What You Should Know about the Vehicle Routing Problem

Gilbert Laporte

Canada Research Chair in Distribution Management, HEC Montréal 3000, chemin de la Côte-Sainte-Catherine, Montréal, Canada H3T 2A7

Received 8 August 2007; accepted 8 August 2007
DOI 10.1002/nav.20261
Published online 19 September 2007 in Wiley InterScience (www.interscience.wiley.com).

Abstract: In the Vehicle Routing Problem (VRP), the aim is to design a set of m minimum cost vehicle routes through n customer locations, so that each route starts and ends at a common location and some side constraints are satisfied. Common applications arise in newspaper and food delivery, and in milk collection. This article summarizes the main known results for the *classical* VRP in which only vehicle capacity constraints are present. The article is structured around three main headings: exact algorithms, classical heuristics, and metaheuristics. © 2007 Wiley Periodicals, Inc. Naval Research Logistics 54: 811–819, 2007

Keywords: vehicle routing problem; capacity constraints; integer linear programming; heuristics; metaheuristics

1. INTRODUCTION

The Vehicle Routing Problem (VRP) consists of designing optimal delivery or collection routes from a central *depot* to a set of geographically scattered *customers*, subject to various constraints, such as vehicle capacity, route length, time windows, precedence relations between customers, etc. This problem is faced on a daily basis by thousands of distributors worldwide and has significant economic importance. Common examples are the delivery of newspapers to retailers, of food and beverages to grocery stores, and the collection of milk products from dairy farmers (Golden et al. [29]). The problem was introduced nearly 50 years ago by Dantzig and Ramser [15] and has since given rise to a rich body of research.

Unlike what happens for several well-known combinatorial optimization problems, there does not exist a single universally accepted definition of the VRP because of the diversity of constraints encountered in practice. Most of the research effort has concentrated on a standardized version of the problem, called the *classical* VRP, with the understanding that many of the algorithms developed for this case, mostly heuristics, can be adapted to suit the more complicated real-life situations. In fact, most heuristics presented in this paper can easily handle an upper limit on route lengths.

The classical VRP is defined on an undirected graph $G = (V, A)$ where $V = \{0, 1, \ldots, n\}$ is the vertex set and $A = \{(i, j) : i, j \in V, i \neq j\}$ is the arc set. Vertex 0 represents a depot at which are located at most m identical vehicles of capacity Q. With each customer $i \in V\backslash\{0\}$ is associated a non-negative demand $q_i \leq Q$. A cost matrix c_{ij} is defined on A. When the cost matrix is symmetric, i.e., $c_{ij} = c_{ji}$ for all i, j, it is common to define the problem on an undirected graph $G = (V, E)$, where $E = \{[i, j] : i, j \in V, i < j\}$ is the edge set. We use the terms travel cost, length and travel time interchangeably. The problem consists of determining a set of m vehicle routes (1) starting and ending at the depot, and such that (2) each customer is visited by exactly one vehicle, (3) the total demand of any route does not exceed Q, and (4) the total routing cost is minimized. Figure 1 depicts a classical VRP solution. Note that this solution contains two intersecting routes, as well as a back and forth route between the depot and a single customer.

The VRP is NP-hard because it includes the Traveling Salesman Problem (TSP) as a special case when $m = 1$ and $Q = \infty$. In practice, the VRP is considerably more difficult to solve than a TSP of the same size. For example, TSPs involving hundreds and even thousands of vertices can be solved routinely by means of advanced branch-and-cut-and-price algorithms (Applegate et al. [2]). In contrast, the most sophisticated exact algorithms for the VRP (see, e.g., Baldacci et al. [5]) can only solve instances of up to about 100 customers, and with a variable success rate. This explains to a large extent why most of the research effort has concentrated on heuristics. Another reason is the fact that heuristics tend to be considerably more flexible than exact algorithms and can

Correspondence to: G. Laporte (gilbert@crt.umontreal.ca)

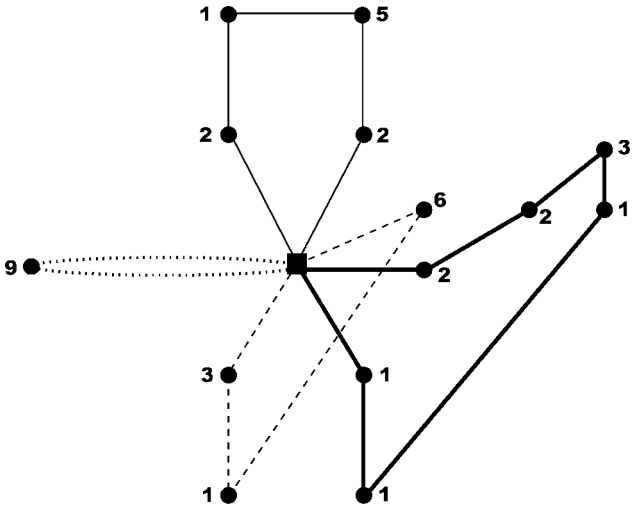

Figure 1. Solution of a classical VRP with 14 customers and four vehicles of capacity $Q = 10$. Customer demands are shown next to the vertices.

be more readily adapted to deal with the diversity of variants arising in practice.

In keeping with the spirit of these cover articles, our aim is not to provide a full survey of the VRP, but rather to summarize the most important concepts, algorithms and results, those that have withstood the test of time and that all researchers should know. For general surveys of the VRP, see Cordeau et al. [12] and Toth and Vigo [59].

The remainder of this article is organized as follows. Section 2 contains a description of the main exact VRP algorithms. Heuristics are subdivided into classical heuristics, presented in Section 3, and metaheuristics, presented in Section 4. Conclusions follow in Section 5.

2. EXACT ALGORITHMS

Several families of exact algorithms have been proposed for the VRP with a symmetric cost structure. These are based on integer linear programming (ILP), dynamic programming, and branch-and-bound. For reviews, see Laporte and Nobert [36] and Toth and Vigo [58]. Here we concentrate on three families of ILP based branch-and-cut algorithms which have proved to be the only workable methodology. Unfortunately they all require a rather heavy mathematical programming machinery and their success in solving realistic size instances is limited.

2.1. Two-Index Vehicle Flow Formulations

Two-index vehicle flow formulations for the VRP are rooted in the work of Laporte et al. [37] and are extensions of the classical TSP formulation of Dantzig et al. [14]. Let x_{ij} be an integer variable representing the number of times edge $[i, j]$ appears in the optimal solution. If $i, j \in V\backslash\{0\}$, then x_{ij} is binary; if $i = 0$, then x_{ij} can be equal to 0, 1, or 2, the latter case corresponding to a return trip between the depot and customer j. The problem is then:

$$\text{(VF) Minimize} \quad \sum_{[i,j]\in E} c_{ij}x_{ij} \tag{1}$$

$$\text{subject to} \quad \sum_{j\in V\backslash\{0\}} x_{0j} = 2m \tag{2}$$

$$\sum_{i<k} x_{ik} + \sum_{j>k} x_{kj} = 2 \quad (k \in V\backslash\{0\}) \tag{3}$$

$$\sum_{\substack{i\in S, j\notin S \\ \text{or } i\notin S, j\in S}} x_{ij} \geq 2b(S) \quad (S \subset V\backslash\{0\}) \tag{4}$$

$$x_{i,j} = 0 \text{ or } 1 \quad (i, j \in V\backslash\{0\}) \tag{5}$$

$$x_{0j} = 0, 1 \text{ or } 2 \quad (j \in V\backslash\{0\}). \tag{6}$$

In this formulation, the objective function minimizes the total routing cost. Constraints (2) define the degree of vertex 0. Note that the right-hand side can be a constant if m is known a priori, or a variable otherwise. In the latter case, a term fm can be added to the objective function, where f is the vehicle fixed cost. Constraints (3) ensure that two edges are incident to each customer vertex. In constraints (4), $b(S)$ is a lower bound on the number of vehicles required to serve all customers of S. These constraints play a dual role: they prevent the formation of subtours by forcing any subset of customers to be connected to the depot, and they ensure that capacity constraints are not violated. In practice, it is common to define $b(S)$ as $\lceil \sum_{i\in S} q_i / Q \rceil$.

Because the number of connectivity constraints is exponential in n, it is common to solve VF by branch-and-cut. Initially the connectivity and integrality constraints are relaxed. Constraints (4) are dynamically generated during the solution process as they are found to be violated, whereas integrality is reached by branching on fractional variables. Several families of valid inequalities can also be used to strengthen the linear relaxation of the problem. These include generalized capacity constraints, frame capacity constraints, and any inequality valid for the TSP, such as comb inequalities, some inequalities combining bin packing and the TSP, as well as inequalities based on the stable set problem. These families of inequalities, and related separation procedures, are described by Naddef and Rinaldi [44].

The best algorithm based on VF is due to Naddef and Rinaldi [44]. Using a branch-and-cut algorithm, these authors have solved at the root node six instances with $22 \leq n \leq 45$,

and nine other instances with $51 \leq n \leq 135$ by using some branching.

2.2. Two-Index Two-Commodity Flow Formulations

Baldacci et al. [6] have proposed a two-index two-commodity flow formulation for the VRP, based on an earlier similar formulation by Finke et al. [20] for the TSP. The formulation makes use of travel directions on edges and works on an extended graph $\overline{G} = (\overline{V}, \overline{E})$ where the vertex set $\overline{V} = V \cup \{n+1\}$ includes a copy $n+1$ of the depot, and $\overline{E} - E \cup \{[i, n+1] : i \in V\}$. With this graph representation, a vehicle route is a directed path from 0 to $n+1$. Binary variables x_{ij} are equal to 1 if and only if edge $[i, j]$ appears in the optimal solution. Binary variables y_{ij} and y_{ji} represent, respectively, the vehicle load on edge $[i, j]$ and the empty space on the vehicle traveling on edge $[i, j]$, i.e., $y_{ij} + y_{ji} = Q$ provided $x_{ij} = 1$. The formulation is:

$$\text{(CF) Minimize} \quad \sum_{[i,j]\in\overline{E}} c_{ij}x_{ij} \tag{7}$$

$$\text{subject to} \quad \sum_{j\in\overline{V}} (y_{ji} - y_{ij}) = 2q_i \quad (i \in V\backslash\{0\}) \tag{8}$$

$$\sum_{j\in V\backslash\{0\}} y_{0j} = \sum_{i\in V\backslash\{0\}} q_i \tag{9}$$

$$\sum_{j\in V\backslash\{0\}} y_{j0} = mQ - \sum_{i\in V\backslash\{0\}} q_i \tag{10}$$

$$\sum_{j\in V\backslash\{0\}} y_{n+1,j} = mQ \tag{11}$$

$$y_{ij} + y_{ji} = Qx_{ij} \quad ([i, j] \in \overline{E}) \tag{12}$$

$$\sum_{i<k} x_{ik} + \sum_{j>k} x_{kj} = 2 \quad (k \in V\backslash\{0\}) \tag{13}$$

$$y_{ij} \geq 0, \quad y_{ji} \geq 0 \quad ([i, j] \in \overline{E}) \tag{14}$$

$$x_{ij} = 0 \text{ or } 1 \quad ([i, j] \in \overline{E}) \tag{15}$$

In this formulation, constraints (8)–(11) and (14) define consistent flows from vertex 0 to vertex $n+1$. Constraints (12) ensure that the y_{ij} and y_{ji} variables take feasible values, while constraints (13) specify the degree of each customer vertex.

The authors have solved this formulation using a branch-and-cut algorithm in which valid VRP inequalities expressed in terms of the x_{ij} variables are gradually introduced. Several instances taken from the VRP literature, with $16 \leq n \leq 135$, could be solved to optimality. The algorithm can also solve, in a consistent fashion, randomly generated instances with $30 \leq n \leq 60$ and $m = 3$ or 5 and, less consistently, larger instances involving up to 100 customers and eight vehicles.

2.3. Set Partitioning Formulations

The VRP can also be formulated as a Set Partitioning Problem as follows (Balinski and Quandt [7]): Let R be the set of all feasible routes, let d_r be the cost of route $r \in R$, and let z_r be a binary variable equal to 1 if and only if route r belongs to the optimal solution. In addition, let a_{ir} be a binary coefficient equal to 1 if and only if customer i belongs to route r. The formulation is then:

$$\text{(SP) Minimize} \quad \sum_{r\in R} d_r z_r \tag{16}$$

$$\text{subject to} \quad \sum_{r\in R} a_{ir} z_r = 1 \quad (i \in V\backslash\{0\}) \tag{17}$$

$$\sum_{r\in R} z_r = m \tag{18}$$

$$z_r = 0 \text{ or } 1 \quad (r \in R). \tag{19}$$

As such, this formulation is impractical because of the large number of variables and of the difficulty of computing the d_r values (each requires solving a TSP over the vertices of r). Column generation, which is a natural methodology for this type of formulation, has proved mostly unsuccessful because the problem is not sufficiently constrained. However, the optimal solution value z(LSP) of the linear relaxation of SP provides a tight lower bound on the optimal VRP value. In addition, this formulation is rather flexible since it can easily accommodate a variety of side constraints such as time windows or maximal route lengths, hence reducing the cardinality of R.

As shown by Baldacci et al. [6], the vehicle flow formulation can be rewritten in terms of the z_r variables by using the following identity:

$$x_{ij} = \sum_{r\in R} \mu_{ij}^r z_r \quad ([i, j] \in E), \tag{20}$$

where if r is the route $(0, h, 0)$, then $\mu_{0h}^r = 2$ and $\mu_{ij}^r = 0$ for all $[i, j] \in E\backslash\{[0, h]\}$; if r contains at least two customers, then $\mu_{ij}^r = 1$ for each edge $[i, j]$ of route r, and $\mu_{ij}^r = 0$ otherwise.

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

$$\sum_{r\in R(S)} z_r \geq b(S) \quad (S \subset V\backslash\{0\}). \tag{21}$$

In addition, any valid inequality for the VRP (see Naddef and Rinaldi [44]; Letchford et al. [39]) of the form

$$\sum_{[i,j]\in E} \alpha_{ij} x_{ij} \geq \beta \tag{22}$$

can be reexpressed in terms of the z_r variables using (20). Finally, inequalities valid for the Set Partitioning Problem (Balas and Padberg [4]; Hoffman and Padberg [32]) can be incorporated into SP. Thus, Baldacci et al. [5] use the clique inequalities. Let H be a graph whose vertices correspond to vehicle routes. Two vertices are in conflict if the corresponding routes share at least one edge. For any clique C of H, the following inequality is valid:

$$\sum_{r \in C} z_r \leq 1. \tag{23}$$

Baldacci et al. [5] work with the augmented SP formulation defined by (16)–(19) and some constraints (21), (22), and (23). Since solving this problem remains intractable in all but trivial cases, they use the dual of its linear relaxation to compute lower bounds of the optimal primal value by means of three different heuristics. These procedures are embedded within a branch-and-cut algorithm to yield optimal VRP solutions. Using this approach the authors have solved to optimality several VRP instances with $37 \leq n \leq 121$. Their results are probably the best available.

In closing this section, we should mention the existence of another exact algorithm by Fukasawa et al. [23] combining a set partitioning formulation and cutting planes. This method seems to yield results almost as good as those of Baldacci, Christofides, and Mingozzi.

3. CLASSICAL HEURISTICS

Classical heuristics for the VRP are naturally divided into *constructive heuristics* and *improvement heuristics*. The qualificative "classical" refers to the fact that the improvement steps of these heuristics perform *descents*, i.e., they always proceed from a solution to a better one in its neighborhood until no further gain is possible. In contrast, metaheuristics (outlined in Section 4) allow the consideration of non-improving and even infeasible intermediate solutions. It is common to test heuristics on the Christofides et al. [8] instances, referred to as the CMT test-bed and consisting of 14 instances with $51 \leq n \leq 199$, and on the Golden et al. [31] instances, referred to as the GWKC test-bed and consisting of 20 larger instances with $200 \leq n \leq 480$. Since optimal solution values are unknown for most of these instances, comparisons are made with the best known values produced by metaheuristics.

3.1. Constructive Heuristics

The most popular construction heuristic is the Clarke and Wright [9] savings algorithm. Initially n back and forth routes $(0, i, 0)(i = 1, 2, \ldots, n)$ are constructed, all of which are feasible. A general iteration of the algorithm consists of merging a route ending at i with another route starting at j by removing arcs $(i, 0)$ and $(0, j)$, and adding arc (i, j), provided the saving $s_{ij} = c_{i0} + c_{0j} - c_{ij}$ is positive and the merged route is feasible. The best variant of the algorithm is the parallel version in which the merge yielding the largest saving is implemented at each iteration. The algorithm ends when no feasible and profitable merges are possible. Since the savings do not change during the algorithm, they can be computed and sorted within the initialization phase, the sorting step being the most time consuming component of the algorithm. As described in Laporte and Semet [38], several variants of the savings algorithm have been proposed, namely to speed up computations (Nelson et al. [46]), to optimize route merges based on savings values (Altinkemer and Gavish [1]; Wark and Holt [62]), and to make the savings definition more flexible (Golden et al. [30]). While this algorithm is not the best available in terms of accuracy (on benchmark instances it produces an average deviation of about 7% on the CMT test-bed), it is rather fast and simple to implement, which explains its popularity.

Another important class of constructive heuristics is made up of *petal heuristics* which consist of first generating a family R' of feasible routes and then solving the SP formulation over R' rather than the full set R. The success of the algorithm depends on the quality of the generated routes. The most elementary version of this type of heuristic is the sweep algorithm of Gillett and Miller [27]. Starting with a half-line rooted at the depot, this heuristic gradually constructs feasible routes by rotating another half-line. Customers are gradually incorporated into the current route in increasing order of the angle they make with the initial half-line. The route closes when the inclusion of a further customer becomes infeasible (see Fig. 2). This procedure only generates non-intersecting routes and is rather rudimentary. More sophisticated route generation procedures, proposed by Foster and Ryan [22], Ryan et al. [52], and Renaud et al. [50], have been described. In particular, the latter authors allow the creation of intersecting and embedded routes. Average deviations obtained with their heuristic on the CMT test-bed are 2.38%.

Fisher and Jaikumar [21] have proposed yet a different type of heuristic based on a two-phase decomposition procedure. In the first phase, a *seed* is located in the region where each route is likely to lie and clusters of customers are created through the solution of a Generalized Assignment Problem (GAP), that is, the sum of distances between customers and the seed to which they are allocated is minimized, subject to the constraints that the total demand of each cluster should not exceed Q. A TSP is then solved on each cluster. There are a number of problems with this approach. One is the determination of seeds which is not explicit in the original article, but some procedures are described in Baker and Sheasby [3]. Another difficulty lies in the solution of the GAP which is itself an NP-hard problem.

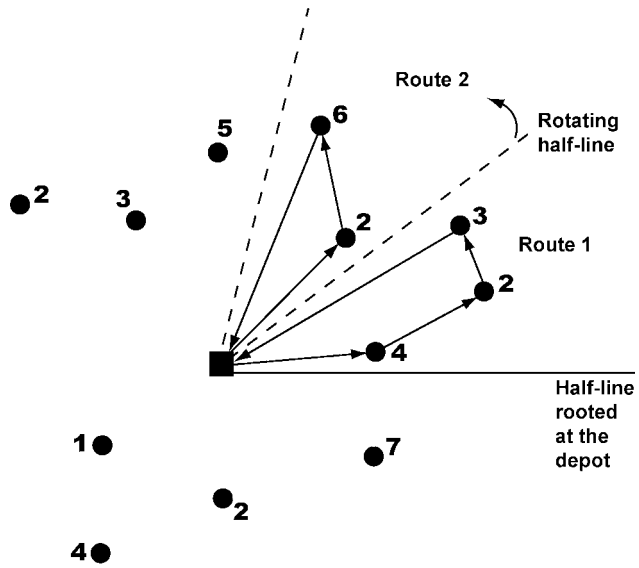

Figure 2. Construction of feasible routes in the sweep algorithm. The vehicle capacity is $Q = 10$. Customer demands are shown next to the vertices.

Given the strength of current improvement heuristics, particularly meta-heuristics, there is little point in fine tuning construction heuristics. A simple scheme, like the savings method, is adequate for most purposes.

3.2. Improvement Heuristics

Two types of improvement algorithms can be applied to VRP solutions. *Intra-route heuristics* postoptimize each route separately by means of a TSP improvement heuristic, e.g., 2-opt or 3-opt. *Inter-route heuristics* consist of moving vertices to different routes (Laporte and Semet [38]). The most common moves are simple transfers from one route to an other, transfers involving several routes, and vertex exchanges between two or more routes. The general frameworks described by Thompson and Psaraftis [57], Van Breedam [61] and Kindervater and Savelsbergh [34] encompass most available inter-route improvement procedures. In particular, Thompson and Psaraftis describe a general b-cyclic, k-transfer scheme in which a circular permutation of b routes is considered, and k vertices from each route are shifted to the next route of the cyclic permutation. The combinations $b = 2$ or b variable, and $k = 1$ or 2 yield interesting results. Van Breedam classifies improvement operations as string cross, string exchange, string relocation and string mix, which can be viewed as special cases of 2-cyclic exchanges. The author shows that string exchange moves are the best. Kindervater and Savelsbergh have described similar operations and have performed experiments mostly in the context of the VRP with time windows.

By and large, the performance of classical improvement heuristics is good but not excellent. They are best used as building blocks within metaheuristics.

4. METAHEURISTICS

Significant progress has been witnessed over the past 15 years in the development of metaheuristics for the VRP. All allow the exploration of the solution space beyond the first local minimum encountered, and all embed procedures borrowed from classical construction and improvement heuristics. A great variety of schemes have been put forward, but these can be broadly classified into three categories: (1) *local search*, (2) *population search*, and (3) *learning mechanisms*. For a survey and a bibliography on these topics, see Cordeau et al. [10] and Gendreau et al. [25].

4.1. Local Search

A local search heuristic starts from an initial solution s_0 (which may be infeasible) and moves at each iteration t from solution s_t of value $f(s_t)$ to another solution located in the neighbourhood $N(s_t)$ of s_t. In most cases s_t is the current solution but some multi-start mechanisms allow a reinitiation of the search from a solution that differs from the current one. The neighbourhood $N(s_t)$ consists of all solutions that can be reached from s_t by applying a given type of transformation, for example relocating a vertex from its current route into another route. The search ends with the best known solution s^* after a stopping criterion has been satisfied, usually a preset number of iterations, or a given number of consecutive iterations without improvement in s^*.

Within this broad framework, several algorithms can be defined. Here are the local search schemes that have proved the most successful for the VRP. In *record-to-record travel* (Dueck [17]), a record is s^*. A solution s is randomly selected in $N(s_t)$, and $s_{t+1} := s$ if $f(s) < \sigma f(s^*)$, where σ is a user controlled parameter generally slightly larger than 1. In *tabu search* (Glover [28]), s_{t+1} is the best solution in $N(s_t) \backslash T(s_t)$, where $T(s_t)$ is the set of tabu (forbidden) solutions at iteration t. Tabu solutions are necessary to prevent the search from cycling. In the VRP, the tabu mechanism can be implemented as follows (Cordeau et al. [12]): A set of attributes $B(s_t) = \{(i,k) : i \in V \backslash \{0\}, k = 1,2,\ldots,m\}$ is associated with solution s_t. A neighbour solution consists of removing a pair (i,k) from $B(s_t)$ and of inserting another pair (i,k') in its place. The pair (i,k) is then declared tabu for θ iterations. In *attribute based tabu search*, a solution s containing a tabu attribute (i,k) can be accepted if it corresponds to the best known solution containing that attribute. This rule is usually called an aspiration criterion. Other mechanisms are often used in tabu search, such as continuous

diversification, a rule aimed at penalizing solutions containing frequently encountered attributes, and intensification, a process aimed at performing a more thorough search around good solutions. It is also now common to allow intermediate infeasible solutions during the search through the use of a penalized objective function and self-adjusting penalties, a mechanism put forward by Gendreau et al. [24]. In *variable neighbourhood search* (Mladenović and Hansen [43]), several nested neighbourhoods are defined. The search is initiated with a given neighbourhood. When a local minimum is encountered, it proceeds to the next neighbourhood in the nested structure; the search restarts from the first neighbourhood whenever a new best solution s^* is identified or all neighbourhoods have been explored. In *very large scale neighbourhood search* (Ergun [18]), the size of $N(s_t)$ is very large and an optimization problem may have to be solved to determine the best neighbour of s_t. In *adaptive large neighbourhood search* (Pisinger and Ropke [47]), several insertion and removal heuristics are applied. Their selection is made randomly, by giving a higher weight to heuristics that have performed well in the past.

4.2. Population Search

Genetic algorithms (Holland [33]) evolve a population of solutions encoded as bitstrings, or chromosomes, through a crossover and mutation process. The crossover takes two parents from the population and combines them to generate one or two offspring solutions. A mutation (typically random) is applied to each offspring, and the offspring replace the worst elements of the population. In the context of the VRP, encoding solutions as bitstrings is not the most natural way to proceed and more natural mechanisms are used instead. For example, Prins [48] transforms the VRP solution into a TSP solution by removing the route delimiters and reconstructs the VRP solution at the end of the process. Also, random mutations are often replaced by the application of a standard improvement heuristic to the offspring, yielding what is commonly known as a *memetic algorithm*. A principle similar to genetic search, called an *adaptive memory procedure*, has been put forward by Rochat and Taillard [51] for the VRP. These authors first execute a tabu search algorithm and keep the best solutions in a memory. These solutions are then recombined through a crossover and local search process in the hope of generating even better solutions. For another implementation of this principle, see Tarantilis and Kiranoudis [56].

4.3. Learning Mechanisms

Learning mechanisms include *neural networks* and *ant colony optimization*. Early attempts to apply neural networks to the VRP have been rather unsuccessful (see, e.g., Ghaziri [26]; Schumann and Retzko [53]) and this line of research seems to have been abandoned. Ant colony optimization heuristics attempt to mimic the behaviour of ants who detect paths containing pheromone and strengthen them with their own pheromone. This leads to the emergence of shortest paths on which pheromone accumulates faster. In metaheuristics pheromone represents the memory of the system and corresponds to edges appearing often in good solutions. Thus the algorithm remembers good edges and is more likely to include them in a solution.

4.4. Summary and Assessment

All metaheuristics just outlined, as well as some hybrid implementations, have been applied to the solution of the VRP. In the area of local search Li et al. [40] have developed a simple yet efficient record-to-record heuristic. Several tabu search implementations have been highly successful, starting with the algorithm of Taillard [54] and its enhancement consisting of the use of an adaptive memory (Rochat and Taillard [51]). These two algorithms have yielded some of the best known solutions on standard test instances. Other successful tabu search implementations are the Unified Tabu Search Algorithm (UTSA) of Cordeau et al. [12] which is highly flexible and applies to a wide variety of routing problems, the general VRP heuristic of Pisinger and Ropke [47], and the Derigs and Kaiser [16] attribute based hill climber heuristic. Recently, Kytöjoki et al. [35] have developed a variable neighbourhood search heuristic capable of solving large scale instances.

Good examples of genetic search are the adaptive memory procedure, the algorithm of Prins [48], and the AGES algorithm of Mester and Bräysy [41, 42] combining genetic search, very large neighbourhoods and granularity. The granularity principle, put forward by Toth and Vigo [60], consists of removing a priori several unpromising arcs or edges from the graph. Another very good memetic algorithm is that of Nagata [45] who initially relaxes the capacity constraint and handles it through a penalty function when exploring neighbourhoods.

The D-ants savings algorithm of Reimann et al. [49] is probably the best example of ant colony optimization applied to the VRP. It generates a pool of good solutions by means of the Clarke and Wright [9] algorithm, where the savings s_{ij} are replaced by $t_{ij}^{\alpha} s_{ij}^{\beta}$, α and β are two user-controlled parameters, and t_{ij} measures how good combining i and j has been in previous solutions. The process is reiterated by adjusting the t_{ij}^{α} coefficients.

This short account cannot do justice to the wealth of metaheuristics put forward for the solution of the VRP. We have purposefully concentrated on the most significant and successful methods. Table 1 lists some of the best VRP metaheuristics, with an indication of their performance on the

Table 1. Comparison of several metaheuristics for the classical VRP.

References	Year	Type of algorithm	% CMT	% GWKC
Taillard [54]	1993	Tabu search	0.05	na
Gendreau et al. [24]	1994	Tabu search	0.86	na
Rochat and Taillard [51]	1995	Tabu search + adaptive memory	0.00	na
Cordeau et al. [12]	2001	Tabu search	0.56	1.57
Tarantilis and Kiranoudis [56]	2002	Tabu search + adaptive memory	0.23	0.76[a]
Toth and Vigo [60]	2003	Tabu search + granularity	0.64	2.99
Prins [48]	2004	Memetic algorithm	0.24	1.03
Reimann et al. [49]	2004	Ant colony optimization	na	0.71
Mester and Bräysy [41]	2005	Genetic algorithm + very large neighbourhoods + granularity		
		Best version	0.03	0.11
		Fast version	0.08	1.05
Tarantilis [55]	2005	Tabu search + adaptive memory		
		Best version	0.18	na
		Standard version	0.20	0.71
Li et al. [40]	2005	Record-to-record travel	0.41[b]	1.05
Ergun et al. [19]	2006	Very large neighbourhood search	0.23	3.88
Mester and Bräysy [42]	2007	Genetic algorithm + very large neighbourhood search		
		Best version	0.03	0.11
		Fast version	0.08	1.01
Derigs and Kaiser [16]	2007	Attribute based tabu search		
		Sav_S	0.21	0.87
		Sav_P	0.28	0.79
Kytöjoki et al. [35]	2007	Guided very large scale neighbourhood search		
		VNS	na	5.63
		GVNS	na	0.79
Pisinger and Ropke [47]	2007	Adaptive large neighbourhood search		
		Average of 10 runs	0.31	1.13
		Best of 10 runs	0.11	0.60
Nagata [45]	2007	Memetic algorithm		
		Average of 10 runs	0.03[b]	3.01[c]
		Best of 10 runs	0.00[b]	2.83[c]

[a]Average on 8 out of 20 instances.
[b]Average on 7 out of 14 instances.
[c]Average on 12 out of 20 instances.

CMT and GWKC test-beds. The % column is the average percentage deviation from the best known solution value. Computing times are highly variable and not reported because of the difficulty of comparing computer speeds. Detailed results are available on the website www.diku.dk/∼sropke. Table 1 requires a word of warning: When reporting results, most researchers (in the field of vehicle routing, but in other fields as well) concentrate on solution quality and computing time. While these two measures are undoubtedly important, they do not tell the whole story. Other qualities such as simplicity of implementation and flexibility are also important. In their quest for accuracy, researchers often introduce too many minor features user-controlled parameters in their algorithms, thus making them less appealing (see Cordeau et al. [11]). It is also important to design algorithms that can easily handle the numerous side constraints that arise in practice. In this respect, UTSA and the general VRP heuristic of Pisinger and Ropke [47] offer the advantage of lending themselves to the solution of several VRP variants.

5. CONCLUSIONS

The VRP is an important and difficult combinatorial optimization problem arising in distribution management. We have outlined the main algorithmic results for the classical VRP involving capacity constraints only. To this day, the VRP remains very difficult to solve optimally. The best exact algorithms, all based on branch-and-cut, can solve some instances involving up to about 100 customers but their performance is not consistent even for such small sizes. In

practice metaheuristics are the preferred solution methodology. The best of these typically yield results within 1% of the best known solution values. Most of the best methods make use of local search, genetic search, or combinations of these two mechanisms.

ACKNOWLEDGMENTS

This work was partially supported by the Canadian Natural Sciences and Engineering Research Council under grant 39682-05. This support is grate fully acknowledged. Thanks are due to Stefan Ropke for his help in compiling Table 1, and to Barry Nelson for his valuable comments.

REFERENCES

[1] K. Altinkemer and B. Gavish, Parallel savings based heuristic for the delivery problem, Oper Res 39 (1991), 456–469.
[2] D.L. Applegate, R.E. Bixby, V. Chvátal, and W.J. Cook, The traveling salesman problem, A computational study, Princeton University Press, Princeton, NJ, 2007.
[3] B.M. Baker and J. Sheasby, Extensions to the generalised assignment heuristic for vehicle routing, Eur J Oper Res 119 (1999), 147–157.
[4] E. Balas and M.W. Padberg, Set partitioning: A survey, SIAM Rev 18 (1976), 710–760.
[5] R. Baldacci, N. Christofides, and A. Mingozzi, An exact algorithm for the vehicle routing problem based on the set partitioning formulation with additional cuts, Math Program Ser A (in press).
[6] R. Baldacci, E. Hadjiconstantinou, and A. Mingozzi, An exact algorithm for the capacitated vehicle routing problem based on a two-commodity network flow formulation, Oper Res 52 (2004), 723–738.
[7] M. Balinski and R. Quandt, On an integer program for a delivery problem, Oper Res 12 (1964), 300–304.
[8] N. Christofides, A. Mingozzi, and P. Toth, "The vehicle routing problem," in: N. Christofides, A. Mingozzi, P. Toth, and C. Sandi (Editors), Combined optimization, Wiley, Chichester, 1979, pp. 315–338.
[9] G. Clarke and J.V. Wright, Scheduling of vehicles from a central depot to a number of delivery points, Oper Res 12 (1964), 568–581.
[10] J.-F. Cordeau, M. Gendreau, A. Hertz, G. Laporte, and J.-S. Sormany, "New heuristics for the vehicle routing problem," in: A. Langevin and D. Riopel (Editors), Logistics systems and optimization, Springer, New York, 2005, pp. 279–297.
[11] J.-F. Cordeau, M. Gendreau, G. Laporte, J.-Y. Potvin, and F. Semet, A guide to vehicle routing heuristics, J Oper Res Soc 53 (2002), 512–522.
[12] J.-F. Cordeau, G. Laporte, and A. Mercier, A unified tabu search heuristic for vehicle routing problems with time windows, J Oper Res Soc 52 (2001), 928–936.
[13] J.-F. Cordeau, G. Laporte, M.W.P. Savelsbergh, and D. Vigo, "Vehicle routing," in: C. Barnhart and G. Laporte (Editors), Transportation, handbooks in operations research and management science, Vol. 14, Elsevier, Amsterdam, 2007, pp. 367–428.
[14] G.B. Dantzig, D.R. Fulkerson, and S.M. Johnson, Solution of a large-scale traveling-salesman problem, Oper Res 2 (1954), 393–410.
[15] G.B. Dantzig and J.H. Ramser, The truck dispatching problem, Manag Sci 6 (1959), 80–91.
[16] U. Derigs and R. Kaiser, Applying the attribute based hill climber heuristic to the vehicle routing problem, Eur J Oper Res 177 (2007), 719–732.
[17] G. Dueck, New optimization heuristics: The great deluge algorithm and the record-to-record travel, J Comput Phy 104 (1993), 86–92.
[18] O. Ergun, New neighborhood search algorithms based on exponentially large scale neighborhoods, Ph.D. Thesis, Massachusetts Institute of Technology, 2001.
[19] O. Ergun, J.B. Orlin, and A. Steele-Feldman, Creating very large-scale neighbourhoods out of smaller ones by compounding moves, J Heuristics 12 (2006), 115–140.
[20] G.A. Finke, A. Claus, and E. Gunn, A two-commodity network flow approach to the traveling salesman problem, Congressus Numerantium 41 (1984), 167–178.
[21] M.L. Fisher and R. Jaikumar, A generalized assignment heuristic for the vehicle routing problem, Networks 11 (1981), 109–124.
[22] B.A. Foster and D.M. Ryan, An integer programming approach to the vehicle scheduling problem, Oper Res 27 (1976), 367–384.
[23] R. Fukasawa, H. Longo, J. Lysgaard, M. Poggi de Aragão, M. Reis, E. Uchoa, and R.F. Werneck, Robust branch-and-cut-and-price for the capacitated vehicle routing problem, Math Program Ser A 106 (2006), 491–511.
[24] M. Gendreau, A. Hertz, and G. Laporte, A tabu search heuristic for the vehicle routing problem, Manag Sci 40 (1994), 1276–1290.
[25] M. Gendreau, J.-Y. Potvin, O. Bräysy, G. Hasle, and A. Løkketangen, "Meta-heuristics for the vehicle routing problem and its extensions: A categorized bibliography," in: B.L. Golden, S. Raghavan, and E.A. Wasil (Editors), The vehicle routing problem: Latest advances and challenges, Springer, Boston, (in press).
[26] H. Ghaziri, "Solving routing problems by a self-organizing map," in: T. Kohonen, K. Makisara, O. Simula, and J. Kangas (Editors), Artificial neural networks, North-Holland, Amsterdam, 1991, pp. 829–834.
[27] B.E. Gillett and L.R. Miller, A heuristic algorithm for the vehicle dispatch problem, Oper Res 22 (1974), 340–349.
[28] F. Glover, Future paths for integer programming and links to artificial intelligence, Comput Oper Res 13 (1986), 533–549.
[29] B.L. Golden, A.A. Assad, and E.A. Wasil, "Routing vehicles in the real world: Applications in the solid waste beverage, food, dairy and newspaper industries," in: P. Toth and D. Vigo (Editors), The vehicle routing problem, SLAM Monographs on discrete mathematics and applications, Society for Industrial and Applied Mathematics, Philadelphia, 2002, pp. 245–286.
[30] B.L. Golden, T.L. Magnanti, and H.Q. Nguyen, Implementing vehicle routing algorithms, Networks 7 (1977), 113–148.
[31] B.L. Golden, E.A. Wasil, J.P. Kelly, and I-M. Chao, "Metaheuristics in vehicle routing," in: T. G. Crainic and G. Laporte (Editors), Fleet management and logistics, Kluwer, Boston, 1998, pp. 33–56.
[32] K. Hoffman and M.W. Padberg, Solving airline crew scheduling problems by branch and cut, Manag Sci 39 (1993), 657–682.

[33] J.H. Holland, Adaptation in natural and artificial systems, The University of Michigan Press, Ann Arbor, MI, 1975.
[34] G.A.P. Kindervater and M.W.P. Savelsbergh, "Vehicle routing: Handling edge exchanges," in: E.H.L. Aarts and J.K. Lenstra (Editors), Local search in combinatorial Optimization, Wiley, Chichester, 1997, pp. 337–360.
[35] J. Kytöjoki, T. Nuortio, O. Bräysy, and M. Gendreau, An efficient variable neighborhood search heuristic for very large scale vehicle routing problems, Comput Oper Res 34 (2007), 2743–2757.
[36] G. Laporte and Y. Nobert, Exact algorithms for the vehicle routing problem, Ann Discrete Math 31 (1987), 147–184.
[37] G. Laporte, Y. Nobert, and M. Desrochers, Optimal routing under capacity and distance restrictions, Oper Res 33 (1985), 1058–1073.
[38] G. Laporte and F. Semet, "Classical heuristics for the capacitated VRP," in: P. Toth and D. Vigo (Editors), The Vehicle routing problem, SIAM monographs on discrete mathematics and applications, Society for Industrial and Applied Mathematics, Philadelphia, 2002, pp. 109–128.
[39] A.N. Letchford, R.W. Eglese, and J. Lysgaard, Multistars, partial multistars and the capacitated vehicle routing problem, Math Progam Ser A 94 (2002), 21–40.
[40] F. Li, B.L. Golden, and E.A. Wasil, Very large-scale vehicle routing: New test problems, algorithms, and results, Comput Oper Res 32 (2005), 1165–1179.
[41] D. Mester and O. Bräysy, Active guided evolution strategies for the large scale vehicle routing problems with time windows, Comput Oper Res 32 (2005), 1593–1614.
[42] D. Mester and O. Bräysy, Active-guided evolution strategies for large-scale capacitated vehicle routing problems, Comput Oper Res 34 (2007), 2964–2975.
[43] N. Mladenović and P. Hansen, Variable neighborhood search, Comput Oper Res 24 (1997), 1097–1100.
[44] D. Naddef and G. Rinaldi, "Branch-and-cut algorithms for the capacitated VRP," in: P. Toth and D. Vigo (Editors), The vehicle routing problem, SIAM monographs on discrete mathematics and applications, Society for Industrial and Applied Mathematics, Philadelphia, 2002, pp. 53–81.
[45] Y. Nagata, "Edge assembly crossover for the capacitated vehicle routing problem," in: C. Cotta and J. van Hemert (Editors), Lecture notes in computer science, vol. 4446, Springer-Verlag, Berlin Heidelberg, 2007, 142–153.
[46] M.D. Nelson, K.E. Nygard, J.H. Griffin, and W.E. Shreve, Implementation techniques for the vehicle routing problem, Comput Oper Res 12 (1985), 273–283.
[47] D. Pisinger and S. Ropke, A general heuristic for vehicle routing problems, Comput Oper Res 34 (2007), 2403–2435.
[48] C. Prins, A simple and effective evolutionary algorithm for the vehicle routing problem, Comput Oper Res 31 (2004), 1985–2002.
[49] M. Reimann, K. Doerner, and R.F. Hartl, D-Ants: Savings based ants divide and conquer the vehicle routing problem, Comput Oper Res 31 (2004), 563–591.
[50] J. Renaud, F.F. Boctor, and G. Laporte, An improved petal heuristic for the vehicle routing problem, J Oper Res Soc 47 (1996), 329–336.
[51] Y. Rochat and É.D. Taillard, Probabilistic diversification and intensification in local search for vehicle routing, J Heuristics 1 (1995), 147–167.
[52] D.M. Ryan, C. Hjorring, and F. Glover, Extentions of the petal method for vehicle routing, J Oper Res Soc 44 (1993), 289–296.
[53] M. Schumann and R. Retzko, "Self-organizing maps for vehicle routing problems – minimizing an explicit cost function," in: F. Fogelman-Soulie (Editor), Proceedings of the International Conference on Artificial Neural Networks, Paris, 1995, pp. 401–406.
[54] É.D. Taillard, Parallel iterative search methods for vehicle routing problem, Networks 23 (1993), 661–673.
[55] C.-D. Tarantilis, Solving the vehicle routing problem with adaptive memory programming methodology, Comput Oper Res 32 (2005), 2309–2327.
[56] C.-D. Tarantilis and C.T. Kiranoudis, BoneRoute: An adaptive memory-based method for effective fleet management, Ann Oper Res 115 (2002), 227–241.
[57] P.M. Thompson and H.M. Psaraftis, Cyclic transfer algorithms for multi-vehicle routing and scheduling problems, Oper Res 41 (1993), 935–946.
[58] P. Toth and D. Vigo, "Exact solution of the vehicle routing problem," in: T.G. Crainic and G. Laporte (Editors), Fleet management and logistics Kluwer, Boston, 1998, pp. 1–31.
[59] P. Toth and D. Vigo (Editors), The vehicle routing problem, SIAM Monographs on discrete mathematics and applications, Philadelphia, 2002.
[60] P. Toth and D. Vigo, The granular tabu search and its application to the vehicle routing problem, INFORMS J Comput 15 (2003), 333–346.
[61] A. van Breedam, An analysis of the behavior of heuristics for the vehicle routing problem for a selection of problems with vehicle-related, customer-related, and time-related constraints, Ph. D. Dissertation, University of Antwerp, 1994.
[62] P. Wark and J. Holt, A repeated matching heuristic for the vehicle routing problem, J Oper Res Soc 45 (1994), 1156–1167.