made with respect to the first algorithms, such as the tree search method by Christofides and Eilon [17], the CVRP is still far from being satisfactorily solved. Our analysis encompasses more than three decades of research and examines the main families of approaches, from direct tree search methods based on Branch-and-Bound to column generation and Branch-and-Cut algorithms presented around the year 2000. The wide variety and richness of methods proposed in these early decades of CVRP history is witnessed by the good number of survey works that analyzed the relevant literature. Following the first comprehensive work of Laporte and Nobert [39], several review papers were devoted to the analysis of exact algorithms for the VRP as those of Laporte [36], Toth and Vigo [52, 53], Bramel and Simchi-Levi [15], Naddef and Rinaldi [47], Cordeau et al. [20], and Baldacci, Toth, and Vigo [10, 11]. More recent Branch-and-Cut-and-Price algorithms, which have successfully combined and enhanced those described in the following, will be covered in detail in Chapter 3.

We will denote with SCVRP and ACVRP the symmetric and asymmetric CVRP, respectively. When the explicit distinction between the two versions is not needed, we simply use CVRP. Moreover, throughout this chapter the graphs, both directed or undirected, are assumed to be complete. In this chapter we extensively refer to the basic notation and to the models presented in Chapter 1, but we recall the most used elements to facilitate the reading. The computational testing of the algorithms is generally performed by using a rather limited set of benchmark instances from the literature. To identify such test instances we adopt the naming convention described in Chapter 1 of Toth and Vigo [54]. Moreover, when available we also report the hardware used in the tests and its relative speed in Mflops given by Dongarra [24].

The chapter is subdivided in three sections, each devoted to the main approaches used in early algorithms. In particular, in Section 2.2 we examine the Branch-and-Bound algorithms and in Section 2.3 the column generation methods based on the set partitioning formulation. Finally, in Section 2.4 we present the algorithms based on Branch-and-Cut paradigms.

2.2 ▪ Branch-and-Bound Algorithms

Up to the late eighties, the most effective exact approaches for the CVRP were mainly direct tree search algorithms based on Branch-and-Bound. Similarly to what was proposed earlier for the TSP (see Little et al. [42]) they incorporated basic combinatorial relaxations based either on the *Assignment Problem* (AP) or the *Shortest Spanning Tree* (SST). Such initial approaches were able to solve to optimality instances with some tens of customers, even with the relatively limited computing hardware available at that time. Given such encouraging results, at the end of the nineties more sophisticated bounds were proposed, such as those based on Lagrangian relaxations and the additive approach, which brought direct tree search methods to their best possible performance prior to the systematic introduction of cutting planes.

To illustrate the relaxations that provide the fundamental component of Branch-and-Bound methods we refer to the two-index vehicle flow formulations of the VRP, which are described in detail in Chapter 1 and briefly summarized hereafter to facilitate the reading. The ACVRP is defined on a complete directed graph $G=(V,A)$, where $V=\{0\}\cup N=\{0,1,\dots,n\}$ is the set of vertices representing the depot (vertex 0) and the customers, which are the vertices in $N=\{1,\dots,n\}$. Set A includes the directed arcs, whereas, in the symmetric case, the undirected graph $G=(V,E)$ contains the set E of undirected edges. The cost matrix c, associated with the arcs or edges, is either asymmetric (for ACVRP) or symmetric (for SCVRP). Finally, with each customer $i\in N$ is associated a demand $q_i\geq 0$ and a fleet K of identical vehicles, each having capacity Q available at the depot. Further details on the notation are given in Section 1.2. The models, denoted as VRP1 for the ACVRP and as VRP2 for the SCVRP, are reported side by side in the following:

		(VRP1)	(VRP2)	
(2.1)		minimize $c^\top x$	minimize $c^\top x$	
(2.2)	s.t.	$x(\delta^+(i))=1$	$x(\delta(i))=2$	$\forall i\in N,$
		$x(\delta^-(j))=1$		$\forall j\in N,$
(2.3)		$x(\delta^+(0))=\lvert K\rvert$	$x(\delta(0))=2\lvert K\rvert,$	
(2.4)		$x(\delta^+(S))\geq r(S)$	$x(\delta(S))\geq 2r(S)$	$\forall S\subseteq N, S\neq\emptyset,$
(2.5)		$x_a\in\{0,1\}\ \forall a\in A$	$x_e\in\{0,1,2\}\ \forall e\in\delta(0),$	
			$x_e\in\{0,1\}\ \forall e\in E\setminus\delta(0).$	

In this formulation, constraints (2.2) and (2.3) are the degree restrictions at the customers and at the depot. Constraints (2.4) impose the capacity constraints. In their expression, $r(S)$ represents the number of vehicles required to serve the vertices in S. Such constraints, called *Generalized Subtour Elimination Constraints* (GSEC), also impose solution connectivity since $r(S)\geq 1$. Finally, constraints (2.5) define the decision variables.

Generally these are binary variables, indicating whether the corresponding arcs or edges are used or not in the optimal CVRP solution; however, for the SCVRP the variables associated with the edges incident into the depot may also take value two, thus representing a route which includes only one customer.

In the following we review the main ingredients of Branch-and-Bound algorithms proposed for the CVRP. We first examine the basic combinatorial relaxations obtained by algebraic manipulation of the above formulation. The quality of the resulting lower bounds is generally poor, and substantial efforts are needed to improve them, as in the bounding procedures based on Lagrangian and additive approaches presented next. The section is concluded examining branching a reduction strategy commonly implemented in the literature.

2.2.1 ▪ Bounds Based on Assignment and Matching

The first type of combinatorial relaxation for the CVRP is an extension of the method proposed by Little et al. [42] for the TSP: it is obtained by considering the directed model VRP1 and dropping the GSECs (2.4). The resulting problem is a *Transportation Problem* (TP), calling for a min-cost collection of circuits of G visiting once all the vertices in N, and $|K|$ times vertex 0. The solution of the relaxed problem can be infeasible for the CVRP since

(i) the total customer demand on a route may exceed the vehicle capacity;

(ii) there may exist "isolated" routes, i.e., subtours not visiting the depot.

To efficiently solve the TP, early algorithms transformed it into an equivalent AP defined on an extended complete directed graph $G' = (V', A')$, obtained by adding $|K|-1$ copies of the depot vertex to V. More precisely, $V' := N \cup W$, where $W := \{0\} \cup \{n+1, \ldots, n+|K|-1\}$ is the set of the $|K|$ vertices of G' associated with the depot and each customer in N is connected to each such copy. Moreover, the cost c'_{ij} of each arc in A' is defined as follows:

$$c'_{ij} := \begin{cases} c_{ij} & \text{for } i, j \in N; \\ c_{i0} & \text{for } i \in N,\ j \in W; \\ c_{0j} & \text{for } i \in W,\ j \in N; \\ \lambda & \text{for } i, j \in W, \end{cases} \tag{2.6}$$

where λ is a parameter whose value influences the number of vehicles used by the solution. In particular, when $\lambda = +\infty$, the model imposes using all the $|K|$ available vehicles, as is typically required in CVRP. Note, however, that other values of the parameter permit using fewer vehicles if convenient. For example, $\lambda = 0$ leads to the min-cost solution using *at most* $|K|$ routes, whereas defining $\lambda = -\infty$ leads to the min-cost solution using exactly $r(N)$ routes.

This type of relaxation was used in the first Branch-and-Bound for SCVRP by Christofides and Eilon [17], who also used a simple bound based on SST and were able to solve two small problems with 6 and 13 customers on an IBM 7090. The same relaxation was used by Laporte, Mercure, and Nobert [37] within a Branch-and-Bound algorithm for the ACVRP which, thanks to the much better quality of the AP relaxation for asymmetric problems, was able to solve randomly generated instances with some tens of customers and up to four vehicles VAX 11/780 (0.14 Mflops).

The same type of relaxation when performed on the symmetric model VRP2 amounts to a so-called b-Matching problem and requires the determination of a min-cost collection of tours covering all the vertices and such that the degree of each vertex i is equal to b_i, where $b_i = 2$ for all the customer vertices, and $b_0 = 2|K|$ for the depot vertex. This relaxation was used by Miller [45] as a base for an effective Lagrangian bound presented in Section 2.2.3. Also in this case it is possible to transform the problem by adding $|K|-1$ copies of the depot, thus obtaining an equivalent 2-Matching relaxation.

According to the experimental evaluation performed in Toth and Vigo [53] on nine test instances including between 44 and 199 customers, the b-Matching is the best relaxation of this type with an average ratio of 76.7% of the corresponding lower bound with respect to the best known feasible solution value. The simpler AP bound has a worse performance on symmetric instances, with a ratio of 67.4%. However, a similar evaluation on asymmetric instances with up to 70 customers shows, not surprisingly, a much better ratio of 91.3%.

We finally mention that Fischetti, Toth, and Vigo [27] used an AP relaxation as the base for a bounding procedure based on a disjunction on infeasible arc subsets. Given a set B of arcs that may not be used all together in any feasible solution of CVRP, a valid lower bound can be computed as the minimum value of the bounds computed by excluding in turn each such arc from the solution. This is easily done by setting to ∞ the cost of an arc in B and determining the corresponding bound. In [27] the AP relaxation was used both to detect possible sets B and to compute the overall bound which resulted slightly better than that obtained with the AP relaxation for ACVRP instances.

2.2.2 ▪ Bounds Based on Spanning Trees and Shortest Paths

Several relaxations based on the solution of the SST, i.e., the problem of finding a minimum-cost subset of edges connecting all vertices of the graph, were proposed for SCVRP. These relaxations are obtained by weakening the GSECs so as to impose only the connectivity of the solution and by ignoring part of the degree requirements of the vertices.

The first attempt in this direction was performed by Christofides and Eilon [17], who directly used the well-known 1-tree relaxation introduced by Held and Karp [34] for the TSP. The 1-tree is a subset of $|V| = n+1$ edges obtained by adding to the n edges, forming an SST over G, the minimum cost edge not belonging to the SST. The direct extension of the 1-tree relaxation to SCVRP was performed by Fisher [28], who used as a basic relaxation a K-tree, defined as a set of $n+|K|$ edges spanning the graph. Fisher modeled the SCVRP as the problem of determining a K-tree with degree equal to $2|K|$ at the depot vertex, and with additional constraints imposing (i) the vehicle capacity requirements, and (ii) the degree of each customer vertex, which must be equal to 2. The determination of a K-tree with degree $2K$ at the depot requires $O(n^3)$ time. Although the quality of the K-tree bound is experimentally quite poor (see Toth and Vigo [53], who report a ratio of 74.7%), it was incorporated into an effective Lagrangian bound described in Section 2.2.3. Similar relaxations may be derived for the ACVRP as described in Toth and Vigo [53].

A different tree-based relaxation for SCVRP was introduced by Christofides, Mingozzi, and Toth [18], who defined the so-called *k-Degree Center Tree* (k-DCT) as a tree with degree k at vertex 0, where $|K| \leq k \leq 2|K|$. The overall relaxation is obtained by adding to the k-DCT a set of $2|K|-k$ edges with minimum total cost. An overall Lagrangian bound was then derived by dualizing the degree constraints at the customers (2.2) and at the depot (2.3). The resulting bound incorporated into a Branch-and-Bound algorithm was able to solve instances with up to 20 customers on a CDC 7600 (3.3 Mflops).

Another very important relaxation of SCVRP, first described in Christofides, Mingozzi, and Toth [18], is based on the so-called q-route, which is not necessarily a simple route with total load equal to q and not including loops with two vertices. In [18] an efficient procedure is given to compute a q-route as a q-path from the depot to customer i plus the arc from i to the depot. The q-routes are used to derive an overall bound that proved to be tighter than the k-DCT one and allowed solving instances with up to 25 customers. In Hadjiconstantinou, Christofides, and Mingozzi [33] an enhanced version of this method, called *through q-route*, is obtained by combining the two shortest paths from the depot to customer i. The q-route concept is also extensively used in recent Branch-and-Cut-and-Price algorithms that are the current best exact approaches for the CVRP and are described in detail in Chapter 3.

2.2.3 ▪ Improved Bounds: Lagrangian and Additive Approaches

As discussed in the previous sections, the basic combinatorial relaxations available for both ACVRP and SCVRP have a poor quality and, when used within Branch-and-Bound approaches, they allow for the optimal solution of small instances only. To considerably increase the solution effectiveness of the Branch-and-Bound algorithms more sophisticated bounding techniques are needed.

Fisher [28] and Miller [45] proposed strengthening the basic SCVRP relaxations by dualizing, in a Lagrangian fashion, some of the relaxed constraints. In particular, Fisher started from his K-tree relaxation and included in the objective function the degree constraints (2.2) and some of the GSECs (2.4) in the cut form called *Capacity Cut Constraints* (CCCs), whereas Miller included a subset of the GSECs that were removed to obtain the b-matching relaxation. 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., Held, Wolfe, and Crowder [35]).

The main difficulty associated with these relaxations is represented by the exponential cardinality of the set of relaxed constraints (i.e., the CCCs and GSECs), which does not allow for the explicit inclusion of all of them into the objective function. To this end, both authors proposed including only a limited family $\mathscr{F}$ of CCCs or GSECs and iteratively adding 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 edges incident to the depot in the current Lagrangian solution are removed. Violated constraints (i.e., CCCs or GSECs, depending on the approach), if any, are *separated* (i.e., detected) by examining the connected components obtained in this way. This separation routine is exact; i.e., 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 and violating the constraint. 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* (i.e., removed) from $\mathscr{F}$.

Fisher [28] 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, whereas the last three customers were the ones maximally distant from the depot and the other seeds. For each seed, 60 sets were generated by including customers according to increasing distances from the seed. New violated CCCs, if any, are identified and added every 50 subgradient 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 and was, on average, larger than 99% of the optimal solution value for the three

Euclidean instances with $n \leq 100$ solved to optimality in Fisher [28] within 60,000 seconds on an Apollo Domain 3000 (0.071 Mflops).

Miller [45] initialized $\mathscr{F}$ as the empty set and at each iteration of the subgradient procedure detected violated GSECs, if any. The iteration was stopped when no improvement was obtained over 50 subgradient iterations. Also in this case the final Lagrangian bound is considerably tight, being on average 98% of the optimal solution value for the eight problems with $n \leq 50$ solved in [45] within 15,000 seconds on a Sun Sparc 2 (4 Mflops).

The relax-and-cut algorithm by Martinhon, Lucena, and Maculan [44] generalizes these Lagrangian-based approaches by considering also comb and multistar inequalities, and they were able to moderately improve the quality of the overall Lagrangian bound.

As to the ACVRP, Fischetti, Toth, and Vigo [27] obtained an improvement with respect to the AP bound by combining several different relaxations into an overall *additive* bounding procedure. The additive approach was proposed by Fischetti and Toth [26] and allows for the combination of different lower bounding procedures, each exploiting different substructures of the considered problem. The bounding procedures are applied in sequence, and the overall additive lower bound is given by the sum of the lower bounds obtained in this way. The bounding procedures that are combined in [27] are the disjunctive relaxation described in Section 2.2.1 and a new one based on min-cost flow computation that permits imposing a subset of GSECs. The resulting additive bound is considerably better than the AP bound and when used in a Branch-and-Bound algorithm permitted to solve random ACVRP instances with up to 300 customers and four vehicles within 1000 CPU seconds on a DECstation 5000/240 (5.3 Mflops). In [27] some real-world ACVRP instances are also solved by using the additive bound, whereas in Toth and Vigo [53] its successful application to SCVRP instances with up to 47 vertices is reported.

An interesting additive approach was also adopted by Hadjiconstantinou, Christofides, and Mingozzi [33] to compute a lower bound to CVRP. In particular, they considered the set partitioning formulation VRP4 of CVRP (see Section 2.3) and the dual of the corresponding linear programming relaxation. It is clear that, by linear programming duality, any feasible solution to such a dual problem provides a valid lower bound for CVRP. Therefore, they combined different relaxations based on q-routes and shortest paths to compute feasible solutions of the dual problem. The resulting lower bound was very tight and permitted to solve problem instances with up to 50 customers within a time limit of 12 hours on a Silicon Graphics Indigo R4000 (12 Mflops).

2.2.4 ▪ Structure of the Branch-and-Bound Algorithms

In addition to the bounding procedures, several other ingredients are crucial for a successful implementation of a Branch-and-Bound algorithm, and CVRP is not an exception. Most of the issues we analyze below are relevant also for other approaches which use the implicit enumeration scheme of Branch-and-Bound, such as Branch-and-Cut and Branch-and-Cut-and-Price.

Branching Scheme. 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 Christofides [17], 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 edge (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., $x_{ij} = 0$). In Miller [45] the arc selected for branching is determined as that expanding the current partial path in the best Lagrangian solution. When no such partial path exists (e.g., at the root node of the branch-decision tree) the selected arc is that connecting the depot with the unserved customer with the largest demand.

Fisher [28] used a mixed scheme where branching on arcs is used whenever 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 arc (v, j), $j \in T$, while in the last node all the arcs (v, j), $j \in T$, are excluded. The mixed branching scheme was used by Fisher to attempt the solution of Euclidean CVRP instances with real distances and about 100 customers, but it was unsuccessful. In fact, Fisher observed that in instances where many small clusters of close customers exist (as is the case of several instances from the literature) any solution in which these customers are served contiguously in the same route has almost the same cost. Thus, when the sequence of these customers has 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 [28] an alternative branching scheme based on specific GSECs 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 $\sum_{e\in\delta(T)} x_e = 2\lceil q(T)/C\rceil$ is added to the current problem, while at the second node the constraint $\sum_{e\in\delta(T)} x_e \geq 2\lceil q(T)/C\rceil + 2$ is imposed. Some ways of identifying suitable subsets, as well as additional dominance rules, are described in [28].

The two algorithms proposed for ACVRP by Laporte, Mercure, and Nobert [37] and by Fischetti, Toth, and Vigo [27] adopted the same branching rule related to the *subtour elimination* scheme used for the asymmetric TSP. At a node ν of the branch-decision tree, let I_ν and F_ν contain the arcs imposed and forbidden in the current solution, respectively (with $I_\nu = \emptyset$ and $F_\nu = \emptyset$ if ν is the root node). Given the set A^* of arcs corresponding to the optimal solution of the current relaxation, a non-imposed arc subset $B := \{(a_1, b_1), (a_2, b_2), \ldots, (a_h, b_h)\} \subset A^*$ on which to branch is chosen. In [27] set B is defined by considering the subset of A^* with the minimum number of non-imposed arcs among those defining a path or a circuit which is infeasible because it is overloaded or disconnected from the depot. Then $h = |B|$ descendant nodes are generated. The subproblem associated with node ν_i, $i = 1, \ldots, h$, is defined by excluding the ith arc of B and by imposing the arcs up to $i-1$:

$$I_{\nu_i} := I_\nu \cup \left\{(a_1, b_1), \ldots, (a_{i-1}, b_{i-1})\right\}, \tag{2.7}$$

$$F_{\nu_i} := F_\nu \cup \{(a_i, b_i)\}, \tag{2.8}$$

where $I_{\nu_1} := I_\nu$.

Laporte, Mercure, and Nobert [37] defined B as an infeasible subtour according to conditions (i) and (ii) of Section 2.2.1, and used a more complex branching rule in which, at each descendant node, at most r arcs of B are simultaneously excluded, where $r := \lceil q(S)/C\rceil$, S is the set of vertices spanned by B, and $q(S)$ represents the sum of the demands of the vertices in S. 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 algorithms for CVRP generally 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* (LIFO) rule.

Reduction and Dominance. Several rules may be used to possibly remove some arcs which cannot belong to an optimal solution, hence forbidding their use in the computation of bounds and allowing for the early detection of infeasibilities and dominance relations, thus speeding up the solution of CVRP. Many of these rules are inspired by the work done on the TSP. In the following we refer, for short, to the more general case of the ACVRP and we explicitly remove arcs from A even if often, to preserve graph completeness, such a removal is implemented by setting the cost of the arcs to be removed equal to a very large positive value equivalent to $+\infty$.

The reduction rules may be applied either to the original problem or to a subproblem associated with a node of the branch-decision tree, where arcs of a given subset I are imposed in the solution, as it happens in Branch-and-Bound and Branch-and-Cut algorithms. In this latter case the arcs of I define complete routes and paths, some of which may enter or leave the depot. For reduction purposes it is convenient to create a reduced graph $\tilde{G} = (\tilde{V}, \tilde{A})$ in which all the customers belonging to the complete routes induced by I are removed from $\tilde{V}$ and the set $\tilde{K}$ of available vehicles is updated accordingly. In addition, all paths induced by I are replaced in $\tilde{G}$ by single vertices with demand equal to the total demand of the vertices in the path. The costs of the arcs entering and leaving each such representative vertex are defined as those of the arcs entering the first and leaving the last vertex in the path, respectively.

The first type of reduction rules tries to remove from $\tilde{A}$ all the arcs that, if used, would produce infeasible solutions, namely those of each pair $i, j \in \tilde{V}$ such that $q_i + q_j > Q$. The second type of reduction rules tries to remove for $\tilde{A}$ the arcs that, if used, would not improve the currently best known solution. For example, let L and U be a lower and an upper bound on the optimal ACVRP solution value, respectively. For each $(i, j) \in \tilde{A}$ let $\overline{c}_{ij}$ be the reduced cost of arc (i, j) associated with the lower bound L. It is well known that the reduced cost of an arc represents a lower bound on the increase of the optimal solution value if this arc is used. Therefore, for each $(i, j) \in \tilde{A}$, if $L + \overline{c}_{ij} \geq U$, we may remove (i, j) from $\tilde{A}$. Whenever a customer has only one entering or leaving arc belonging to $\tilde{A}$, we may impose this arc (by adding it to I), redefine the graph $\tilde{G}$, and execute again the reductions above.

The performance of the branching schemes may be enhanced by means of a dominance test proposed by Fischetti and Toth [25]. 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 insertion and exchange procedures.

Finally, several Branch-and-Bound algorithms include the use of heuristic algorithms which exploit the information associated with the current relaxed problems to obtain feasible solutions which may possibly improve the current incumbent solution.

2.3 ▪ Early Set Partitioning Algorithms

An alternative formulation that has been widely used to model CVRP and its variants is that based on *Set Partitioning* (SP) or *Set Covering* (SC). The formulation was originally proposed by Balinski and Quandt [12] and uses a possibly exponential number of binary variables. The formulation VRP4, presented in Chapter 1, is recalled here to facilitate the reading. Let Ω denote the collection of all the circuits of G, corresponding to feasible CVRP routes. Each route $r \in \Omega$ has an associated cost c_r, and let a_{ir} be a binary coefficient which takes value 1 if vertex $i \in N$ is visited (i.e., *covered*) by route r, and 0 otherwise. The binary variable λ_r, $r \in \Omega$, is equal to 1 if and only if route r is selected in the optimal CVRP solution. The resulting extensive model for CVRP is then

$$\text{(2.9)} \qquad \text{(VRP4)} \qquad \text{minimize } c^\top \lambda$$

$$\text{(2.10)} \qquad \text{s.t.} \quad \sum_{r \in \Omega} a_{ir} \lambda_r = 1 \qquad \forall i \in N,$$

$$\text{(2.11)} \qquad \mathbb{1}^\top r = |K|,$$

$$\text{(2.12)} \qquad \lambda \in \{0,1\}^r.$$

Constraints (2.10) impose that each customer i is covered by exactly one of the selected routes, and (2.11) requires that $|K|$ routes be selected. As route feasibility is implicitly considered in the definition of set Ω, this is a very general model which may easily take into account additional constraints. Moreover, when the cost matrix satisfies the triangle inequality (i.e., $c_{ij} \le c_{ik} + c_{kj}$ for all $i, j, k \in V$), the SP model VRP4 may be transformed into an equivalent covering model, VRP4$_\geq$, by replacing equality with the inequality "$\geq$" in (2.11). Any feasible solution to model VRP4 is clearly feasible for VRP4$_\geq$, and any feasible solution to VRP4$_\geq$ may be transformed into a feasible solution of VRP4 of not greater cost. Indeed, if one customer is visited more than once in a VRP4$_\geq$ solution, it may be removed from all but one of the routes where it is served by applying shortcuts that will not increase the solution cost because of the triangle inequality. The main advantage of using the VRP4$_\geq$ formulation with respect to the VRP4 one is that in the former only inclusion-maximal feasible circuits, among those with the same cost, need to be considered in the definition of Ω. This considerably reduces the number $|\Omega|$ of variables. In addition, when using the VRP4$_\geq$ formulation the dual solution space is considerably reduced since dual variables are restricted to non-negative values only.

One of the main drawbacks of models VRP4 and VRP4$_\geq$ is represented by the huge number of variables, which, in non–tightly constrained instances with dozens of customers, may easily run into the billions. Thus, one has to resort to a *Column Generation* (CG) approach to solve the linear programming relaxation of these models, as described in detail in Bramel and Simchi-Levi [15]. The CG method starts from a small subset of routes Ω' and solves the linear relaxation of the corresponding reduced model VRP4' (or VRP4'$_\geq$) deriving the optimal dual variables associated with the constraints. Given the dual information, the CG problem (also called the *pricing* problem) amounts to finding the route not in Ω' with the most negative reduced cost or proving that no such route exists. In this latter case the current solution to the linear model VRP4' is the optimal linear relaxation of VRP4 as well and the process terminates. Otherwise the route returned by CG is added to Ω' and a new iteration is performed. The resulting bound is typically very tight, and this motivated recent extensive research on this approach, leading to the current Branch-and-Cut-and-Price algorithms, which greatly outperformed the early approaches described in this chapter. Since Branch-and-Cut-and-Price algorithms for the CVRP are described in Chapter 3 we limit our exposition here to the first seminal papers that opened this fruitful research direction.

The first of these approaches is due to Agarwal, Mathur, and Salkin [3], who considered a relaxation of model VRP4 with an unlimited number of vehicles, i.e., not including constraint (2.11). To solve the resulting model, they implemented a CG approach in which the pricing problem is faced through a dedicated Branch-and-Bound algorithm. Within this algorithm, a lower bound on the reduced cost of the route is obtained by solving an appropriate knapsack problem. Agarwal, Mathur, and Salkin used their algorithm to solve seven Euclidean CVRP instances with up to 25 customers on a IBM 370 (0.2-0-4 Mflops). Interesting alternative ways of computing the lower bounds used within the CG approach were proposed by Bixby, Coullard, and Simchi-Levi [13], who used a cutting plane algorithm for a suitably defined Prize-Collecting TSP, as well as the famous approach based on dynamic programming by Desrochers, Desrosiers, and Solomon [23], which inspired most pricing schemes used in current Branch-and-Cut-and-Price algorithms. Finally, we recall here the additive approach by Hadjiconstantinou, Christofides, and Mingozzi [33], presented in Section 2.2.3, used to compute approximate solutions of the dual problem associated with model VRP4 that yields tight lower bounds for SCVRP.

2.4 ▪ Branch-and-Cut Algorithms

In this section we review the main research works on the Branch-and-Cut algorithms for the SCVRP realized from 1980 to 2005. They are based on the seminal work by Laporte, Nobert, and Desrochers [40], who introduced the two-index formulation VRP2 of the SCVRP reported in Section 2.2 and described a first Branch-and-Cut algorithm for its solution. In this section, we build upon a brief review of their work to describe the developments proposed during the following 20 years. We mainly describe additional cuts and the associated separation procedures.

In their article, Laporte, Nobert, and Desrochers [40] consider a relaxation of model VRP2 in which the GSECs (2.4), which impose the capacity requirements, are removed together with the restrictions on the integrality of the variables. Given the optimal solution of the relaxation, either the solution is feasible for the SCVRP and the algorithm terminates, or it is a non-feasible solution. Thanks to heuristic separation procedures (see below), they identify violated capacity inequalities (2.4). They compute $r(S)$ as the smallest number of vehicles required to serve vertices in S : $r(S) = \lceil \sum_{i \in S} q_i / Q \rceil$. At the root node, Gomory cuts are also introduced. Next, they add all generated constraints to the relaxed model and reiterate. When they are not able to detect such constraints, they create subproblems by branching on a fractional variable.

In 1995, Augerat described a Branch-and-Cut algorithm in his PhD thesis [7], which included for the first time valid inequalities not present in the model. Augerat separated four families of inequalities (see below): (i) the rounded capacity inequalities; (ii) the generalized capacity constraints; (iii) the comb inequalities; (iv) the hypotour inequalities. Moreover, a tabu search–based heuristic was used to generate an initial upper bound and to update it on the basis of the fractional solutions visited within the course of the algorithm. Last, Augerat explored various branching schemes based on constraints. Given a fractional solution, he identified a subset of vertices S such that $x(\delta(S)) \approx 2k + 1 + \epsilon$, where k is integer and ϵ takes a real value. Then, two subproblems were created by imposing $x(\delta(S)) \leq 2k$ and $x(\delta(S)) \geq 2(k+1)$. Augerat performed extensive computational experiments to determine the best value for ϵ setting $k = 1$. He also considered strategies where ϵ is within some interval and some additional criterion is considered, such as the cardinality of S, the total demand of the vertices in S, and the distance from S to the depot. He concluded that the best strategy consists of selecting the best set S among those identified by applying independently each simple strategy considered. His computational results illustrated that this strategy clearly outperformed the simple strategy based on a single variable. However, a strategy mixing branching on variables and branching on constraints (as described before) led to similar results.

In 2003, Ralphs et al. [51] proposed a Branch-and-Cut algorithm following a different approach. First, they separated the capacity constraints thanks to three heuristics. When the heuristics failed to identify a violated inequality, they proposed using a decomposition algorithm to find additional constraints. First, the original network is expanded through the addition of $|K|-1$ copies of the depot with the corresponding edges as described in Section 2.2. Given a fractional solution on the extended graph, the decomposition algorithm aims to determine whether or not this solution can be written as a convex combination of Hamiltonian cycles. When it succeeds, the Hamiltonian cycles are inspected to find violated capacity inequalities. When it fails, the branching step is invoked. Finally, when the current fractional solution cannot be decomposed, a Farkas inequality is generated. The decomposition algorithm requires knowing a priori the set of the Hamiltonian cycles defined on the extended network. First, an enumerative search is used to generate a preset number of cycles. Then a column generation algorithm is invoked to generate dynamically additional cycles. Ralphs et al. [51] found that the most efficient branching scheme was to branch on variables using a strong branching strategy.

A new Branch-and-Cut algorithm for the CVRP was introduced by Lysgaard, Letchford, and Eglese [43] in 2004. Their algorithm relies on new separation procedures for valid inequalities already known: (i) the rounded capacity inequality; (ii) the framed capacity inequalities; (iii) the strengthened comb inequalities; (iv) the hypotour inequalities. Homogeneous multistar and partial multistar inequalities are also separated according to heuristics described in a previous paper (Letchford, Eglese, and Lysgaard [41]). To perturb the current fractional solution, mixed-integer Gomory cuts are also introduced once at the root node. This may lead to separate additional inequalities thanks to the separation heuristics used. The branching scheme is analogous to the one considered in Augerat [7]. Several sets S_i such that $x(\delta(S_i)) \approx 3$ for $i = 1, \ldots, t$ are identified and ordered according to $|x(\delta(S)) - 3| / \sum_{j \in S} q_j$. Following this order, the lower bounds LB_1 and LB_2 associated with the branches $x(\delta(S)) \leq 2$ and $x(\delta(S)) \geq 4$ are computed. The set leading to the best lower bound $(min(LB_1, LB_2))$ is selected. $max(LB_1, LB_2)$ is used to break a tie.

Other Branch-and-Cut algorithms were proposed during this period. Achuthan, Caccetta, and Hill [1] developed a method which relies on the separation of rounded capacity constraints thanks to heuristics. An improved version based on the separation of additional inequalities related to multistar constraints was described in a subsequent paper by Achuthan, Caccetta, and Hill [2]. Blasum and Hochstättler [14] studied three families of valid inequalities for which separation procedures were presented: (i) the multistar inequalities; (ii) the pathbin inequalities; (iii) the hypotour inequalities. However, most inequalities considered in these classes are less general than those considered in the papers described above and the implemented Branch-and-Cut algorithm is very similar to the approach by Augerat [7]. Baldacci, Hadjiconstantinou, and Mingozzi [9] presented a new integer programming formulation based on a two-commodity network flow approach. Since flow variables can be expressed in terms of arc variables, all valid inequalities for model VRP2 are also valid for the new model. Thus, the authors derived capacity constraints, separated as in [7]. Finally, when all customers have unit demand, two Branch-and-Cut algorithms were described by Araque et al. [6] and by Ghiani, Laporte, and Semet [30].

2.4.1 ▪ Families of Cuts

In this section, we present valid inequalities for the polytope of the CVRP defined as the convex hull of all solutions of the CVRP described by the two-index flow formulation. Since the dimension of the polytope is a complex function of the parameters of the problem, it is difficult to prove whether these valid constraints are facet-defining (see, e.g., Campos, Corberán, and Mota [16] when all the demands are equal). To derive facial properties, relaxations of the polytope of the CVRP are frequently considered. The most common relaxation is the graphical relaxation introduced by Cornuéjols and Harche [21]. The *Graphical Vehicle Routing Problem* (GrVRP) is as the CVRP except that each customer is visited on at least one route but served on exactly one route. The polytope of the CVRP is a face of the polyhedron of the GrVRP. Several valid inequalities described below have been proved to be facet-defining for this polyhedron (see, e.g., Cornuéjols and Harche [21], De Vitis, Harche, and Rinaldi [22], and Blasum and Hochstättler [14]).

TSP-Related Valid Inequalities. A first attempt to propose valid inequalities was done by generalizing constraints which were first developed for the TSP. A general result was obtained by Naddef and Rinaldi [46]. They showed that every constraint valid for the polytope of the symmetric TSP, put into a tight triangular form, is a valid inequality for the CVRP. For $a, x \in \Re^{|E|}$ and $b \in \Re$, an inequality $ax \geq b$ is said to be in tight triangular form if for every triplet of edges $((i,j),(i,k),(j,k))$, the triangle inequality is satisfied, $a_{(i,j)} \leq a_{(i,k)} + a_{(k,j)}$, and for all $i \in V$, two vertices $i_1, i_2 \in V$ exist such that $a_{(i_1,i_2)} = a_{(i,i_1)} + a_{(i,i_2)}$. Naddef and Rinaldi [46] showed that every constraint valid for the symmetric TSP can be put into the tight triangular form.

Capacity Constraints. The capacity constraints are constraints which can be expressed as in (2.4). Depending on how $r(S)$ is computed, this set of constraints has different names. We will see in the next section that these different classes can be more or less easy to separate. If we consider the lowest value for the right-hand side, $r(S) = \sum_{i \in S} q_i / Q$, they are named *fractional capacity inequalities*. When the previous value is rounded up, $r(S) = \lceil \sum_{i \in S} q_i / Q \rceil$, we obtain the *rounded capacity inequalities*. Given S, a valid value for $r(S)$ is the optimal value of the bin-packing problem for which the weights of the objects are equal to q_i for $i \in S$ and the bin capacity is set to Q. These constraints are called *weak capacity inequality*. In general, they are not supporting hyperplanes since the size of the fleet and the demand outside S are not considered. Last, if $r(S)$ is equal to the minimum number of vehicles required to serve the vertices of S when $|K|$ vehicles are available, the resulting inequalities are *global capacity constraints*.

More formally, $\mathbf{P} = \left\{P_1, \ldots, P_{|K|}\right\}$ is a $|K|$-partition of N if $\mathbf{P}$ is a partition of N satisfying $\sum_{j \in P_i} q_j \leq Q$ for $1 \leq i \leq |K|$. Let $\mathscr{P}$ be the set of the $|K|$-partitions. For every non-empty subset S of N and for every $\mathbf{P} \in \mathscr{P}$, we define $\beta_P(S)$ as the number of vehicles required to serve the vertices in S according to $\mathbf{P}$, i.e.,

$$\beta(\mathbf{P}, S) = |\{i : 1 \leq i \leq |K|, P_i \cap S \neq \emptyset\}|.$$

Then, we obtain $r(S) = \min_{P \in \mathscr{P}} (\beta(\mathbf{P}, S))$. Clearly, these four expressions provide increasing values for $r(S)$. Cornuéjols and Harche [21] provide an example where the inequalities are strict for the last three expressions. By construction, the global capacity constraints are supporting hyperplanes.

Augerat [7] considers an extended version of the global capacity constraints: the *generalized capacity constraints* (see also De Vitis, Harche, and Rinaldi [22]). Let $\mathbf{S} = \{S_1, \ldots, S_t\}$

be a set of t disjoint subsets of N; then the following inequality is valid for the CVRP:

$$\sum_{i=1}^{t} x(\delta(S_i)) \geq 2 \min_{P \in \mathscr{P}} \left(\sum_{i=1}^{t} \beta(\mathbf{P}, S_i) \right). \tag{2.13}$$

The generalized capacity constraint (2.13) dominates the aggregation of the global capacity constraints defined on S_i for $i = 1, \ldots, t$. Since the right-hand side is difficult to evaluate, a weaker form of the generalized capacity constraint (2.13) has been considered in Augerat [7]. It consists of replacing the right-hand side with a lower bound based on the solution of a bin-packing problem. Let $\mathbf{S}$ be such that $\sum_{j \in S_i} q_j \leq Q$ for $1 \leq i \leq t$. $r(N|S_1, \ldots, S_t)$ is the optimal value of the bin-packing problem where the bin capacity is Q and where an object is associated with each S_i with a weight equal to the total demand of the vertices of S_i, and an object is associated with each vertex j in $N \backslash (\bigcup_{i=1}^{t} S_i)$ with a weight equal to q_j. Then the following value is a lower bound on the right-hand side of (2.13):

$$2(t + r(N|S_1, \ldots, S_t) - |K|).$$

Framed Capacity Inequalities. These constraints, introduced by Augerat [7], are an extension of the weak generalized capacity constraints. This class of constraints is defined for all $H, S_1, \ldots, S_t \subseteq N$ such that

$$S_i \subset H, \qquad i = 1, \ldots, t, \tag{2.14}$$

$$S_i \cap S_j = \emptyset, \qquad i \neq j \; i, j = 1, \ldots, t, \tag{2.15}$$

$$\sum_{k \in S_i} q_k \leq Q, \qquad i = 1, \ldots, t. \tag{2.16}$$

The framed capacity inequality is expressed as follows:

$$x(\delta(H)) + \sum_{i=1}^{t} x(\delta(S_i)) \geq 2(t + r(H|S_1, \ldots, S_t)), \tag{2.17}$$

where $r(H|S_1, \ldots, S_t)$ is defined as above for set H. Intuitively, this constraint means that if the capacity constraints are tight for each $S_i \subset H$, then $r(H|S_1, \ldots, S_t)$ is a lower bound on the number of vehicles required to serve the vertices in H. Lysgaard, Letchford, and Eglese [43] considered an extended variant of these constraints in which the hypothesis on the total demand for set S_i (2.16) is relaxed. Then, the right-hand side of the framed capacity inequality becomes

$$2(r(H|S_1, \ldots, S_t) + \textstyle\sum_{i=1}^{t} r(S_i)).$$

Last, Augerat [7] proposed additional but more complex constraints of this family.

Comb Inequalities. This class of constraints was proposed by Chvàtal [19] and Grötschel and Padberg [32] for the symmetric TSP. Since comb inequalities can be put into the tight triangular form, they are valid inequalities for the CVRP (see above). Considering the CVRP, they can be expressed as follows. Let $H, T_1, \ldots, T_t \subset N$ be a handle and the associated teeth such that $t \geq 3$ and odd, $H \cap T_i \neq \emptyset$ and $T_i \backslash H \neq \emptyset$ for $i = 1, \ldots, t$, and $T_i \cap T_j = \emptyset$ for $i, j = 1, \ldots, t, i \neq j$. Then the comb inequality is (see Augerat [7])

$$x(\delta(H)) \geq (t+1) - \sum_{i=1}^{t} (x(\delta(T_i)) - 2). \tag{2.18}$$

Assuming that the subtour elimination constraints: $x(\delta(T_i)) \geq 2$ are tight for $i = 1,\ldots,t$, the comb inequality states that $x(\delta(H))$ is at least $(t+1)$. To derive such an inequality, subtour elimination constraints on $T_i, T_i \setminus H, T_i \cap H$ for $i = 1,\ldots,t$ are summed up as well as degree and non-negativity inequalities. Comb inequalities can be strengthened in the CVRP case by taking into account the packing structure. Thus, Laporte and Nobert [38] consider the capacity constraints (2.4) instead of the subtour elimination constraints. Assuming that for every tooth T_i the following inequality holds, $r(T_i \setminus H) + r(T_i \cap H) > r(T_i)$, they obtain the strengthened comb inequality

$$x(\delta(H)) \geq (t+1) - \sum_{i=1}^{t}(x(\delta(T_i)) - 2r(T_i)). \tag{2.19}$$

The explanation of the constraint is as in the previous case. Constraint (2.19) is more or less strong depending on how $r(S)$ is computed (see above) for $S = H, T_1,\ldots,T_t$. In their article, Laporte and Nobert [38] consider a lower bound or the optimal value of the bin-packing problem associated with S.

Other adaptations of the comb inequalities have been proposed. Araque [4] proposed several extended comb inequalities for the CVRP with unit demands. Some of them were adapted to the general case by Lysgaard, Letchford, and Eglese [43].

Hypotour Inequalities. Also these inequalities were introduced by Augerat [7]. This family of constraints aims to identify subnetworks of G which cannot include feasible solutions for the CVRP. More formally, let $G' = (V', E')$ be a subgraph of G such that G' does not contain any feasible solution for the CVRP, but a solution can be identified on $G' \cup \{e\}$ for all $e \in E \setminus E'$. Since the number of edges in any solution of the CVRP is $|N| + |K|$, the *hypotour inequality* $x(E') \leq |N| + |K| - 1$ is valid for the CVRP. Let $F = E \setminus E'$ define a subset of edges such that any feasible solution contains at least one edge of F. The hypotour inequality can be rewritten as $x(F) \geq 1$. Augerat [7] proposed various families of subsets F. The simplest one is as follows. Consider a vertex v and a tree T such that v is not a leaf of T and the total demand on every path including v between two leaves strictly exceeds Q. If $E(\bar{T})$ is the set of edges not present in T and incident with non-pendant vertices of T, then $x(E(\bar{T})) \geq 1$ is a valid inequality.

Augerat [7] introduced also the *extended hypotour inequalities*. Lysgaard, Letchford, and Eglese [43] considered one of them, called a *2-edges extended hypotour inequality*. For any subset $W \subset N$, $e_1, e_2 \in \delta(W)$, and $F \subset E$ as above, the following inequality is valid:

$$x(\delta(W)) + 2x(F) \geq 2(x_{e_1} + x_{e_2}). \tag{2.20}$$

This constraint states that at least one edge from F must be present in any feasible solution as soon as $x_{e_1} = x_{e_2} = 1$.

Multistar Inequalities. These constraints were proposed by Araque, Hall, and Magnanti [5] in the case of the CVRP with unit demands. Let S and T be two set of vertices with $S \subset N$ and $T \subset N \setminus S$. For α, β, γ given, any valid inequality of the form

$$\alpha x(E(S)) + \beta x(\delta(S) \cap \delta(T)) \leq \gamma \tag{2.21}$$

is called a *multistar inequality*. Araque, Hall, and Magnanti [5] identify three types of such constraints. In their paper, Letchford, Eglese, and Lysgaard [41] propose extended versions of this class of inequalities for the case of general demands. The main idea consists of first computing an upper bound on the value of $x(E(S))$. Then, for each value of

$x(E(S))$, an upper bound on the $x(\delta(S) \cap \delta(T))$ is evaluated. The multistar inequalities are obtained by determining the convex hull of the resulting set of points in the $(x(E(S)),$ $x(\delta(S) \cap \delta(T)))$ plane. Note that related inequalities called *generalized large multistar inequalities* were proposed by Gouveia [31].

2.4.2 ▪ Separation Procedures

TSP-Related Valid Inequalities. Separation procedures for constraints that generalized TSP valid inequalities are analogous to those proposed in this context. Naddef and Thienel [48, 49] present a detailed description of them.

Capacity Constraints. Fractional capacity constraints are simple to separate. Indeed, as subtour elimination constraints for the TSP, they can be identified by solving flow problems. Separating all other types of capacity constraints is much more challenging since the separation problem turns to be NP-complete (see Augerat [7]). This is the reason why various heuristics have been suggested.

A first approach introduced by Laporte, Nobert, and Desrochers [40] is based on the decomposition of a support graph in connected components. More precisely, consider the support graph $G' = (N, E')$, where E' is the set of all edges e in $E \setminus \delta(0)$ for which $x_e > 0$. For each connected component S of G', they checked whether the associated rounded capacity inequality is violated or not. If it is not the case, the vertex, which can lead to a violation, is excluded from S and a new check is performed while there are vertices remaining in S.

Several heuristics are based on *contracted* graphs. Given the support graph, a contracted graph is obtained by identifying a subset $S \subset N$ of adjacent vertices and by replacing S by a *supervertex* s. The demand of s is equal to the total demand of the vertices of S, and for $i \in S$, $j \notin S$ edges (i, j) are replaced by an edge (s, j) with a weight equal to $x(\delta(S) \cap \delta(j))$. A simple greedy heuristic proposed by Augerat [7] consists of building a series of contracted graphs obtained by successive edge contractions ($|S| = 2$). The selected edge is chosen among the edges non-incident to the depot and with a weight greater than or equal to one. After each contraction, the rounded capacity constraint associated with the supervertex formed is checked to be violated or not. Variants of this method were proposed by Ralphs et al. [51] and by Lysgaard, Letchford, and Eglese [43]. Based on this heuristic, Augerat et al. [8] proposed an adaptation of the tabu metaheuristic for identifying violated rounded capacity constraints.

The identification of *generalized capacity* and *framed capacity constraints* (see below) is NP-hard, as the computation of the right-hand sides requires the solution of bin-packing problems. Augerat [7] developed greedy procedures to separate generalized capacity constraints. These routines are based on contracted graphs. At each step, a partition of N is obtained by considering the nodes of the current contracted graph, and the identification of a violated inequality is performed by solving the associated bin-packing problem.

Framed Capacity Inequalities. Augerat [7] described a greedy separation procedure for the class of framed capacity constraints based on the routines developed for the generalized capacity constraints. The main difference lies in the selection of the initial set of vertices used to generate contracted graphs. While N is the set considered in the previous case, here the vertices are randomly selected to constitute the set H of vertices of a framed capacity inequality.

Lysgaard, Letchford, and Eglese [43] implemented an enumeration procedure to separate these constraints. First, they consider as H a connected component, or a subset of such a component, of the support graph of the current solution. Next, they enumerate partitions of this set through a search tree by imposing and forbidding edge contractions. The bin-packing problem associated with each partition is solved to identify violated inequalities.

Comb Inequalities. The complexity of comb inequality separation remains unknown for the CVRP. Comb inequalities were identified by Augerat [7] and by Lysgaard, Letchford, and Eglese [43] using similar heuristics to that proposed by Padberg and Rinaldi [50] for the TSP. In both cases, the value of $r(S)$ (see (2.19)) was given by $r(S) = \lceil \sum_{i \in S} q_i / Q \rceil$. For the TSP, more effective approximate separation routines were proposed by Naddef and Thienel [48] as well as a polynomial separation algorithm for a subclass of comb inequalities by Fleischer, Letchford, and Lodi [29].

Hypotour Inequalities. For the hypotour inequalities based on trees described above, Augerat [7] proposed an enumerative procedure. Given the support graph of the current solution and a vertex v, all paths including v with a total demand less than Q are identified. If none of them is connected with the depot, a violated hypotour constraint is identified since at least one edge not present in the current solution must be included in a feasible solution. This set of edges can be refined by considering only edges, not present in the current solution, which provide a feasible extension to one of the enumerated paths. Since this procedure relies on a complete enumerative scheme, it may require significant computation times. However, the number of edges in the support graph is usually smaller than in the original graph, and Augerat [7] indicated a number of enhancements to increase the efficiency of the algorithm. To separate the 2-edges extended hypotour inequality (2.20), Lysgaard, Letchford, and Eglese [43] developed a multi-step heuristic. The main steps are the choice of the set W as well as edges $e_1, e_2 \in \delta(W)$ thanks to a greedy heuristic and the solution of an assignment problem to build a set F.

Multistar Inequalities. The complexity of multistar inequality separation remains unknown. To identify violated multistar inequalities, Letchford, Eglese, and Lysgaard [41] proposed greedy heuristics. Following the notation given above, they first build sets S by including additional vertices in S, as can be done when contracted graphs are built (see Augerat [7]). For S given, they build T by removing from $N \setminus T$ the vertex i with the minimum value of $x(\delta(S) \cap \delta(i))$ repeatedly. Next, for each pair (S, T), they proceed as explained in the constraint description to obtain a set of multistar inequalities. Note that the *generalized large multistar inequalities* can be separated in polynomial time thanks to a network flow algorithm Blasum and Hochstättler [14].

2.4.3 ▪ Comparison of Branch-and-Cut Methods: Do We Have the Appropriate Model?

It is difficult to draw a comparison between the approaches proposed by Augerat [7], Lysgaard, Letchford, and Eglese [43], and Ralphs et al. [51]. Hardware and implementation vary from one algorithm to the other, and results on the same testbed are not available. However, it is possible to give some general trends. It seems that the Branch-and-Cut algorithm developed by Augerat [7] is the least efficient one. Instances proposed by Christofides and Eilon or by Fisher are not frequently solved to optimality. When optimal solutions are obtained, the Branch-and-Cut method due to Lysgaard, Letchford, and Eglese [43] produce them in shorter computation times. The algorithm due to Lysgaard, Letchford, and Eglese [43] compares favorably with the method developed by Ralphs et al. [51]. Indeed, by considering the 50 instances of type A and B originally proposed by Augerat [7], 37 of them are solved optimality by Lysgaard, Letchford, and Eglese [43], while the optimal solutions of 24 instances are obtained by Ralphs et al. [51]. The largest instances solved by both algorithms involved 100 customers and 7 to 8 vehicles, although this was obtained sometimes with prohibitive computation times. For example, the instance E076-08s (also known as E-n76-k8) was solved to optimality by the parallel code developed by Ralphs et al. [51] on a 700MHz Intel Pentium III Xeon platform in almost two million seconds.

Such results lead to the following comments. Over 25 years, researchers mainly focused their works on the two-index flow formulation. This model is an extended version of the classical formulation for the TSP, and many cuts have been proposed which are TSP-like cuts. While the Branch-and-Cut algorithms based on strong polyhedral results turn out to be very effective for the TSP, the results were somehow deceiving with respect to the CVRP. Clearly, the research efforts were not comparable. But, are there more fundamental questions? When the two-index flow model is used for the CVRP, the problem is viewed mainly as a sequencing problem, whereas other points of view could be consider. For instance, the CVRP could be studied from the viewpoint of a packing problem. More generally speaking, did we work with the appropriate model?

2.5 ▪ Conclusions and Future Research Directions

In this chapter we reviewed the most important exact methods that were proposed for the solution of CVRP during more than 30 years from the seventies out to about 2005. This was a very active research period in which all new techniques developed for the exact solution of hard combinatorial problems, from Branch-and-Bound to column generation and to Branch-and-Cut, were fruitfully applied to CVRP. Although the current best algorithms, all belonging to the Branch-and-Cut-and-Price family, greatly improved the performance of the methods presented in this chapter, such a huge research body still has a great value. This is not only because Branch-and-Cut-and-Price combines and improves what was done by previous methods, but also because early algorithms, particularly those using combinatorial relaxations, remain somehow much simpler to implement, thus still being very useful to define viable approaches for novel variants of VRP or large-scale instances that are intractable for linear programming–based approaches. In addition, some space for further improvement remains with respect to Branch-and-Bound and Branch-and-Cut methods. Two main research avenues are, in fact, still open: (i) the development of new models, and (ii) the identification of new cut families, as well as more effective separation procedures and branching strategies that will allow one to compute better bounds more quickly. We also note that the complexity of the separation problem for most of the cut families we presented has not been studied so far. The potential of the classical approaches we examined in this chapter is therefore far from being fully exhausted.

Bibliography

[1] N. R. ACHUTHAN, L. CACCETTA, AND S. P. HILL, *A new subtour elimination constraint for the vehicle routing problem*, European Journal of Operational Research, 91 (1996), pp. 573–586.

[2] ——, *An improved branch-and-cut algorithm for the capacitated vehicle routing problem*, Transportation Science, 37 (2003), pp. 153–169.

[3] Y. Agarwal, K. Mathur, and H. Salkin, *A set-partitioning-based exact algorithm for the vehicle routing problem*, Networks, 19 (1989), pp. 731–749.

[4] J. R. Araque, *Contributions to the polyhedral approach to vehicle routing*, Discussion Paper 90-74, CORE, Catholic University of Louvain La Neuve, Belgium, 1990.

[5] J. R. Araque, L. Hall, and T. L. Magnanti, *Capacitated trees, capacitated routing and associated polyhedra*, Discussion Paper 90-61, CORE, Catholic University of Louvain La Neuve, Belgium, 1990.

[6] J. R. Araque, G. Kudva, T. L. Morin, and J. F. Pekny, *A branch-and-cut algorithm for vehicle routing problems*, Annals of Operations Research, 50 (1994), pp. 37–59.

[7] P. Augerat, *Approche Polyèdrale du Problème de Tournées de Véhicules*, PhD thesis, Institut National Polytechnique de Grenoble, Grenoble, France, 1995.

[8] P. Augerat, J. Belenguer, E. Benavent, A. Corberán, and D. Naddef, *Separating capacity inequalities in the CVRP using tabu search*, European Journal of Operational Research, 106 (1998), pp. 546–557.

[9] R. Baldacci, E. Hadjiconstantinou, and A. Mingozzi, *An exact algorithm for the capacitated vehicle routing problem based on a two-commodity network flow formulation*, Operations Research, 52 (2004), pp. 723–738.

[10] R. Baldacci, P. Toth, and D. Vigo, *Exact algorithms for routing problems under vehicle capacity constraints*, Annals of Operations Research, 175 (2010), pp. 213–245.

[11] ——, *Exact solution of the capacitated vehicle routing problem*, in Wiley Encyclopedia in Operations Research and Management Science, vol. 3, Wiley, New York, 2011, pp. 1795–1807.

[12] M. Balinski and R. Quandt, *On an integer program for a delivery problem*, Operations Research, 12 (1964), pp. 300–304.

[13] A. Bixby, C. Coullard, and D. Simchi-Levi, *The capacitated prize-collecting traveling salesman problem*, Working Paper, Department of Industrial Engineering and Engineering Management, Northwestern University, Evanston, IL, 1997.

[14] U. Blasum and W. Hochstättler, *Application of the branch and cut method to the vehicle routing problem*, Technical Report ZPR2000-386, ZPR, Universität zu Köln, Germany, 2000.

[15] J. Bramel and D. Simchi-Levi, *Set-covering-based algorithms for the capacitated VRP*, in The Vehicle Routing Problem, P. Toth and D. Vigo, eds., SIAM, Philadelphia, 2002, ch. 4, pp. 85–108.

[16] V. Campos, A. Corberán, and E. Mota, *Polyhedral results for a vehicle routing problem*, European Journal of Operational Research, 52 (1991), pp. 75–85.

[17] N. Christofides and S. Eilon, *An algorithm for the vehicle dispatching problem*, Operational Research Quarterly, 20 (1969), pp. 309–318.

[18] N. Christofides, A. Mingozzi, and P. Toth, *Exact algorithms for the vehicle routing problem based on the spanning tree and shortest path relaxations*, Mathematical Programming, 20 (1981), pp. 255–282.

[19] V. CHVÀTAL, *Edmonds polytopes and weakly Hamiltonian graphs*, Mathematical Programming, 5 (1973), pp. 29–40.

[20] J.-F. CORDEAU, G. LAPORTE, M. W. P. SAVELSBERGH, AND D. VIGO, *Vehicle routing*, in Transportation, C. Barnhart and G. Laporte, eds., vol. 14 of Handbooks in Operations Research and Management Science, Elsevier, 2007, ch. 6, pp. 367–428.

[21] G. CORNUÉJOLS AND F. HARCHE, *Polyhedral study of the capacitated vehicle routing problem*, Mathematical Programming, 60 (1993), pp. 21–52.

[22] A. DE VITIS, F. HARCHE, AND G. RINALDI. *Generalized capacity inequalities for vehicle routing problems*. unpublished manuscript, 2000.

[23] M. DESROCHERS, J. DESROSIERS, AND M. SOLOMON, *A new optimization algorithm for the vehicle routing problem with time windows*, Operations Research, 40 (1992), pp. 342–354.

[24] J. DONGARRA, *Performance of various computers using standard linear equations software*, Technical Report CS–89–85, University of Tennessee, Knoxville, TN, 2013.

[25] M. FISCHETTI AND P. TOTH, *A new dominance procedure for combinatorial optimization*, Operations Research Letters, 7 (1988), pp. 181–187.

[26] ——, *An additive bounding procedure for combinatorial optimization problems*, Operations Research, 37 (1989), pp. 319–328.

[27] M. FISCHETTI, P. TOTH, AND D. VIGO, *A branch-and-bound algorithm for the capacitated vehicle routing problem on directed graphs*, Operations Research, 42 (1994), pp. 846–859.

[28] M. FISHER, *Optimal solution of vehicle routing problems using minimum k-trees*, Operations Research, 42 (1994), pp. 626–642.

[29] L. FLEISCHER, A. LETCHFORD, AND A. LODI, *Polynomial-time separation of a superclass of simple comb inequalities*, Mathematics of Operations Research, 31 (2006), pp. 696–713.

[30] G. GHIANI, G. LAPORTE, AND F. SEMET, *The black and white traveling salesman problem*, Operations Research, 54 (2006), pp. 366–378.

[31] L. GOUVEIA, *A result on projection for the vehicle routing problem*, European Journal of Operational Research, 85 (1995), pp. 610–624.

[32] M. GRÖTSCHEL AND M. W. PADBERG, *On the symmetric travelling salesman problem* I*: Inequalities*, Mathematical Programming, 16 (1979), pp. 265–280.

[33] E. HADJICONSTANTINOU, N. CHRISTOFIDES, AND A. MINGOZZI, *A new exact algorithm for the vehicle routing problem based on q-paths and k-shortest paths relaxations*, Annals of Operations Research, 61 (1995), pp. 21–43.

[34] M. HELD AND R. KARP, *The traveling salesman problem and minimum spanning trees: Part* II, Mathematical Programming, 1 (1971), pp. 6–25.

[35] M. HELD, P. WOLFE, AND M. CROWDER, *Validation of the subgradient optimization*, Mathematical Programming, 6 (1974), pp. 62–88.

[36] G. LAPORTE, *The vehicle routing problem: An overview of exact and approximate algorithms*, European Journal of Operational Research, 59 (1992), pp. 345–358.

[37] G. LAPORTE, H. MERCURE, AND Y. NOBERT, *An exact algorithm for the asymmetrical capacitated vehicle routing problem*, Networks, 16 (1986), pp. 33–46.

[38] G. LAPORTE AND Y. NOBERT, *Comb inequalities for the vehicle routing problem*, Methods of Operations Research, 51 (1984), pp. 271–276.

[39] ——, *Exact algorithms for the vehicle routing problem*, Annals of Discrete Mathematics, 31 (1987), pp. 147–184.

[40] G. LAPORTE, Y. NOBERT, AND M. DESROCHERS, *Optimal routing under capacity and distance restrictions*, Operations Research, 33 (1985), pp. 1050–1073.

[41] A. N. LETCHFORD, R. W. EGLESE, AND J. LYSGAARD, *Multistars, partial multistars and the capacitated vehicle routing problem*, Mathematical Programming, 94 (2002), pp. 21–40.

[42] J. D. C. LITTLE, K. G. MURTY, D. W. SWEENEY, AND C. KAREL, *An algorithm for the traveling salesman problem*, Operations Research, 11 (1963), pp. 972–989.

[43] J. LYSGAARD, A. N. LETCHFORD, AND R. W. EGLESE, *A new branch-and-cut algorithm for the capacitated vehicle routing problem*, Mathematical Programming, 100 (2004), pp. 423–445.

[44] C. MARTINHON, A. LUCENA, AND N. MACULAN, *A relax and cut algorithm for the vehicle routing problem*, Technical Report RT-05/00, Universidade Federal Fluminense, Niterói, Brasil, 2000.

[45] D. MILLER, *A matching based exact algorithm for capacitated vehicle routing problems*, ORSA Journal on Computing, 7 (1995), pp. 1–9.

[46] D. NADDEF AND G. RINALDI, *The graphical relaxation: A new framework for the symmetric traveling salesman polytope*, Mathematical Programming, 58 (1993), pp. 53–88.

[47] ——, *Branch-and-cut algorithms for the capacitated VRP*, in The Vehicle Routing Problem, P. Toth and D. Vigo, eds., SIAM, Philadelphia, 2002, ch. 3, pp. 53–84.

[48] D. NADDEF AND S. THIENEL, *Efficient separation routines for the symmetric traveling salesman problem* I*: General tools and comb separation*, Mathematical Programming, 92 (2002), pp. 237–255.

[49] ——, *Efficient separation routines for the symmetric traveling salesman problem* II*: Separating multi-handle inequalities*, Mathematical Programming, 92 (2002), pp. 257–285.

[50] M. PADBERG AND G. RINALDI, *Facet identification for the symmetric traveling salesman polytope*, Mathematical Programming, 47 (1990), pp. 219–257.

[51] T. K. RALPHS, L. KOPMAN, W. R. PULLEYBLANK, AND L. E. TROTTER, *On the capacitated vehicle routing problem*, Mathematical Programming, 94 (2003), pp. 343–359.

[52] P. TOTH AND D. VIGO, *Models, relaxations and exact approaches for the capacitated vehicle routing problem*, Discrete Applied Mathematics, 123 (2002), pp. 487–512.

[53] ——, *Branch-and-Bound algorithms for the capacitated VRP*, in The Vehicle Routing Problem, P. Toth and D. Vigo, eds., SIAM, Philadelphia, 2002, ch. 2, pp. 29–51.

[54] ——, eds., *The Vehicle Routing Problem*, SIAM, Philadelphia, 2002.