Transportation Science

Publication details, including instructions for authors and subscription information:
http://pubsonline.informs.org

Fifty Years of Vehicle Routing

Gilbert Laporte,

TRANSPORTATION SCIENCE
Vol. 43, No. 4, November 2009, pp. 408–416
ISSN 0041-1655 | EISSN 1526-5447 | 09 | 4304 | 0408

DOI 10.1287/trsc.1090.0301

Fifty Years of Vehicle Routing

Gilbert Laporte
CIRRELT, Distribution Management, HEC Montréal, 3000, Montréal, Québec H3T 2A7, Canada
gilbert@crt.umontreal.ca

The *Vehicle Routing Problem* (VRP) was introduced 50 years ago by Dantzig and Ramser under the title "The Truck Dispatching Problem." The study of the VRP has given rise to major developments in the fields of exact algorithms and heuristics. In particular, highly sophisticated exact mathematical programming decomposition algorithms and powerful metaheuristics for the VRP have been put forward in recent years. The purpose of this article is to provide a brief account of this development.

Key words: vehicle routing problem; traveling salesman problem; exact algorithms; heuristics; metaheuristics; survey
History: Received: August 2009; revision received: September 2009; accepted: September 2009. Published online in *Articles in Advance* October 21, 2009.

1. Introduction

The year 2009 marks the 50th anniversary of first paper published on the *vehicle routing problem* (VRP), under the title "The Truck Dispatching Problem" (Dantzig and Ramser 1959). To commemorate this event we highlight the main contributions in the history of this important problem.

The VRP can be simply defined as the problem of designing least-cost delivery routes from a depot to a set of geographically scattered customers, subject to side constraints. This problem is central to distribution management and must be routinely solved by carriers. In practice several variants of the problem exist because of the diversity of operating rules and constraints encountered in real-life applications. Thus the VRP should perhaps be viewed as a class of problems. However, it is useful from a methodological point of view to work on archetypal versions of the problem. Here we focus on the *classical VRP* formally defined as follows. Let $G = (V, A)$ be a directed graph where $V = \{0, \ldots, n\}$ is the vertex set and $A = \{(i, j): i, j \in V, i \neq j\}$ is the arc set. Vertex 0 represents the depot whereas the remaining vertices correspond to customers. A fleet of m identical vehicles of capacity Q is based at the depot. The fleet size is given a priori or is a decision variable. Each customer i has a nonnegative demand q_i. A cost matrix c_{ij} is defined on A. For simplicity, we consider travel costs, distances and travel times to be equivalent. The VRP consists of designing m vehicle routes such that each route starts and ends at the depot, each customer is visited exactly once by a single vehicle, the total demand of a route does not exceed Q, and the total length of a route does not exceed a preset limit L.

In the symmetric case, i.e., when $c_{ij} = c_{ji}$ for all $(i, j) \in A$, it is customary to work with an edge set $E = \{(i, j): i, j \in V, i < j\}$. In what follows we concentrate on the symmetric case, but some formulations associate variables with arcs and some heuristics apply indiscriminately to both the symmetric and the asymmetric cases. For a review of exact algorithms for the asymmetric VRP, see Toth and Vigo (2002b).

There exists a rich scientific literature on the VRP. For books and survey articles, the reader is referred to Toth and Vigo (2002a), Golden, Raghavan, and Wasil (2008), Laporte and Nobert (1987), Fisher (1995), Laporte and Semet (2002), Gendreau, Laporte, and Potvin (2002), Cordeau et al. (2005, 2007), and Laporte (2007). The two books by Toth and Vigo (2002a) and the book by Golden, Raghavan, and Wasil (2008) also contain a wealth of information on the most common variants of the VRP.

The VRP generalizes the well-known *traveling salesman problem* (TSP) but is much more difficult to solve in practice. Whereas there exist exact algorithms capable of routinely solving TSPs containing hundreds or thousands of vertices (Applegate et al. 2007), this is not the case of the VRP for which the best exact algorithms can only solve instances involving approximately 100 vertices (Fukasawa et al. 2006; Baldacci, Christofides, and Mingozzi 2008). Because real instances of the VRP often exceed this size and solutions must often be determined quickly, most algorithms used in practice are heuristics. In recent years, several powerful metaheuristics have been developed.

In the following three sections of this paper some of the main exact algorithms, classical heuristics, and metaheuristics for the VRP will be reviewed. This will be followed by some conclusions.

2. Exact Algorithms

Over the past 40 years exact algorithms for the VRP have evolved from basic branch-and-bound schemes to highly sophisticated mathematical programming engineering pieces. In what follows we retrace the main steps of this development.

2.1. Branch-and-Bound Algorithms

One of the first known branch-and-bound algorithms for the VRP appeared in "An Algorithm for the Vehicle Dispatching Problem" (Christofides and Eilon 1969), heralding a slight departure from the original Dantzig and Ramser (1959) appellation. In this algorithm, m is an input parameter. The graph is first extended by adding $m-1$ artificial depots to the graph and setting the inter-depot arc costs equal to infinity. An m-TSP is then solved on this graph by branching on arcs, as in the Little et al. (1963) TSP algorithm, except that a shortest spanning tree bound is computed at each node, instead of a relaxed assignment problem. The VRP side constraints are handled through simple fathoming rules. An improvement to this algorithm, based on a modification of the Carpaneto and Toth (1980) TSP algorithm was later proposed by Laporte, Mercure, and Nobert (1986). The Christofides (1976) paper seems to be the first to have used the name "vehicle routing problem." The branch-and-bound algorithm it describes branches on routes rather than arcs. The resulting search tree is therefore rather wide but has a depth limited to m. Neither of these two algorithms was very successful in solving anything but rather small or easy instances.

The performance of early branch-and-bound algorithms was greatly enhanced by the derivation of two sharp lower bounds based on k-degree centre trees (k-DCT) and q-routes (Christofides, Mingozzi, and Toth 1981a). The first computes a tree in which the degree of vertex 0 is equal to k, where $k=2m-y$ and $y \leq m$. The set of all edges belonging to a solution can be partitioned into $\{E_1, E_2, E_3\}$, where E_1 is the set of edges forming a k-DCT, E_2 is a set of y edges incident to 0, and E_3 is a set of $m-y$ edges not incident to 0. The problem is formulated in terms of three sets of variables associated with the three elements of the partition, with linking constraints. Lower bounds are then computed for the subproblem defined by each variable type by incorporating the linking constraints in the objective function in a Lagrangean fashion.

The second lower bound is based on the concept of q-route put forward by Houck et al. (1980). Let W be the ordered set of all possible demands of a potential vehicle route, starting with the smallest value. Let $q(l)$ be the value of the lth element of W and let $\psi_l(i)$ be the value of a least cost route (1) passing through i, (2) starting and ending at the depot, (3) having no two-vertex cycle, and (4) having a demand equal to $q(l)$. Then $\sum_{i=1}^{n} > \min_{l=1,\ldots,|W|} > \{\psi_l(i)q_i/q(l)\}$ is a valid lower bound on the cost of an optimal VRP solution. This bound is the sum, over all customers i, of a lower bound on the contribution made by i to the routing cost. This bound can be improved through an ascent procedure.

Christofides, Mingozzi, and Toth (1981a) have successfully solved instances with $10 \leq n \leq 25$ with these two lower bounds. The bound derived from q-routes is generally better than the k-DCT bound. Using the same bounds, Hadjiconstantinou, Christofides, and Mingozzi (1995) have developed an improved branch-and-bound algorithm capable of optimally solving instances with $n \leq 50$. Later, Fisher (1994) later incorporated the k-DCT lower bound within a branch-and-cut algorithm for a restricted version of the VRP in which return trips between the depot and a customer are disallowed. He was able to successfully solve, within a small tolerance, several instances containing up to 134 vertices.

2.2. Dynamic Programming

Eilon, Watson-Gandy, and Christofides (1971) have formulated the VRP as a dynamic program as follows. Let $c(S)$ be the optimal cost of a single vehicle route through the vertices of $S \subseteq V\backslash\{0\}$. The objective is to minimize $\sum_{r=1}^{m} > c(S_r)$ over all feasible partitions $\{S_1, \ldots, S_m\}$ of $V\backslash\{0\}$. Let $f_k(U)$ be the least cost achievable using k vehicles and delivering to a subset U of $V\backslash\{0\}$. Then

$$f_k(U)=\begin{cases} c(U) & (k=1), \\ \min\limits_{U^* \subseteq U \subseteq V\backslash\{0\}} & \{f_{k-1}(U\backslash U^*)+c(U^*)\}(k>1). \end{cases} \tag{1}$$

The solution cost is $f_m(V\backslash\{0\})$ and the optimal partition corresponds to the optimizing subsets in (1).

The state space can be reduced by using feasibility or dominance criteria. A state-space relaxation algorithm has also been applied by Christofides, Mingozzi, and Toth (1981b). These authors report having solved instances with $10 \leq n \leq 25$ using this technique, but this avenue of research does not seem to have been pursued.

2.3. Vehicle Flow Formulations and Algorithms

Two-index vehicle flow formulations for the VRP are rooted in the work of Laporte and Nobert (1983) and Laporte, Nobert, and Desrochers (1985) and extend the classical TSP formulation of Dantzig, Fulkerson, and Johnson (1954). Let x_{ij} be a 0–1–2 variable equal to the number of times a vehicle travels on edge (i, j). Then the problem is

$$\text{(VF)} \quad \text{minimize} \sum_{(i,j)\in E} c_{ij}x_{ij} \tag{2}$$

$$\text{subject to} \sum_{j=1}^{n} x_{0j}=2m, \tag{3}$$

$$\sum_{i<k} x_{ik} + \sum_{j>k} x_{kj} = 2 \quad (k \in V\setminus\{0\}), \tag{4}$$

$$\sum_{i,j\in S} x_{ij} \le |S| - v(S) \quad (S \subseteq V\setminus\{0\}), \tag{5}$$

$$x_{0j} = 0, 1, 2 \quad (j \in V\setminus\{0\}), \tag{6}$$

$$x_{ij} = 0, 1 \quad (i, j \in V\setminus\{0\}), \tag{7}$$

where $v(S)$ is a lower bound on the number of vehicles needed to service all vertices of S. Constraints (2) and (3) are degree constraints, whereas constraints (4) eliminate subtours and enforce the capacity and route length restrictions. In the presence of capacity constraints, $v(S)$ is often set equal to $\lceil \sum_{i\in S} q_i/Q \rceil$ or is computed by solving a bin packing problem with item weights $q_i (i \in S)$ and bin size Q. Laporte, Nobert, and Desrochers (1985) show how to compute $v(S)$ within a branch-and-cut scheme in the presence of route length constraints.

The model defined by (VF) can be reinforced through the inclusion of valid inequalities. These include so-called generalized capacity constraints, frame capacity constraints, VRP comb inequalities, inequalities combining bin packing and the TSP, and some inequalities based on the stable set problem. For an overview, see Naddef and Rinaldi (2002). These authors have developed a branch-and-cut algorithm enabling them to solve six instances ($22 \le n \le 45$) without branching, and nine others ($51 \le n \le 135$) with some branching. For a more recent branch-and-cut algorithm based on the (VF) formulation, see Lysgaard, Letchford, and Eglese (2004).

A number of three-index vehicle flow formulations have also been put forward. These make use of binary variables x_{ijk} equal to 1 if and only if arc (i, j) is traversed by vehicle k. They do not seem to have been as successful as two-index formulations. Examples are provided by Golden, Magnanti, and Nguyen (1977) and by Fisher and Jaikumar (1981).

2.4. Commodity Flow Formulations and Algorithms

In commodity flow formulations, variables y_{ij} (or y_{ijk}) define the vehicle load carried on arc (i, j). An early example can be found in Gavish and Graves (1979), but these authors report no computational result. A more recent example is the formulation of Baldacci, Hadjiconstantinou, and Mingozzi (2004) which is based on the TSP model of Finke, Claus, and Gunn (1984). This formulation works on an extended graph $\bar{G} = (\bar{V}, \bar{E})$, where $\bar{V} = V \cup \{n+1\}$, $n+1$ is a copy of the depot, and $\bar{E} = E \cup \{(i, n+1)\colon i \in V\}$. A vehicle route is defined as a directed path from 0 to $n+1$. Binary variables x_{ij} are equal to 1 if and only if edge (i, j) is used in the solution, variables y_{ij} represent the vehicle load on (i, j), and variables $y_{ji} = Q - y_{ij}$ represent the empty vehicle space on (i, j) whenever $x_{ij} = 1$.

The formulation is

$$\text{(CF)} \quad \text{minimize} \sum_{(i,j)\in \bar{E}} c_{ij} x_{ij} \tag{8}$$

$$\text{subject to} \sum_{j\in\bar{V}} (y_{ji} - y_{ij}) = 2q_i \quad (i \in V\setminus\{0\}), \tag{9}$$

$$\sum_{j\in V\setminus\{0\}} y_{0j} = \sum_{i\in V\setminus\{0\}} q_i, \tag{10}$$

$$\sum_{j\in V\setminus\{0\}} y_{j0} = mQ - \sum_{i\in V\setminus\{0\}} q_i, \tag{11}$$

$$\sum_{j\in V\setminus\{0\}} y_{n+1,j} = mQ, \tag{12}$$

$$y_{ij} + y_{ji} = Q x_{ij} \quad ((i, j) \in \bar{E}), \tag{13}$$

$$\sum_{i<k} x_{ik} + \sum_{j>k} x_{kj} = 2 \quad (k \in V\setminus\{0\}), \tag{14}$$

$$y_{ij} \ge 0, y_{ji} \ge 0 \quad ((i, j) \in \bar{E}), \tag{15}$$

$$x_{ij} = 0, 1 \quad ((i, j) \in \bar{E}). \tag{16}$$

Constraints (9)–(12) and (15) define consistent flows from 1 to $n+1$, constraints (13) ensure that the y_{ij} and y_{ji} variables are feasible, and constraints (14) are degree constraints. This problem was solved by branch-and-cut, with valid VRP inequalities expressed in terms of the x_{ij} variables. Several instances with $16 \le n \le 135$ were solved optimally. The success rate is very high if $30 \le n \le 60$ and $3 \le m \le 5$.

2.5. Set Partitioning Formulations and Algorithms

A straightforward set partitioning formulation of the VRP was first provided by Balinski and Quandt (1964). Let r denote a route, let a_{ir} be a binary coefficient equal to 1 if and only if vertex $i \in V\setminus\{0\}$ belongs to route r, let c^* be the optimal cost of route r, and let y_k be a binary variable equal to 1 if and only if route r is used in the optimal solution. The problem is then

$$\text{(SP)} \quad \text{minimize} \sum_r c_r^* y_r \tag{17}$$

$$\text{subject to} \sum_r a_{ir} = 1 \quad (i \in V\setminus\{0\}), \tag{18}$$

$$\sum_r y_r = m, \tag{19}$$

$$y_r = 0, 1 \quad (\text{all } r). \tag{20}$$

Strictly speaking, constraints (19) are not part of the standard set partitioning formulation, but they are used by most authors in the context of the VRP.

A direct application of this formulation is impractical because of the large number of potential routes encountered in most nontrivial instances and of the difficulty of computing the c_r^* coefficients which requires solving an exponential number of instances of an NP-hard problem.

A number of column generation algorithms were proposed to solve this problem. The first, by Rao and Zionts (1968) does not seem to have been numerically tested. A second method, by Foster and Ryan (1976) generates routes by dynamic programming but was not run to completion. A full column generation algorithm was developed by Agarwal, Mathur, and Salkin (1989) who solved instances with $15 \leq n \leq 25$.

Two of the most successful VRP algorithms, developed in recent years, make partial use of a set partitioning formulation. The first by Fukasawa et al. (2006) works with the following set partitioning formulation in which the columns correspond to q-routes. Let a_{ijr} be a binary coefficient equal to 1 if and only if edge (i, j) appears in q-route r, y_r, c_r^* are defined as in (SP), and x_{ij} as in (VF). Then the problem is

$$\text{(SP')} \quad \text{minimize} \quad \sum_r c_r^* y_r \tag{21}$$

$$\text{subject to} \quad \sum_r a_{ijr} y_r - x_{ij} = 0 \quad ((i, j) \in E), \tag{22}$$

$$\text{and} \quad (4), (6), (7), (19), \text{and } (20).$$

Combining (VF) and (SP') yields the following formulation:

$$\text{(VF-SP')} \quad \text{minimize} \quad \sum_r \sum_{(i,j)\in E} c_{ij} a_{ijr} y_r \tag{23}$$

$$\text{subject to} \quad \sum_r \sum_{j \in V\setminus\{0\}} a_{ijr} y_r = 2m, \tag{24}$$

$$\sum_r \left(\sum_{i<k} a_{ikr} y_r + \sum_{j>k} a_{jkr} y_r \right) = 2 \quad (k \in V\setminus\{0\}), \tag{25}$$

$$\sum_r \sum_{\substack{i\in S,\, j\notin S \\ \text{or } i\notin S,\, j\in S}} a_{ijr} y_r \geq 2v(S) \quad (S \subseteq V\setminus\{0\}), \tag{26}$$

$$\sum_r a_{ijr} y_r \leq 1 \quad ((i, j) \in E\setminus\{(0, j)\colon j \in V\setminus\{0\}\}), \tag{27}$$

$$y_r = 0, 1 \quad (\text{all } r). \tag{28}$$

A lower bound on the VRP objective is computed by solving the linear relaxation of (VF-SP') to which are added the following valid inequalities proposed by Letchford, Eglese, and Lysgaard (2002) and Lysgaard, Letchford, and Eglese (2004): framed capacity inequalities, strengthened combs, multistar, partial multistar, and generalized multistar inequalities, and hypotour cuts. Embedding this master problem within a branch-and-cut-and-price algorithm, the authors were capable of solving instances containing up to 135 vertices.

The second algorithm, by Baldacci, Christofides, and Mingozzi (2008), works with the (SP) formulation, augmented with some inequalities valid for (VF). This is made possible by using the following identity proved in Baldacci, Hadjiconstantinou, and Mingozzi (2004):

$$x_{ij} = \sum_r a_{ijr} y_r \quad ((i, j) \in E), \tag{29}$$

where if r is the route $(0, j, 0)$, then $a_{0jr} = 2$, and $a_{ijr} = 0$ for all $(i, j) \in E\setminus\{(0, j)\}$; if r contains at least two customer vertices, then $a_{ijr} = 1$ for each edge (i, j) of route r, and $a_{ijr} = 0$ otherwise.

The (SP) formulation can be strengthened through the introduction of valid inequalities. For $S \subseteq V\{0\}$, let $R(S)$ be the set of routes containing at least one customer of S. Then the capacity constraints

$$\sum_{r\in R(S)} y_r \geq v(S) \quad (S \subseteq V\setminus\{0\}) \tag{30}$$

are valid. Also, any valid inequality for the VRP (Naddef and Rinaldi 2002; Letchford, Eglese, and Lysgaard 2002) of the form

$$\sum_{(i,j)\in E} \alpha_{ij} x_{ij} \geq \beta \tag{31}$$

can be expressed in terms of the y_r variables by using (29). Finally valid inequalities for the *Set Partitioning Problem* (Balas and Padberg 1976; Hoffman and Padberg 1993) are also valid for (SP). Specifically, the authors use the clique inequalities. Consider a graph H whose vertices correspond to vehicle routes; two vertices conflict whenever the routes they represent share some customers. Then, for any clique C of H, the inequality

$$\sum_{r\in C} y_r \leq 1 \tag{32}$$

is valid.

Baldacci, Christofides, and Mingozzi (2008) use (SP) together with some of constraints (30), (31), and (32). They compute lower bounds on the dual of the linear relaxation of this problem, obtained by applying three ascent heuristics. The final dual solution is used to generate a reduced problem containing columns of reduced cost between the upper bound and the lower bound achieved. This problem is solved by an integer linear programming solver. This algorithm was capable of solving instances with $37 \leq n \leq 121$ and seems to be slightly better than that of Fukasawa et al. (2006).

3. Classical Heuristics

The seminal Dantzig and Ramser (1959) paper introduced the VRP to the research community and presented the first heuristic for the problem. Essentially, this algorithm iteratively matches vertices, or vertices and partial routes, to form a set of vehicle routes. It does so by solving a sequence of linear programs with variables x_{ij} and x_{ji} equal to 1 if and only if two points (vertices, partial routes) are matched. Each iteration is called a "stage of aggregation." The method is illustrated on a small seven-vertex instance.

Between 1964 and the early 1990s numerous heuristics were put forward. Some are purely constructive but most also include an improvement phase. We call these heuristics "classical" because they do not contain mechanisms allowing the objective function to deteriorate from one iteration to the next. This feature is present in "metaheuristics" which have been developed primarily over the past twenty years.

Heuristics are usually tested on two sets of benchmark instances. The first set, called CMT, was proposed by Christofides, Mingozzi, and Toth (1979) and contains 14 instances with $51 \leq n \leq 199$. The second, called GWKC, was proposed by Golden et al. (1998) and contains 20 instances with $200 \leq n \leq 480$. It is customary to compute optimality gaps with respect to best-known solution values. We now outline some of the most important classical heuristics for the VRP.

3.1. The Savings Algorithm

The savings heuristic put forward by Clarke and Wright (1964) is easy to describe and to implement, and yields reasonably good solutions. This explains its ongoing popularity. It starts with an initial solution made up of n back-and-forth routes $(0, i, 0)$ $(i \in V\backslash\{0\})$. At each iteration, it merges a route ending with i with another route starting with j, maximizing the saving $s_{ij} = c_{i0} + c_{0j} - c_{ij}$, and provided the merge is feasible. The process stops when it is no longer possible to merge routes. The number of vehicles used in the solution is an output of the algorithm. This process is normally followed by a three-opt procedure applied to each route. Applying this algorithm to the CMT instances produced deviations of about 7% within insignificant computing times (Laporte and Semet 2002).

Several enhancements to this algorithm have been proposed, namely multiplying c_{ij} by a positive weight λ (Golden, Magnanti, and Nguyen 1977), optimizing the route merges in a global fashion through the use of a matching algorithm (Altinkemer and Gavish 1991, Wark and Holt 1994), accelerating the savings computation (Paessens 1988), and making use of efficient data structures to speed up the computations (Nelson et al. 1985). The first enhancement is useful in avoiding circumferential routes that tend to occur in the original Clarke and Wright algorithm. Solving matching problems is highly time consuming and not worth the effort in comparison with other heuristics. The last two enhancements are probably of little use these days given the state of the art in computer technology and improvement heuristics.

3.2. Set Partitioning Heuristics

A simple heuristic scheme consists of solving the (SP) formulation with a subset of promising vehicle routes, often called "petals" in this context. This method normally assumes that the vertices are distributed on a plane. An early example of this methodology is the sweep algorithm of Gillett and Miller (1974) in which nonoverlapping petals are sequentially generated by rotating a half-line rooted at the depot, as long as the capacity and route length constraints for a single route are satisfied. More sophisticated schemes in which intersecting or embedded routes are allowed have been put forward by Foster and Ryan (1976), Ryan, Hjorring, and Glover (1993), and Renaud, Boctor, and Laporte (1996). On the CMT testbed the latter method has produced an average deviation of 2.38%.

3.3. Cluster-First, Route-Second Heuristics

The cluster-first, route-second heuristic of Fisher and Jaikumar (1981) first locates *m seeds* and constructs a cluster for each seed so as to minimize the sum of customer-to-seed distances, while satisfying the capacity constraint. This is achieved by solving a *generalized assignment problem* (GAP). A route is then determined on each cluster by solving a TSP. Some procedures for selecting the seeds are described in Baker and Sheasby (1999). It is not clear exactly how the route length constraint is handled within this method but some of the CMT instances solved by Fisher and Jaikumar (1981) do contain a length constraint. Exact comparisons with other algorithms are difficult to make because the distance rounding convention used in the experiments is not specified. This method is natural in that it follows the two phases often applied by human dispatchers. It is also interesting from a methodological point of view because it can benefit from algorithmic improvements for the GAP or for the TSP.

3.4. Improvement Heuristics

Two broad types of methods can be employed to postoptimize a VRP solution. *Intraroute* moves consist of improving each route separately by means of a TSP algorithm, whereas *interroute* moves act on several routes simultaneously. It is common to alternate between these two schemes within the same improvement heuristic.

Intraroute heuristics have been well documented in the TSP literature (Laporte 2009). Interroute methods typically consist of removing one or several customers from a number of routes and relocating them. A large number of algorithms have been proposed within this broad framework, most of which are special cases of the b-cyclic, k-transfer scheme of Thompson and Psaraftis (1993), where a circular permutation of b routes is selected and k customers from each route are moved to the next route of the permutation. The combinations $b = 2$ or b variable and $k = 1$ or 2 seem to produce good results. Kindervater and Savelsbergh (1997) have described efficient ways of handling edge exchanges, mostly within the context of the VRP with time windows.

4. Metaheuristics

Metaheuristics can be broadly classified into local search, population search, and learning mechanisms. Most VRP metaheuristics are of the first kind but there are interesting examples in the other two categories.

Most metaheuristics can be regarded as improvement methods. The best ones are rather robust and perform very well even if they are initiated from a low-quality solution. A general tendency has been to move from algorithms based on a single paradigm to hybrid methods that draw on several principles. Thus, memetic algorithms (Moscato and Cotta 2003) combine population search and local search, resulting in a search trajectory that is at the same time broad and deep.

The number of variants of VRP metaheuristics published in recent years and their level of intricacy make it rather difficult to provide a comprehensive account of the relevant literature. Instead, we will explain the basic principles underlying each class of algorithms while pointing out the distinguishing features of the most important ones.

4.1. Local Search

Essentially, local search explores the solution space by moving at each iteration from the current solution to another solution in its neighbourhood. Classical examples include tabu search (Glover 1986), simulated annealing (Kirkpatrick, Gelatt, and Vecchi 1983), deterministic annealing (Dueck and Scheurer 1990; Dueck 1993), variable neighbourhood search (Mladenović and Hansen 1997), very large neighbourhood search (Ergun, Orlin, and Steele-Feldman 2006) and adaptive large neighbourhood search (Ropke and Pisinger 2006). The main ingredients of local search are the rules employed to define the neighbourhood of a solution and the mechanism put in place to explore it. Typically, local search heuristics perform inter-route moves, as described in §3.4, with occasional intra-route reoptimizations.

In tabu search the solution space is explored by moving from the current solution to the best solution in a subset of its neighbourhood. To avoid cycling, solutions possessing a given attribute of the current solution are not considered for a number of iterations (they are declared *tabu*). An exception is when such a solution constitutes a new best solution among all known solutions possessing the current attribute. This principle was first formalized by Cordeau, Gendreau, and Laporte (1997) and is now known as attribute based search (Derigs and Kaiser 2007). In simulated annealing, a solution x is randomly drawn from the neighbourhood $N(x_t)$ of the current solution x_t at iteration t. If the objective solution f is to be minimized, then $x_{t+1} := x$ whenever $f(x_{t+1}) \leq f(x_t)$. Otherwise, the search proceeds to $x_{t+1} := x$ with probability p_t and to $x_{t+1} := x_t$ with probability $1 - p_t$, where p_t is a decreasing function of t and of $f(x) - f(x_t)$. In deterministic annealing, x is also randomly selected in $N(x_t)$ and two different rules can be applied. In a threshold-accepting algorithm (Dueck and Scheurer 1990), $x_{t+1} := x$ whenever $f(x) < f(x_t) + \theta_1$, where θ_1 is a positive tolerance; otherwise $x_{t+1} := x_t$. In record-to-record travel (Dueck 1993), a record is the best known solution x^*. Then $x_{t+1} := x$ whenever $f(x) \leq \theta_2 f(x^*)$, where θ_2 is a positive tolerance; otherwise $x_{t+1} := x_t$.

Variable neighbourhood search (Mladenović and Hansen 1997) works with an ordered list of neighbourhoods, which are usually nested. The algorithm starts with a given neighbourhood and switches to the next neighbourhood in the list when it reaches a local minimum. The search is reinitiated from the first neighbourhood whenever a new best solution is identified or when all neighbourhoods have been explored. An application of variable neighbourhood search to the VRP is described in Kytöjoki et al. (2007). Very large-scale neighbourhood search simultaneously destroys and reconstructs several parts of the solution, usually by solving a network flow problem to identify the best neighbour. In a sense, the principle underlying this method is related to the destroy and repair mechanism of Shaw (1998). This metaheuristic has been applied to the VRP by Ergun, Orlin, and Steele-Feldman (2006). Finally, in adaptive large neighbourhood search (Ropke and Pisinger 2006, Pisinger and Ropke 2007) several insertion and removal heuristics are defined. At each iteration one of these is randomly selected by giving it a weight proportional to its success rate in previous iterations.

The local search mechanisms just outlined, and others that could eventually be proposed, can be enhanced through various mechanisms. Parallel computing, implemented in the tabu search heuristic of Taillard (1993), is an obvious example. The granularity principle (Toth and Vigo 2003) is also of wide applicability. It consists of removing long edges from the data (those that are unlikely to be part of an optimal solution) to ease the computational burden. This idea was successfully applied by Li, Golden, and Wasil (2005) in conjunction with record-to-record search. It is common to apply a diversification mechanism within a local search. A popular rule is to penalize frequently performed moves in the objective function (Taillard 1993; Gendreau, Hertz, and Laporte 1994; Cordeau, Gendreau, and Laporte 1997; Cordeau, Laporte, and Mercier 2001). Several local search schemes also allow the exploration of intermediate infeasible solutions. This is done through the inclusion in the objective function of penalty terms weighted by self-adjusting positive parameters (Gendreau, Hertz, and Laporte 1994; Cordeau, Gendreau, and Laporte 1997; Cordeau, Laporte, and Mercier 2001). Finally, limitation strategies can be applied to restrict the number of moves considered within a given neighbourhood structure (Nagata and Bräysy 2008). For example, when applying a three-opt algorithm, several potential edge exchanges are unpromising and can thus be disregarded.

4.2. Population Search

Population search works with a population of solutions. Genetic algorithms (Holland 1975) are the best-known example of this paradigm. At each iteration of a genetic algorithm, some parent solutions are extracted from the current population and recombined to create offspring which then replace the worst elements of the population if this yields improvements. It is standard to apply a diversification mechanism, called mutation, to the offspring before considering their inclusion in the population.

As far as we are aware, all known genetic algorithms applied to the VRP have been applied in conjunction with local search. This is usually achieved by improving the offspring through local search. Some fine examples are provided in Prins (2004, 2009), Mester and Bräysy (2005, 2007), Nagata (2007), and Nagata and Bräysy (2009).

Yet a different way to combine genetic search and local search is to first apply local search and record a set of best-known solutions in a memory. After the main phase of the search has ended, the solutions lying in the memory are recombined to create new partial solutions, local search is reinitiated from these partial solutions, and the memory is updated. Rochat and Taillard (1995) who have introduced this procedure, report excellent results with it. Other applications are described in Tarantilis and Kiranoudis (2002) and in Tarantilis (2005).

4.3. Learning Mechanisms

Learning mechanisms include neural networks which are derived from concepts borrowed from artificial intelligence. They can learn from experience and incrementally adjust their weights in an iterative fashion. The limited application of this concept to the VRP (Ghaziri 1991; Schumann and Retzko 1995) has met with mixed success. Ant colony optimization is another form of learning mechanism. It mimics the behaviour of ants foraging for food and laying pheromone on their paths. With time, pheromone accumulates faster on the shortest paths which are then followed by more ants. In an ant colony optimization algorithm, this idea translates into gradually giving more weight to the edges appearing frequently in good solutions. An interesting and rather successful application of this type of algorithm is provided by Reimann, Doerner, and Hartl (2004).

4.4. Computational Results

On the CMT instances (Christofides, Mingozzi, and Toth 1979), the most accurate metaheuristics appear to be those of Mester and Bräysy (2005, 2007), Nagata (2007), Nagata and Bräysy (2008, 2009), Taillard (1993), and Rochat and Taillard (1995). They all yield average deviations from the best-known solution in the 0.00% to 0.05% range. On the GWKC instances, these algorithms are also among the best, but the top-ten list also includes Prins (2009), Pisinger and Ropke (2007), Reimann, Doerner, and Hartl (2004), and Tarantilis (2005). Here the average deviations range from 0.01% to 0.76%. Computation times are hard to compare because of the variety of computers used for the experiments and because some heuristics are based on parallel computing. However, computing times are often very modest and much smaller than those of the early metaheuristics put forward 20 years ago. Many of the recent metaheuristics can solve instances involving more than 100 vertices within 0.1% of the best-known solution value in less than a minute.

5. Conclusions

Much progress has been accomplished in the field of vehicle routing in the past 50 years. The VRP is more popular than ever. It has attracted the attention of the operations research community; it has been an engine for the phenomenal growth we have witnessed in the fields of exact algorithms and heuristics; and some of these algorithms have found their way into commercial solvers used by industry.

In the area of exact algorithms, it is apparent that the VRP is considerably more difficult to solve than the TSP. We have just about broken the 100 barrier and it is difficult at this stage to predict how much further we can go.

The VRP has also been a major source of motivation for the development of heuristics. For all practical purposes the classical VRP can be solved adequately for realistic instance sizes. There is, however, a sense that several of the most successful metaheuristics are over-engineered and one should now attempt to produce simple and more flexible algorithms capable of handling a larger variety of constraints, even if this were to translate into a small loss in accuracy. On this topic, see Cordeau et al. (2002). In addition, time has probably come to develop algorithms better able to incorporate dynamic and stochastic features that are so common in practice.

Acknowledgments
This work was partly supported by the Canadian Natural Sciences and Engineering Research Council under Grant 05-39682; that support is gratefully acknowledged. Thanks are due to a referee who provided valuable comments on an earlier version of this paper.

References

Agarwal, Y., K. Mathur, H. M. Salkin. 1989. A set-partitioning-based exact algorithm for the vehicle routing problem. *Networks* **19** 731–749.

Altinkemer, K., B. Gavish. 1991. Parallel savings-based heuristic for the delivery problem. *Oper. Res.* **39** 456–469.

Applegate, D. L., R. E. Bixby, V. Chvátal, W. J. Cook. 2007. *The Traveling Salesman Problem. A Computational Study*. Princeton University Press, Princeton, NJ.

Baker, B. M., J. Sheasby. 1999. Extensions to the generalised assignment heuristic for vehicle routing. *Eur. J. Oper. Res.* **119** 147–157.

Balas, E., M. W. Padberg. 1976. Set partitioning: A survey. *SIAM Rev.* **18** 710–760.

Baldacci, R., N. Christofides, A. Mingozzi. 2008. An exact algorithm for the vehicle routing problem based on the set partitioning formulation with additional cuts. *Math. Programming Ser. A* **115** 351–385.

Baldacci, R., E. Hadjiconstantinou, A. Mingozzi. 2004. An exact algorithm for the capacitated vehicle routing problem based on a two-commodity network flow formulation. *Oper. Res.* **52** 723–738.

Balinski, M., R. Quandt. 1964. On an integer program for a delivery problem. *Oper. Res.* **12** 300–304.

Carpaneto, G., P. Toth. 1980. Some new branching and bounding criteria for the asymmetric travelling salesman problem. *Management Sci.* **26** 736–743.

Christofides, N. 1976. The vehicle routing problem. *RAIRO (Recherche Opérationnelle)* **10** 55–70.

Christofides, N., S. Eilon. 1969. An algorithm for the vehicle dispatching problem. *Oper. Res. Quart.* **20** 309–318.

Christofides, N., A. Mingozzi, P. Toth. 1979. The vehicle routing problem. N. Christofides, A. Mingozzi, P. Toth, C. Sandi, eds. *Combined Optimization*. Wiley, Chichester, UK, 315–338.

Christofides, N., A. Mingozzi, P. Toth. 1981a. Exact algorithms for the vehicle routing problem, based on spanning tree shortest path relaxations. *Math. Programming* **20** 255–282.

Christofides, N., A. Mingozzi, P. Toth. 1981b. Space state relaxation procedures for the computation of bounds to routing problems. *Networks* **11** 145–164.

Clarke, G., J. W. Wright. 1964. Scheduling of vehicles from a central depot to a number of delivery points. *Oper. Res.* **12** 568–581.

Cordeau, J.-F., M. Gendreau, G. Laporte. 1997. A tabu search heuristic for periodic and multi-depot vehicle routing problems. *Networks* **30** 105–119.

Cordeau, J.-F., G. Laporte, A. Mercier. 2001. A unified tabu search heuristic for vehicle routing problems with time windows. *J. Oper. Res. Soc.* **52** 928–936.

Cordeau, J.-F., G. Laporte, M. W. P. Savelsbergh, D. Vigo. 2007. Vehicle routing. C. Barnhart, G. Laporte, eds. *Transportation, Handbooks in Operations Research and Management Science*, Vol. 14. Elsevier, Amsterdam, 367–428.

Cordeau, J.-F., M. Gendreau, A. Hertz, G. Laporte, J.-S. Sormany. 2005. New heuristics for the vehicle routing problem. A. Langevin, D. Riopel, eds. *Logistics Systems and Optimization*. Springer, New York, 279–297.

Cordeau, J.-F., M. Gendreau, G. Laporte, J.-Y. Potvin, F. Semet. 2002. A guide to vehicle routing heuristics. *J. Oper. Res. Soc.* **53** 512–522.

Dantzig, G. B., J. H. Ramser. 1959. The truck dispatching problem. *Management Sci.* **6** 80–91.

Dantzig, G. B., D. R. Fulkerson, S. M. Johnson. 1954. Solution of a large-scale travelling-salesman problem. *Oper. Res.* **2** 393–410.

Derigs, U., R. Kaiser. 2007. Applying the attribute based hill climber heuristic to the vehicle routing problem. *Eur. J. Oper. Res.* **177** 719–732.

Dueck, G. 1993. New optimization heuristics: The great deluge algorithm and the record-to-record travel. *J. Computational Phys.* **104** 86–92.

Dueck, G., T. Scheurer. 1990. Threshold accepting: A general purpose optimization algorithm. *J. Computational Phys.* **90** 161–175.

Eilon, S., C. D. T. Watson-Grandy, N. Christofides. 1971. *Distribution Management: Mathematical Modelling and Practical Analysis*. Griffin, London.

Ergun, O., J. B. Orlin, A. Steele-Feldman. 2006. Creating very large-scale neighbourhoods out of smaller ones by compounding moves. *J. Heuristics* **12** 115–140.

Finke, G. A., A. Claus, E. Gunn. 1984. A two-commodity network flow approach to the traveling salesman problem. *Congressus Numerantium* **41** 167–178.

Fisher, M. L. 1994. Optimal solution of vehicle routing problems using minimum K-trees. *Oper. Res.* **42** 626–642.

Fisher, M. L. 1995. Vehicle routing. M. O. Ball, T. L. Magnanti, C. L. Monma, G. L. Nemhauser, eds. *Network Routing. Handbooks in Operations Research and Management Science*, Vol. 8. North-Holland, Amsterdam, 1–33.

Fisher, M., R. Jaikumar. 1981. A generalized assignment heuristic for vehicle routing. *Networks* **11** 109–124.

Foster, B. A., D. M. Ryan. 1976. An integer programming approach to the vehicle scheduling problem. *Oper. Res.* **27** 367–384.

Fukasawa, R., H. Longo, J. Lysgaard, M. Poggi de Aragão, M. Reis, E. Uchoa, R. F. Werneck. 2006. Robust branch-and-cut-and-price for the capacitated vehicle routing problem. *Math. Programming Ser. A* **106** 491–511.

Gavish, B., S. C. Graves. 1979. The travelling salesman problem and related problems. Working Paper 7905, Graduate School of Management, University of Rochester, Rochester, NY.

Gendreau, M., A. Hertz, G. Laporte. 1994. A tabu search heuristic for the vehicle routing problem. *Management Sci.* **41** 1276–1290.

Gendreau, M., G. Laporte, J.-Y. Potvin. 2002. Metaheuristics for the capacitated VRP. P. Toth, D. Vigo, eds. *The Vehicle Routing Problem. SIAM Monographs on Discrete Mathematics and Applications*. Society for Industrial and Applied Mathematics, Philadelphia, 129–154.

Ghaziri, H. 1991. Solving routing problems by a self-organizing map. T. Kohonen, K. Makisara, O. Simula, J. Kangas, eds. *Artificial Neural Networks*. North-Holland, Amsterdam, 829–834.

Gillett, B. E., L. R. Miller. 1974. A heuristic algorithm for the vehicle dispatch problem. *Oper. Res.* **22** 340–349.

Glover, F. 1986. Future paths for integer programming and links to artificial intelligence. *Comput. Oper. Res.* **13** 533–549.

Golden, B., T. L. Magnanti, H. Q. Nguyen. 1977. Implementing vehicle routing algorithms. *Networks* **7** 113–148.

Golden, B. L., S. Raghavan, E. A. Wasil, eds. 2008. *The Vehicle Routing Problem: Latest Advances and New Challenges*. Springer, New York.

Golden, B. L., E. A. Wasil, J. P. Kelly, I.-M. Chao. 1998. Metaheuristics in vehicle routing. T. G. Crainic, G. Laporte, eds. *Fleet Management and Logistics*. Kluwer, Boston, 33–56.

Hadjiconstantinou, E., N. Christofides, A. Mingozzi. 1995. A new exact algorithm for the vehicle routing problem based on q-paths and k-shortest paths relaxations. *Ann. Oper. Res.* **61** 21–43.

Hoffman, K., M. W. Padberg. 1993. Solving airline crew scheduling problems by branch-and-cut. *Management Sci.* **39** 657–682.

Holland, J. H. 1975. *Adaptation in Natural and Artificial Systems*. The University of Michigan Press, Ann Arbor, MI.

Houck, D., J.-C. Picard, M. Queyranne, R. Vemuganti. 1980. The travelling salesman problem as a constrained shortest path problem: Theory and computation experience. *Opsearch* **17** 93–109.

Kindervater, G. A. P., M. W. P. Savelsbergh. 1997. Vehicle routing: Handling edge exchanges. E. H. L. Aarts, J. K. Lenstra, eds. *Local Search in Combinatorial Optimization*. Wiley, Chichester, UK, 337–360.

Kirkpatrick, S., C. D. Gelatt, Jr., M. P. Vecchi. 1983. Optimization by simulated annealing. *Science* **220** 671–680.

Kytöjoki, J., T. Nuortio, O. Bräysy, M. Gendreau. 2007. An efficient variable neighborhood search heuristic for very large scale vehicle routing problems. *Comput. Oper. Res.* **34** 2743–2757.

Laporte, G. 2007. What you should know about the vehicle routing problem. *Naval Res. Logist.* **54** 811–819.

Laporte, G. 2009. A concise guide to the traveling salesman problem. *J. Oper. Res. Soc.* Forthcoming.

Laporte, G., Y. Nobert. 1983. A branch and bound algorithm for the capacitated vehicle routing problem. *Oper. Res. Spektrum* **5** 77–85.

Laporte, G., Y. Nobert. 1987. Exact algorithms for the vehicle routing problem. *Ann. Discrete Math.* **31** 147–184.

Laporte, G., F. Semet. 2002. Classical heuristics for the capacitated VRP. P. Toth, D. Vigo, eds. *The Vehicle Routing Problem. SIAM Monographs on Discrete Mathematics and Applications*, Society for Industrial and Applied Mathematics, Philadelphia, 109–128.

Laporte, G., H. Mercure, Y. Nobert. 1986. An exact algorithm for the asymmetrical capacited vehicle routing problem. *Networks* **16** 33–46.

Laporte, G., Y. Nobert, M. Desrochers. 1985. Optimal routing under capacity and distance restrictions. *Oper. Res.* **33** 1050–1073.

Letchford, A. N., R. W. Eglese, J. Lysgaard. 2002. Multistars, partial multistars and the capacitated vehicle routing problem. *Math. Programming Ser. A* **94** 21–40.

Li, F., B. L. Golden, E. A. Wasil. 2005. Very large-scale vehicle routing: New test problems, algorithms, and results. *Comput. Oper. Res.* **32** 1165–1179.

Little, J. D. C., K. G. Murty, D. W. Sweeney, C. Karel. 1963. An algorithm for the travelling salesman problem. *Oper. Res.* **11** 972–989.

Lysgaard, J., A. N. Letchford, R. W. Eglese. 2004. A new branch-and-cut algorithm for the capacitated vehicle routing problem. *Math. Programming* **100** 423–445.

Mester, D., O. Bräysy. 2005. Active guided evolution strategies for the large scale vehicle routing problems with time windows. *Comput. Oper. Res.* **32** 1593–1614.

Mester, D., O. Bräysy. 2007. Active guided evolution strategies for the large scale capacitated vehicle routing problems. *Comput. Oper. Res.* **34** 2964–2975.

Mladenović, N., P. Hansen. 1997. Variable neighborhood search. *Comput. Oper. Res.* **24** 1097–1100.

Moscato, P., C. Cotta. 2003. A gentle introduction to memetic algorithms. F. Glover, G. A. Kochenberger, eds. *Handbook of Metaheuristics*. Kluwer, Boston, 105–144.

Naddef, D., G. Rinaldi. 2002. Branch-and-cut algorithms for the capacitated VRP. P. Toth, D. Vigo, eds. *The Vehicle Routing Problem. SIAM Monographs on Discrete Mathematics and Applications*. Society for Industrial and Applied Mathematics, Philadelphia, 53–81.

Nagata, Y. 2007. Edge assembly crossover for the capacitated vehicle routing problem. C. Cotta, J. van Hemert, eds. *Lecture Notes in Computer Science*, Vol. 4446. Springer-Verlag, Berlin Heidelberg, 142–153.

Nagata, Y., O. Bräysy. 2008. Efficient local search limitation strategies for the vehicle routing problem. Working paper, Agora Innoroad Laboratory, University of Juväskylä, Juväskylä, Finland.

Nagata, Y., O. Bräysy. 2009. Edge assembly based memetic algorithm for the capacitated vehicle routing problem. Working paper, Agora Innoroad Laboratory, University of Juväskylä, Juväskylä, Finland.

Nelson, M. D., K. E. Nygard, J. H. Griffin, W. E. Shreve. 1985. Implementation techniques for the vehicle routing problem. *Comput. Oper. Res.* **12** 273–283.

Paessens, H. 1988. The savings algorithm for the vehicle routing problem. *Eur. J. Oper. Res.* **34** 336–344.

Pisinger, D., S. Ropke. 2007. A general heuristic for vehicle routing problems. *Comput. Oper. Res.* **34** 2403–2435.

Prins, C. 2004. A simple and effective evolutionary algorithm for the vehicle routing problem. *Comput. Oper. Res.* **31** 1985–2002.

Prins, C. 2009. A GRASP X evolutionary local search hybrid for the vehicle routing problem. F. B. Pereira, J. Tavares, eds. *BIO-Inspired Algorithms for the Vehicle Routing Problem*. Springer, Berlin-Heidelberg, 35–53.

Rao, M. R., S. Zionts. 1968. Allocation of transportation units to alternative trips—A column generation scheme with out-of-kilter subproblems. *Oper. Res.* **16** 52–63.

Reimann, M., K. Doerner, R. F. Hartl. 2004. D-Ants: Savings based ants divide and conquer the vehicle routing problem. *Comput. Oper. Res.* **31** 563–591.

Renaud, J., F. F. Boctor, G. Laporte. 1996. An improved petal heuristic for the vehicle routing problem. *J. Oper. Res. Soc.* **47** 329–336.

Rochat, Y., É. D. Taillard. 1995. Probabilistic diversification and intensification in local search for vehicle routing. *J. Heuristics* **1** 147–167.

Ropke, S., D. Pisinger. 2006. A unified heuristic for a large class of vehicle routing problems with backhauls. *Eur. J. Oper. Res.* **171** 750–775.

Ryan, D. M., C. Hjorring, F. Glover. 1993. Extensions of the petal method for vehicle routing. *J. Oper. Res. Soc.* **44** 289–296.

Schumann, M., R. Retzko. 1995. Self-organizing maps for vehicle routing problems—Minimizing an explicit cost function. F. Fogelman-Soulie, ed. *Proc. Internat. Conf. Artificial Neural Networks, Paris*, 401–406.

Shaw, P. 1998. Using constraint programming and local search methods to solve vehicle routing problems. *CP98 Fourth Internat. Conf. Principles Practice Constraint Programming*, Vol. 1520, *Lecture Notes in Computer Science*. Springer, Berlin, 417–431.

Taillard, É. D. 1993. Parallel iterative search methods for vehicle routing problems. *Networks* **23** 661–673.

Tarantilis, C.-D. 2005. Solving the vehicle routing problem with adaptive memory programming methodology. *Comput. Oper. Res.* **32** 2309–2327.

Tarantilis, C.-D., C. T. Kiranoudis. 2002. BoneRoute: An adaptive memory-based method for effective fleet management. *Ann. Oper. Res.* **115** 227–241.

Thompson, P. M., H. M. Psaraftis. 1993. Cyclic transfer algorithms for multi-vehicle routing and scheduling problems. *Oper. Res.* **41** 935–946.

Toth, P., D. Vigo, eds. 2002a. *The Vehicle Routing Problem. SIAM Monographs on Discrete Mathematics and Applications*. Society for Industrial and Applied Mathematics, Philadelphia.

Toth, P., D. Vigo. 2002b. Branch-and-bound algorithms for the capacitated VRP. P. Toth, D. Vigo, eds. *The Vehicle Routing Problem. SIAM Monographs on Discrete Mathematics and Applications*. Society for Industrial and Applied Mathematics, Philadelphia, 29–51.

Toth, P., D. Vigo. 2003. The granular tabu search and its application to the vehicle routing problem. *INFORMS J. Comput.* **15** 333–346.

Wark, P., J. Holt. 1994. A repeated matching heuristic for the vehicle routing problem. *J. Oper. Res. Soc.* **45** 1156–1167.