C. Barnhart and G. Laporte (Eds.), *Handbook in OR & MS, Vol. 14*
Copyright © 2007 Elsevier B.V. All rights reserved
DOI: 10.1016/S0927-0507(06)14006-2

Chapter 6

Vehicle Routing

Jean-François Cordeau

Canada Research Chair in Logistics and Transportation, HEC Montréal,
3000 chemin de la Côte-Sainte-Catherine, Montréal, H3T 2A7, Canada
E-mail: Jean-Francois.Cordeau@hec.ca

Gilbert Laporte

Canada Research Chair in Distribution Management, HEC Montréal,
3000 chemin de la Côte-Sainte-Catherine, Montréal, H3T 2A7, Canada
E-mail: gilbert@crt.umontreal.ca

Martin W.P. Savelsbergh

School of Industrial and Systems Engineering, Georgia Institute of Technology,
Atlanta, GA 30332-0205, USA
E-mail: mwps@isye.gatech.edu

Daniele Vigo

Dipartimento di Elettronica, Informatica e Sistemistica, University of Bologna,
Viale Risorgimento 2, 40136 Bologna, Italy
E-mail: dvigo@deis.umibo.it

1 Introduction

The *vehicle routing problem* lies at the heart of distribution management. It is faced each day by thousands of companies and organizations engaged in the delivery and collection of goods or people. Because conditions vary from one setting to the next, the objectives and constraints encountered in practice are highly variable. Most algorithmic research and software development in this area focus on a limited number of prototype problems. By building enough flexibility in optimization systems one can adapt these to various practical contexts.

Much progress has been made since the publication of the first article on the "truck dispatching" problem by Dantzig and Ramser (1959). Several variants of the basic problem have been put forward. Strong formulations have been proposed, together with polyhedral studies and exact decomposition algorithms. Numerous heuristics have also been developed for vehicle routing problems. In particular the study of this class of problems has stimulated the emergence and the growth of several metaheuristics whose performance is constantly improving.

This chapter focuses on some of the most important vehicle routing problem types. A number of other variants have been treated in recent articles and book chapters (see, e.g., Toth and Vigo, 2002a). The *pickup and delivery vehicle routing problem*, which has also been extensively studied, is covered in the "Transportation on Demand" chapter.

The remainder of this chapter is organized as follows. Section 2 is devoted to the *classical* vehicle routing problem (simply referred to as VRP), defined with a single depot and only capacity and route length constraints. Problems with time windows are surveyed in Section 3. Section 4 is devoted to inventory routing problems which combine routing and customer replenishment decisions. Finally, Section 5 covers the field of stochastic vehicle routing in which some of the problem data are random variables.

2 The classical vehicle routing problem

The Classical Vehicle Routing Problem (VRP) is one of the most popular problems in combinatorial optimization, and its study has given rise to several exact and heuristic solution techniques of general applicability. It generalizes the Traveling Salesman Problem (TSP) and is therefore NP-hard. A recent survey of the VRP can be found in the first six chapters of the book edited by Toth and Vigo (2002a). The aim of this section is to provide a comprehensive overview of the available exact and heuristic algorithms for the VRP, most of which have also been adapted to solve other variants, as will be shown in the remaining sections.

The VRP is often defined under capacity and route length restrictions. When only capacity constraints are present the problem is denoted as CVRP. Most exact algorithms have been developed with capacity constraints in mind but several apply mutatis mutandis to distance constrained problems. In contrast, most heuristics explicitly consider both types of constraint.

2.1 Formulations

The symmetric VRP is defined on a complete undirected graph $G = (V, E)$. The set $V = \{0, \dots, n\}$ is a vertex set. Each vertex $i \in V \backslash \{0\}$ represents a customer having a nonnegative demand q_i, while vertex 0 corresponds to a depot. To each edge $e \in E = \{(i, j)\colon\ i, j \in V, i < j\}$ is associated a travel cost c_e or c_{ij}. A fixed fleet of m identical vehicles, each of capacity Q, is available at the depot. The symmetric VRP calls for the determination of a set of m routes whose total travel cost is minimized and such that: (1) each customer is visited exactly once by one route, (2) each route starts and ends at the depot, (3) the total demand of the customers served by a route does not exceed the vehicle capacity Q, and (4) the length of each route does not exceed a preset limit L. (It is common to assume constant speed so that distances, travel times and travel costs are considered as synonymous.) A solution can be viewed as a set of m cycles sharing a common vertex at the depot. The asymmetric VRP is similarly defined on a directed graph $G = (V, A)$, where $A = \{(i, j)\colon$

$i, j \in V, i \neq j\}$ is an arc set. In this case a circuit (directed cycle) is associated with a vehicle route. Most results of Sections 2.1 and 2.2 apply to the symmetric CVRP.

An integer linear programming formulation of the CVRP follows, where for each edge $e \in E$ the integer variable x_e indicates the number of times edge e is traversed in the solution. Let $r(S)$ denote the minimum number of vehicles needed to serve the customers of a subset S of customers. The value of $r(S)$ may be determined by solving an associated Bin Packing Problem (BPP) with item set S and bins of capacity Q. Finally, for $S \subset V$, let $\delta(S) = \{(i, j)\colon i \in S, j \notin S$ or $i \notin S, j \in S\}$. If $S = \{i\}$, then we simply write $\delta(i)$ instead of $\delta(\{i\})$. The CVRP formulation proposed by Laporte et al. (1985) is then:

$$\text{(CVRP1)} \quad \text{minimize} \quad \sum_{e \in E} c_e x_e \tag{1}$$

subject to

$$\sum_{e \in \delta(i)} x_e = 2, \quad i \in V \setminus \{0\}, \tag{2}$$

$$\sum_{e \in \delta(0)} x_e = 2m, \tag{3}$$

$$\sum_{e \in \delta(S)} x_e \geqslant 2r(S), \quad S \subseteq V \setminus \{0\}, S \neq \varnothing, \tag{4}$$

$$x_e \in \{0, 1\}, \quad e \notin \delta(0), \tag{5}$$

$$x_e \in \{0, 1, 2\}, \quad e \in \delta(0). \tag{6}$$

The degree constraints (2) state that each customer is visited exactly once, whereas the depot degree constraint (3) means that m routes are created. Capacity constraints (4) impose both the connectivity of the solution and the vehicle capacity requirements by forcing a sufficient number of edges to enter each subset of vertices. We note that since the BPP is NP-hard in the strong sense, $r(S)$ may be approximated from below by any BPP lower bound, such as $\lceil \sum_{i \in S} q_i / Q \rceil$. Finally, constraints (5) and (6) impose that each edge between two customers is traversed at most once and each edge incident to the depot is traversed at most twice. In this latter case, the vehicle performs a route visiting a single customer.

A widely used alternative formulation is based on the set partitioning or set covering models. The formulation was originally proposed by Balinski and Quandt (1964) and contains a potentially exponential number of binary variables. Let $\mathcal{R} = \{R_1, \ldots, R_s\}$ denote the collection of all feasible routes, with $s = |\mathcal{R}|$. Each route R_j has an associated cost γ_j, and a_{ij} is a binary coefficient equal to 1 if and only if vertex i is visited (i.e., *covered*) by route R_j. The binary variable x_j, $j = 1, \ldots, s$, is equal to 1 if and only if route R_j is selected in the solution. The model is:

$$\text{(CVRP2)} \quad \text{minimize} \quad \sum_{j=1}^{s} \gamma_j x_j \tag{7}$$

subject to

$$\sum_{j=1}^{s} a_{ij} x_j = 1, \quad i \in V \setminus \{0\}, \tag{8}$$

$$\sum_{j=1}^{s} x_j = m, \tag{9}$$

$$x_j \in \{0, 1\}, \quad j = 1, \ldots, s. \tag{10}$$

Constraints (8) impose that each customer i is covered by exactly one route, and (9) requires that m routes be selected. Because route feasibility is implicitly considered in the definition of $\mathcal{R}$, this is a very general model which may easily take additional constraints into account. Moreover, when the cost matrix satisfies the triangle inequality (i.e., $c_{ij} \leqslant c_{ik} + c_{kj}$ for all $i, j, k \in V$), the set partitioning model CVRP2 may be transformed into an equivalent set covering model CVRP2′ by replacing the equality sign with "$\geqslant$" in (8). Any feasible solution to CVRP2 is clearly feasible for CVRP2′, and any feasible solution to CVRP2′ may be transformed into a feasible CVRP2 solution of smaller or equal cost. Indeed, if the CVRP2′ solution is infeasible for CVRP2, then one or more customers are visited more than once. These customers may therefore be removed from their route by applying shortcuts which will not increase the solution cost because of the triangle inequality. The main advantage of using CVRP2′ is that only inclusion-maximal feasible routes, among those with the same cost, need be considered in the definition of $\mathcal{R}$. This significantly reduces the number of variables. In addition, when using CVRP2′ the dual solution space is considerably reduced since dual variables are restricted to be nonnegative. One of the main drawbacks of models CVRP2 and CVRP2′ lies in their very large number of variables, which in loosely constrained medium size instances may easily run into the billions. Thus, one has to resort to a column generation algorithm to solve these problems. The linear programming relaxation of these models tends to be very tight, as shown by Bramel and Simchi-Levi (1997). Further details on these formulations and their extensions, as well as additional formulations for the symmetric and asymmetric cases, can be found in Laporte and Nobert (1987) and in Toth and Vigo (2002b, 2002d).

2.2 *Exact algorithms for the CVRP*

We now review the main exact approaches presented in the last two decades for the solution of the CVRP. For a thorough review of previous exact methods, see Laporte and Nobert (1987). We first describe the algorithms based on branch-and-bound, including those that make use of the set partitioning formulation and column generation schemes, and we then examine the algorithms based on branch-and-cut. In practice, the CVRP turns out to be significantly harder to solve than the TSP. The best CVRP algorithms can rarely tackle instances involving more than 100 vertices, while TSP instances with hundreds and even thousands of vertices are now routinely solved to optimality.

2.2.1 Branch-and-bound and set partitioning based algorithms

Several branch-and-bound algorithms are available for the solution of the CVRP. Until the late 1980s, the most effective exact methods were mainly branch-and-bound algorithms based on elementary combinatorial relaxations. Recently, more sophisticated bounds have been proposed, namely those based on Lagrangian relaxations or on the additive bounding procedure, which have substantially increased the size of the problems that can be solved to optimality. We now describe some branch-and-bound algorithms with an emphasis on lower bound computations which constitute the most critical component of methods of this type. More details on the structure of branch-and-bound algorithm strategies and dominance rules may be found in Toth and Vigo (1998, 2002c, 2002d). We also review in this section exact set partitioning based algorithms for the CVRP.

Many different elementary combinatorial relaxations were used in early branch-and-bound algorithms, including those based on the Assignment Problem (AP), on the degree-constrained *shortest spanning tree*, and on state-space relaxation. Here we outline the two families of relaxations used as a basis for the more recent branch-and-bound algorithms for the symmetric and asymmetric CVRP. A first relaxation is obtained from the integer programming formulations of these problems by dropping the connectivity and capacity constraints. In the symmetric case the resulting problem is a b-Matching Problem (b-MP), i.e., the problem of determining a minimum cost set of cycles covering all vertices and such that the degree of each vertex i is equal to b_i, where $b_i = 2$ for all the customer vertices, and $b_0 = 2m$ for the depot vertex. It is easy to see that by adding $m - 1$ copies of the depot to G the relaxation becomes a 2-MP. In the asymmetric case the relaxed problem is the well-known transportation problem which may be transformed into an AP by introducing copies of the depot. Also in this case, the AP may be seen as the problem of determining a set of circuits covering all vertices and such that each vertex has one entering and one leaving arc. The solution of these relaxed problems may be infeasible for the CVRP since the demand associated with a cycle or circuit may exceed the vehicle capacity, and some of these may be disconnected from the depot. The relaxed problems may then be solved in polynomial time (see, e.g., Miller and Pekny, 1995, for the b-MP and Dell'Amico and Toth, 2000 for the AP). However, the quality of the lower bounds obtained with these relaxations is generally very poor and not sufficient to solve instances with more than 15 or 20 customers. Toth and Vigo (2002c) report average gaps in excess of 20% with respect to the optimal solution value on benchmark CVRP instances. The situation is slightly better for the AP relaxation of the asymmetric CVRP that yields average gaps of about 10% or less. Laporte et al. (1986) have proposed a branch-and-bound algorithm for asymmetric CVRP, based on the AP relaxation and capable of solving randomly generated problems involving tens of customers and between two and four vehicles.

The second family of elementary relaxations used in recent branch-and-bound algorithms is based on degree-constrained spanning trees. These relaxations extend the well-known 1-tree relaxation proposed by Held and Karp (1971) for the TSP. The earliest branch-and-bound algorithm based on this relaxation, proposed by Christofides et al. (1981a), could only solve relatively small instances. More recently, Fisher (1994) has presented another tree based relaxation requiring the determination of a so-called m-tree, defined as a minimum cost set of $n + m$ edges spanning the graph. The approach used by Fisher is based on CVRP1 with the additional assumption that single-customer routes are not allowed. Fisher modeled the CVRP as the problem of determining an m-tree with degree equal to $2m$ at the depot vertex, with additional constraints on vehicle capacity and a degree of 2 for each customer vertex. The determination of an m-tree with degree $2m$ at the depot requires $O(n^3)$ time. The degree-constrained m-tree relaxation is easily obtained from CVRP1 by removing the degree constraints (2) for customer vertices and weakening the capacity constraints (4) into connectivity constraints, i.e., by replacing their right-hand side with 1. The m-tree solution is not always feasible for the CVRP since some vertices may have a degree different from 2 and the demand associated with the subtrees incident to the depot may exceed the vehicle capacity.

For the asymmetric CVRP, similar relaxations may be derived from directed trees, also called *arborescences*, spanning the graph and having an outdegree equal to m at the depot vertex. To obtain the final bound a minimum cost set of m vertex-disjoint arcs entering the depot are added to the constrained arborescence. In this case, the relaxed subproblem may be solved in polynomial time, but again the quality of the resulting lower bound is very poor. Toth and Vigo (2002c) report that on benchmark asymmetric instances, the average gap of these relaxations with respect to the optimal solution value is larger than 25%.

Different improved bounding techniques were later developed to narrow the gap between the lower bound and the optimal solution value of the CVRP. These include two bounding procedures based on Lagrangian relaxation proposed by Fisher (1994) and Miller (1995). These are strengthenings of the basic CVRP relaxations obtained by dualizing some of the relaxed constraints in a Lagrangian fashion. In particular, they both include in the objective function a suitable subset of the capacity constraints (4), whereas the Fisher relaxation also incorporates degree constraints (2) which were relaxed in the m-tree relaxation. As in related problems, good values for the Lagrangian multipliers associated with the relaxed constraints are determined by using a subgradient optimization procedure (see, e.g., Held and Karp, 1971; Held et al., 1974). The main difficulty associated with these relaxations lies in the exponential cardinality of the set of relaxed constraints which does not allow for their complete inclusion in the objective function. These authors include a limited family $\mathcal{F}$ of capacity constraints and iteratively generate the constraints violated by the current solution of the Lagrangian problem. The process terminates when no violated constraint is detected (hence the Lagrangian solution is feasible) or a preset number of subgradient iterations have been executed. Redundant constraints are periodically removed from $\mathcal{F}$. The relax-and-cut algorithm of Martinhon et al. (2000) generalizes these Lagrangian-based approaches by also considering comb and multistar inequalities, and moderately improves the quality of the Lagrangian bound.

Some exact algorithms for the CVRP are based on the set partitioning formulation CVRP2. The first of these is due to Agarwal et al. (1989) who considered a relaxation of model CVRP2 not including constraints (9) and solved the resulting model through column generation. Agarwal, Mathur, and Salkin used their algorithm to solve seven Euclidean CVRP instances with up to 25 customers. Hadjiconstantinou et al. (1995) proposed a branch-and-bound algorithm in which the lower bound was obtained by considering the dual of the linear relaxation of model CVRP2, following the approach introduced by Mingozzi et al. (1994). By linear programming duality, any feasible solution to this dual problem yields a valid lower bound. Hadjiconstantinou et al. (1995) determined the heuristic dual solutions by combining two relaxations of the original problem: the q-path relaxation of Christofides et al. (1981a) and the m-shortest path relaxation of Christofides and Mingozzi (1989). The algorithm was able to solve randomly generated Euclidean instances with up to 30 vertices and benchmark instances with up to 50 vertices. Further details on set partitioning-based algorithms for the CVRP are provided in Bramel and Simchi-Levi (2002).

Fischetti et al. (1994) have improved the AP relaxation of the asymmetric CVRP by combining into an additive bounding procedure two new lower bounds based on disjunctions on infeasible arc subsets and on minimum cost flows. The additive approach was proposed by Fischetti and Toth (1989) and allows for the combination of different lower bounding procedures, each exploiting a different substructure of the problem under consideration. The resulting branch-and-bound approach was able to solve randomly generated instances containing up to 300 vertices and four vehicles. Other bounds for the asymmetric CVRP may be derived by generalizing the methods proposed for the symmetric case. For example, Fisher (1994) proposed a way of extending to the asymmetric CVRP the Lagrangian bound based on m-trees. In this extension the Lagrangian problem calls for the determination of an undirected m-tree on the undirected graph obtained by replacing each pair of arcs (i, j) and (j, i) with a single edge (i, j) of cost $c'_{ij} = \min\{c_{ij}, c_{ji}\}$. No computational testing for this bound was presented by Fisher (1994). Potentially better bounds may be obtained by explicitly considering the asymmetry of the problem, i.e., by using m-arborescences rather than m-trees and by strengthening the bound in a Lagrangian fashion as proposed by Toth and Vigo (1995, 1997) for the *capacitated shortest spanning arborescence problem* and for the *VRP with backhauls*.

2.2.2 Branch-and-cut algorithms

Branch-and-cut algorithms currently constitute the best available exact approach for the solution of the CVRP. Research in this area has been strongly motivated by the emergence and the success of polyhedral combinatorics as a framework for the solution of hard combinatorial problems, particularly the TSP. However, in a recent survey on branch-and-cut approaches for the CVRP, Naddef and Rinaldi (2002) state: "…the amount of research effort spent to solve CVRP by this method is not comparable with what has been dedicated to the TSP […the research in this field] is still quite limited and most of it is not published yet". In the following we summarize the main available branch-and-cut approaches for the CVRP. The reader is referred to Naddef and Rinaldi (2002) for a more detailed presentation.

The use of branch-and-cut for the CVRP is rooted in the exact algorithm of Laporte et al. (1985). This algorithm uses the Linear Programming (LP) relaxation of model CVRP1 without capacity constraints (4) as a basis for the solution of the VRP with capacity and maximum distance restrictions. This initial relaxation is iteratively strengthened by adding violated capacity constraints which are heuristically separated by considering the connected components induced by the set of nonzero variables in the current LP solution. Gomory cuts are also introduced at the root node of the branch-and-cut tree. The algorithm was capable of solving randomly generated loosely constrained Euclidean and non-Euclidean instances with two or three vehicles and up to 60 customers.

The first polyhedral study of the CVRP was presented by Cornuéjols and Harche (1993). The presence of equalities (2) and (3) makes the CVRP nonfully-dimensional. Therefore, as in the TSP, Cornuéjols and Harche first considered the full-dimensional polyhedron, containing the CVRP polyhedron as a face, associated with the so-called Graphical VRP (GVRP) where customers may be visited more than once. The basic properties of the GVRP polyhedron were also investigated. Conditions under which the nonnegativity, degree and capacity constraints define facets of the GVRP and CVRP polyhedra were also determined. Cornuéjols and Harche have extended to the GVRP and the CVRP several other families of valid inequalities proposed for the TSP and the graphical TSP. In particular, comb, path, wheelbarrow, and bicycle inequalities were extended to the capacitated case and again, sufficient conditions under which these inequalities define facets of the GVRP and CVRP polyhedra were derived. These inequalities were used by Cornuéjols and Harche as cutting planes to solve two instances of CVRP with 18 and 50 customers, within a branch-and-cut algorithm. The detection of violated inequalities was performed manually, starting from the current optimal LP solution.

Augerat et al. (1995) have developed the first complete branch-and-cut approach for the CVRP. They described several heuristic separation procedures for the classes of valid inequalities proposed by Cornuéjols and Harche, as well as four new classes of valid inequalities. Separation procedures were further investigated by Augerat et al. (1999). The resulting approach was able to solve several CVRP instances containing up to 134 customers. Ralphs et al. (2003) have presented a branch-and-cut algorithm for the CVRP in which an exact separation of valid m-TSP inequalities is used in addition to heuristic separation of capacity inequalities. The resulting algorithm was implemented within the SYMPHONY parallel branch-and-cut-and-price framework and was able to solve several instances involving fewer than 100 vertices. Lysgaard et al. (2004) have developed new separation procedures for most of the families of valid inequalities proposed so far (see also Letchford et al., 2002). Their overall branch-and-cut approach, which is further enhanced by the use of Gomory cuts, was able to solve within moderate computing times previously solved instances and three new medium size ones.

Baldacci et al. (2004) have put forward a branch-and-cut algorithm based on a two-commodity network flow formulation of the CVRP and requiring a polynomial number of integer variables. It seems to provide an interesting alternative to other classical formulations (see also Gouveia, 1995, for a single-commodity formulation). The overall algorithm strengthens the LP relaxation by adding violated capacity inequalities and implements various variable reduction and branching rules. The results obtained with this approach are comparable with those of the other branch-and-cut algorithms just described.

Finally, Fukasawa et al. (2006) have proposed a successful branch-and-cut-and-price algorithm combining branch-and-cut with the q-routes relaxation of Christofides et al. (1981a), used here in a column generation fashion. This method produces tighter bounds than other branch-and-cut algorithms and is capable of solving several previously unsolved instances with up to 75 customers. Baldacci et al. (2006) have used their set partitioning algorithm, previously developed for a rollon–rolloff VRP, to solve difficult CVRP instances. Their approach yields bounds whose quality is comparable to those of Fukasawa et al. (2006), but seems much quicker.

Other branch-and-cut algorithms are described in Achuthan et al. (1996, 2003) and Blasum and Hochstättler (2000). We also mention that the polyhedral structure of the special case of CVRP where all the customers have a unit demand was studied by Campos et al. (1991) and by Araque et al. (1990). Branch-and-cut algorithms for this problem are presented by Araque et al. (1994) and by Ghiani et al. (2006).

2.3 Heuristics for the VRP

An impressive number of heuristics have been proposed for the VRP. Initially these were mainly standard route construction algorithms, whereas more recently powerful metaheuristic approaches have been developed. In the following we separately review these two families of algorithms. Almost all of these methods were developed, described and tested for the symmetric VRP. In addition, since finding a feasible solution with exactly m vehicles is itself an NP-complete problem, almost all methods assume an unlimited number of available vehicles. However, it should be observed that many of the proposed methods may be quite easily adapted to take into account additional practical constraints, although these may affect their overall performance (see, e.g., Vigo, 1996, for an extension of some classical heuristics to the asymmetric case).

2.3.1 Classical heuristics

Using the classification proposed by Laporte and Semet (2002), we describe classical VRP heuristics under these headings: route construction methods, two-phase methods, and route improvement methods.

Route construction heuristics. Route construction methods were among the first heuristics for the CVRP and still form the core of many software implementations for various routing applications. These algorithms typically start from an empty solution and iteratively build routes by inserting one or more customers at each iteration, until all customers are routed. Construction algorithms are further subdivided into sequential and parallel, depending on the number of eligible routes for the insertion of a customer. Sequential methods expand only one route at a time, whereas parallel methods consider more than one route simultaneously. Route construction algorithms are fully specified by their three main ingredients, namely an initialization criterion, a selection criterion specifying which customers are chosen for insertion at the current iteration, and an insertion criterion to decide where to locate the chosen customers into the current routes.

The first and most famous heuristic of this group was proposed by Clarke and Wright (1964) and is based on the concept of *saving*, an estimate of the cost reduction obtained by serving two customers sequentially in the same route, rather than in two separate ones. If i is the last customer of a route and j is the first customer of another route, the associated saving is defined as $s_{ij} = c_{i0} + c_{0j} - c_{ij}$. If s_{ij} is positive, then serving i and j consecutively in a route is profitable. The Clarke and Wright algorithm considers all customer pairs and sorts the savings in nonincreasing order. Starting with a solution in which each customer appears separately in a route, the customer pair list is examined and two routes are merged whenever this is feasible. Generally, a route merge is accepted only if the associated saving is nonnegative but, if the number of vehicles is to be minimized, then negative saving merges may also be considered. The Clarke and Wright algorithm is inherently parallel since more than one route is active at any time. However, it may easily be implemented in a sequential fashion. The resulting algorithm is quite fast but may have a poor performance (see, e.g., Laporte and Semet, 2002). Golden et al. (1977), Paessens (1988), and Nelson et al. (1985) have proposed various enhancement strategies of the savings approach aimed at improving either its effectiveness or its computational efficiency by means of better data structures. Other attempts to improve the effectiveness of the savings method were made by Desrochers and Verhoog (1989), Altinkemer and Gavish (1991), and by Wark and Holt

(1994) who proposed to implement route merges by using a matching algorithm, together with a more sophisticated estimate of actual merge savings. The results obtained with these algorithms are in general better than those of previous savings methods, but matching-based algorithms require much larger computing times.

Another classical route construction heuristic is the sequential insertion algorithm of Mole and Jameson (1976). The algorithm uses as selection and insertion criterion the evaluation of the extra distance resulting from the insertion of an unrouted customer k between two consecutive customers i and j of the current route, namely $\alpha(i, k, j) = c_{ik} + c_{kj} - \lambda c_{ij}$, where λ is a user-controlled parameter. Variations of this criterion taking into account other factors, such as the distance of the customer from the depot, were also considered. After each insertion, the current route is possibly improved by using a 3-opt procedure. A more general and effective two-step insertion heuristic was proposed by Christofides et al. (1979). In the first step, a sequential insertion algorithm is used to determine a set of feasible routes. The second step is a parallel insertion approach. For each route determined in the first step, a representative customer is selected and a set of single-customer routes is initialized with these customers. The remaining unrouted customers are then inserted by using a regret criterion, where the difference between the best and the second-best insertion cost is taken into account, and partial routes are improved by means of a 3-opt procedure. The resulting algorithm is superior to that of Mole and Jameson and represents a good compromise between effectiveness and efficiency.

Two-phase heuristics. Two-phase methods are based on the decomposition of the VRP solution process into the two separate subproblems:

(1) clustering: determine a partition of the customers into subsets, each corresponding to a route, and
(2) routing: determine the sequence of customers on each route.

In a cluster-first-route-second method, customers are first grouped into clusters and the routes are then determined by suitably sequencing the customers within each cluster. Different techniques have been proposed for the clustering phase, while the routing phase amounts to solving a TSP.

The *sweep* algorithm, due to Wren (1971), Wren and Holliday (1972), and Gillett and Miller (1974), is often referred to as the first example of cluster-first-route-second approach. The algorithm applies to planar VRP instances. The algorithm starts with an arbitrary customer and then sequentially assigns the remaining customers to the current vehicle by considering them in order of increasing polar angle with respect to the depot and the initial customer. As soon as the current customer cannot be feasibly assigned to the current vehicle, a new route is initialized with it. Once all customers are assigned to vehicles, each route is separately defined by solving a TSP. Another early two-phase method is the truncated branch-and-bound method of Christofides et al.

(1979) in which the set of routes is determined through an adaptation of an exact branch-and-bound algorithm that uses a branching-on-routes strategy. The decision tree contains as many levels as the number of available vehicles, and at each level of the decision tree a given node corresponds to a partial solution made up of some complete routes. The descendant nodes correspond to all possible routes including a subset of the unrouted customers. The running time of the algorithm is controlled by limiting to one the number of routes generated at each level.

The Fisher and Jaikumar (1981) algorithm solves the clustering step by means of an appropriately defined Generalized Assignment Problem (GAP) which calls for the determination of a minimum cost assignment of items to a given set of bins of capacity Q, and where the items are characterized by a weight and an assignment cost for each bin. Each vehicle is assigned a representative customer, called a *seed*, and the assignment cost of a customer to a vehicle is equal to its distance to the seed. The GAP is then solved, either optimally or heuristically, and the final routes are determined by solving a TSP on each cluster.

Another two-phase method working with a fixed number m of vehicles was described by Bramel and Simchi-Levi (1995). This algorithm determines route seeds by solving a capacitated location problem, where m customers are selected by minimizing the total distance between each customer and its closest seed, and by imposing that the total demand associated with each seed be at most Q. Once seeds have been determined and the single-customer routes are initialized, the remaining customers are inserted in the current routes by minimizing insertion costs. Various ways of approximating the insertion cost are proposed and analyzed. It is worth noting that all three cluster-first-route-second approaches just described allow for a direct control of the number of routes in the final solution, whereas the sweep algorithm does not. The performance of these algorithms is generally comparable to that of route construction algorithms in terms of effectiveness. The location based approach of Bramel and Simchi-Levi produces better solutions but requires much larger computing times.

A different family of two-phase methods is the class of so-called *petal* algorithms. These generate a large set of feasible routes, called petals, and select the final subset by solving a set partitioning model. Foster and Ryan (1976) and Ryan et al. (1993) have proposed heuristic rules for determining the set of routes to be selected, while Renaud et al. (1996b) have described an extension that considers more involved configurations, called 2-petals, consisting of two embedded or intersecting routes. The overall performance of these algorithms is generally superior to that of the sweep algorithm.

Finally, in route-first-cluster-second methods, a giant TSP tour over all customers is constructed in a first phase and later subdivided into feasible routes. Examples of such algorithms are given by Beasley (1983), Haimovich and Rinnooy Kan (1985), and Bertsimas and Simchi-Levi (1996), but the performance of this approach is generally poor.

Route improvement heuristics. Local search algorithms are often used to improve initial solutions generated by other heuristics. Starting from a given solution, a local search method applies simple modifications, such as arc exchanges or customer movements, to obtain neighbor solutions of possibly better cost. If an improving solution is found, it then becomes the current solution and the process iterates; otherwise a local minimum has been identified.

A large variety of neighborhoods are available. These may be subdivided into *intra-route* neighborhoods, if they operate on a single route at a time, or *inter-route* neighborhoods if they consider more than one route simultaneously. The most common neighborhood type is the λ-opt heuristic of Lin (1965) for the TSP, where λ edges are removed from the current solution and replaced by λ others. The computing time required to examine all neighbors of a solution is proportional to n^{λ}. Thus, only $\lambda = 2$ or 3 are used in practice. As an alternative, one can use restricted neighborhoods characterized by subsets of moves associated with larger λ values, such as Or-exchanges (Or, 1976) or the 4-opt* neighborhood of Renaud et al. (1996a) which considers only a subset of all potential 4-opt exchanges. Laporte and Semet (2002) have conducted a computational comparison of some basic route improvement procedures. More complex inter-route neighborhoods are analyzed by Thompson and Psaraftis (1993), Van Breedam (1994), and Kindervater and Savelsbergh (1997).

2.3.2 *Metaheuristics*

Several metaheuristics have been applied to the VRP. With respect to classical heuristics, they perform a more thorough search of the solution space and are less likely to end with a local optimum. These can be broadly divided into three classes:

(1) local search, including simulated annealing, deterministic annealing, and tabu search;
(2) population search, including genetic search and adaptive memory procedures;
(3) learning mechanisms, including neural networks and ant colony optimization.

The best heuristics often combine ideas borrowed from different metaheuristic principles. Recent surveys of VRP metaheuristics can be found in Gendreau et al. (2002), Cordeau and Laporte (2004), and Cordeau et al. (2005).

Local search algorithms explore the solution space by iteratively moving from a solution x_t at iteration t to a solution x_{t+1} in the neighborhood $N(x_t)$ of x_t until a stopping criterion is satisfied. If $f(x)$ denotes the cost of solution x, then $f(x_{t+1})$ is not necessarily smaller than $f(x_t)$. As a result, mechanisms must be implemented to avoid cycling. In simulated annealing, a solution x is drawn randomly from $N(x_t)$. If $f(x) \leqslant f(x_t)$, then $x_{t+1} := x$. Otherwise,

$$x_{t+1} := \begin{cases} x & \text{with probability } p_t, \\ x_t & \text{with probability } 1 - p_t, \end{cases}$$

where p_t is a decreasing function of t and of $f(x) - f(x_t)$. This probability is often equal to

$$p_t = \exp\left(-\frac{f(x) - f(x_t)}{\theta_t}\right),$$

where θ_t is the *temperature* at iteration t, usually defined as a nonincreasing function of t. Deterministic annealing (Dueck, 1990, 1993) is similar. There are two main versions of this algorithm: in a threshold-accepting algorithm, $x_{t+1} := x$ if $f(x) < f(x_t) + \theta_1$, where θ_1 is a user controlled parameter; in record-to-record travel, a record is the best known solution x^*, and $x_{t+1} := x$ if $f(x_{t+1}) < \theta_2 f(x^*)$, where θ_2 is also user controlled. In tabu search, in order to avoid cycling, any solution possessing some given attribute of x_{t+1} is declared tabu for a number of iterations. At iteration t, the search moves to the best nontabu solution x in $N(x_t)$. These local search algorithms are rarely implemented in their basic version, and their success depends on the careful implementation of several mechanisms. The rule employed to define neighborhoods is critical to most local search heuristics. In simulated annealing several rules have been proposed to define θ_t (see Osman, 1993). Tabu search relies on various strategies to implement tabu tenures (also known as short term memory), search diversification (also known as long term memory), and search intensification which accentuates the search in a promising region.

Population search algorithms operate on several generations of solution populations. In genetic search it is common to repeat the following operation k times: extract two parent solutions from the populations to create two offspring using a crossover operation, and apply a mutation operation to each offspring; then remove the $2k$ worst elements from the population and replace them with the $2k$ offspring. Several crossover rules are available for sequencing problems (Bean, 1994; Potvin, 1996; Drezner, 2003; Prins, 2004). In adaptive memory procedures, an offspring is created by extracting and recombining elements of several parents. In the initial version proposed by Rochat and Taillard (1995) for the VRP, nonoverlapping routes are extracted from several parents to create a partial solution. This solution is then gradually completed and optimized by tabu search.

Neural networks are models composed of richly interconnected units through weighted links, like neurons in the brain. They gradually construct a solution through a feedback mechanism that modifies the link weights to better match an observed output to a described output. In the field of vehicle routing neural network models called the elastic net and the self-organizing map are deformable templates that adjust themselves to the contour of the vertices to generate a feasible VRP solution. An example is provided by Ghaziri (1993). Ant colony algorithms (see Dorigo et al., 1999) also use a learning mechanism. They are derived from an analogy with ants which lay some pheromone on their trail when foraging for food. With time more pheromone is deposited on the more frequented trails. When constructing a VRP solution a move can be assigned a higher probability of being selected if it has previously led to a better solution in previous iterations.

In what follows we summarize the most effective metaheuristics for the CVRP. Initially the best methods were almost exclusively based on tabu search but in recent years several excellent methods inspired from different paradigms have been proposed.

Local search heuristics. A limited number of simulated annealing heuristics for the CVRP were proposed in the early 1990s. Osman's implementation (Osman, 1993) is the most involved and also the most successful. It defines neighborhoods by means of a 2-interchange scheme and applies a different rule of temperature changes. Instead of using a nonincreasing function, as do most authors in the field, Osman decreases θ_t continuously as long as the solution improves, but whenever $x_{t+1} = x_t$, θ_t is either halved or replaced by the temperature at which the incumbent was identified. This algorithm succeeded in producing good solutions but was not competitive with the best tabu search implementations available at the same period.

A large number of tabu search algorithms have been produced over the past fifteen years (a survey is available in Cordeau and Laporte, 2004). In the first known implementation, due to Willard (1989), a CVRP solution is represented as a giant tour containing several copies of the depot and inter-depot chains corresponding to feasible vehicle routes, and neighborhoods are defined by means of 3-opt exchanges. The method was soon to be superseded by more powerful algorithms, including those of Osman (1993), Taillard (1993), and Gendreau et al. (1994).

Taillard's algorithm remains to this day one of the most successful tabu search implementations for the CVRP. It is based on the use of an 1-interchange mechanism to define neighbor solutions, combined with periodic route reoptimizations by means of an exact TSP algorithm (Volgenant and Jonker, 1983). The algorithm also uses random tabu durations. A continuous diversification mechanism that penalizes frequently performed moves is implemented in order to provide a more thorough exploration of the search space. Finally, Taillard's algorithm employs a decomposition scheme that allows for the use of parallel computing. In planar problems the customer set is partitioned into sectors and then concentric rings, while in random instances the regions are defined by means of shortest spanning arborescences rooted at the depot. The region boundaries are periodically updated to produce a diversification effect.

The Taburoute algorithm of Gendreau et al. (1994) moves at each iteration a vertex from its current route to another route containing one of its closest neighbors. Insertions are performed simultaneously with a local reoptimization of the route, based on the GENI procedure (Gendreau et al., 1992). Only a subset of vertices are considered for reinsertion at any given iteration. No vertex can return to its former route during the next θ iterations, where θ is randomly selected in a closed interval. Taburoute also uses a continuous diversification mechanism. During the course of the search infeasible solutions are penalized. This mechanism is implemented by replacing the solution value $f(x)$ associated at solution x with a penalized objective $f'(x) = f(x) + \alpha Q(x) + \beta L(x)$, where $Q(x)$ is the total capacity violation of solution x and $L(x)$ is the total route length violation. The two parameters α and β self-adjust during the search to produce a mix of feasible and infeasible solutions: every μ iterations, α (resp. β) is divided by 2 if the past μ solutions were feasible with respect to capacity (resp. route length), or multiplied by 2 if they were all infeasible with respect to capacity (resp. route length). Other features of Taburoute include the use of random tabu durations, periodic route reoptimizations by means of the US procedure of Gendreau et al. (1992), false starts to initialize the search, and a final intensification phase around the best known solution.

The Rego and Roucairol (1996) Tabuchain algorithm is based on the use of ejection chains involving ℓ routes to define neighborhoods. This process bumps a vertex from one route of the chain to another route. The last bumped vertex may be relocated in the position of the first bumped vertex or elsewhere. The process ensures that no arc or edge is considered more than once in the solution. As in Taburoute, intermediate infeasible solutions are allowed. The authors have also implemented a sequential and a parallel version of their method. Another ejection scheme, called Flower, was later developed by Rego (1998). It is based on the idea of exploiting the representation of routes as blossoms and of paths as stems, and of performing ejection moves by means of edge deletions and creations. This method was not as successful as Tabuchain. Another method employing ejection chains was developed by Xu and Kelly (1996). It oscillates between ejection chains and vertex swaps between two routes. The ejection chains are obtained by solving an auxiliary network flow problem. On the whole this method succeeded in obtaining several good CVRP solutions on benchmark instances but it is rather involved and time consuming.

More recently, Ergun et al. (2003) have developed a Very Large Neighborhood Search (VLNS) algorithm for the VRP. This algorithm operates on several routes simultaneously, not unlike what is done in cyclic transfers (Thompson and Psaraftis, 1993) or in ejection chains. Neighborhoods are defined by a combination of 2-opt moves, vertex swaps between routes, and vertex insertions in different routes. The best choice of moves and of routes involved in the moves is determined through the solution of a network flow problem on an auxiliary graph. One advantage of VLNS is that it allows a broad search by acting on several routes at once. Its main disadvantage lies in the effort required at each iteration to perform moves.

A very useful concept put forward by Toth and Vigo (2003) is that of Granular Tabu Search (GTS). This algorithm a priori removes from the graph long edges that are unlikely to belong to an optimal solution. To determine these edges, the problem is first solved by means of a fast heuristic, e.g., the Clarke and Wright (1964) algorithm, and the average edge cost $\bar{c}$ in this solution is determined. Then only two families of edges are retained: those incident to the depot, and those whose cost does not exceed $\beta\bar{c}$, where β is a user-defined sparsification parameter. The authors show that on benchmark instances, choosing β in [1.0, 2.0] yields the elimination of between 80–90% of all edges. Granular tabu search was implemented in conjunction with some of the features of Taillard's algorithm (Taillard, 1993) and Taburoute (Gendreau et al., 1994), and neighbor solutions were obtained by performing intra-route and inter-route exchanges.

Deterministic annealing was first applied to the VRP by Golden et al. (1998) and more recently by Li et al. (2005). The latter algorithm combines the record-to-record principle of Dueck (1993) with GTS. It works on a sparsified graph containing only a proportion α of the 40 shortest edges incident to each vertex, where α varies throughout the algorithm. The algorithm is applied several times from three initial solutions generated by the Clarke and Wright (1964) algorithm, with savings s_{ij} defined as $c_{i0} + c_{0j} - \lambda c_{ij}$, and $\lambda = 0.6, 1.4,$ and 1.6. Neighbors are defined by means of intra- and inter-route 2-opt moves, and nonimproving solutions are accepted as long as their cost does not exceed that of the incumbent by more than 1%. Whenever the solution has not improved for a number of iterations, a perturbation is applied to the best known solution to restart the search. This is achieved by temporarily moving some vertices to different positions.

Population search heuristics. The Adaptive Memory Procedure (AMP) put forward by Rochat and Taillard (1995) constitutes a major contribution to the field of metaheuristics. Initially developed in the context of the VRP, it is of general applicability and has been used, for example, to solve political districting problems (Bozkaya et al., 2003). An adaptive memory is a pool of good solutions which is updated by replacing its worst elements with better ones. In order to generate a new solution, several solutions are selected from the pool and recombined. In the context of the VRP, vehicle routes are extracted from these solutions and used as the basis of a new solution. The extraction process is applied as long as it is possible to identify routes that do not overlap with previously selected routes. When this is no longer possible, a search process (e.g., tabu search) is initiated from a partially constructed solution made up of the selected routes and some unrouted customers. Any solution constructed in this fashion replaces the worst solution of the pool if it has a better cost. Tarantilis and Kiranoudis (2002) have proposed a variant to this scheme. In a first phase a solution is obtained by means of the Paessens (1988) constructive procedure, which is an application of the Clarke and Wright savings heuristic followed by 2-opt moves, vertex swaps between routes, and vertex reinsertions. In order to generate new solutions from the adaptive memory, Tarantilis and Kiranoudis extract route segments, called bones, as opposed to full vehicle routes as did Rochat and Taillard.

Prins (2004) has developed an algorithm combining two main features of evolutionary search, namely crossovers and mutations. Crossovers consist of creating offspring solutions from parents, while mutations are obtained here by applying a local search algorithm to an offspring. This combination of solution recombination and local search is sometimes referred to as a memetic algorithm (Moscato and Cotta, 2003). In this algorithm, solutions are represented as a giant tour without trip delimiters. To create an offspring from two parents, a chain $(i, \dots, j)$ is first selected from the first parent and the vertices of the second parent are scanned from position $j + 1$ by skipping those of the chain $(i, \dots, j)$. A second offspring is generated in a similar way by reversing the roles of the two parents. Offspring are improved by applying a combination of vertex and edge reinsertions, vertex swaps, combined vertex and edge swaps.

Two other memetic algorithms have recently been proposed by Berger and Barkaoui (2004) and by Mester and Bräysy (2005). The first works on two populations whose sizes are kept constant through the replacement of parents by newly created offspring, and migrations take place between the two populations. Offspring are obtained by combining routes from two parents as long as this can be done without overlapping, and by inserting the unrouted customers according to a proximity criterion. A VLNS heuristic (Shaw, 1998) combining three insertion mechanisms is then applied to the offspring, followed by an improvement scheme consisting of removing vertices from the solution and reinserting them by means of the I1 procedure of Solomon (1987).

The Active Guided Evolution Strategies (AGES) of Mester and Bräysy was initially developed to solve the VRP with time windows and was later applied to the classical VRP. It combines local search (Voudouris, 1997) with an evolution strategy (Rechenberg, 1973) to produce an iterative two-stage procedure. The evolutionary strategy uses a deterministic rule to select a parent solution and create a single offspring from a single parent. The offspring replaces the parent if it improves upon it. Offspring are improved by means of an elaborate search procedure combining granular tabu search, continuous diversification, vertex swaps and moves, 2-opt* moves (Potvin and Rousseau, 1995), VLNS (Shaw, 1998), and restarts.

Learning mechanisms. A limited number of heuristics based on learning mechanisms have been proposed for the VRP. None of the known neural networks based methods is satisfactory, and the early ant colony based heuristics could not compete with the best available approaches. Recently, however, Reimann et al. (2004) have proposed a well-performing heuristics called D-ants. The method repeatedly applies two phases until a stopping criterion is reached. In the first phase, a first generation of good solutions is generated through the applications of a savings based heuristic (Clarke and Wright, 1964) and a 2-opt improvement procedure is applied to each solution. New generations of solutions are then created by benefiting from the knowledge gained in producing past generations. Thus, instead of using the standard savings $s_{ij} = c_{i0} + c_{0j} - c_{ij}$, an attractiveness value $\chi_{ij} = \tau_{ij}^{\alpha} s_{ij}^{\beta}$ is now employed, where τ_{ij}^{α} contains information on how good linking i and j turned out to be in previous generations, and α and β are user-controlled parameters. Vertices

i and j are linked with probability $p_{ij} = \chi_{ij}/(\sum_{(h,\ell)\in\Omega_k} \chi_{h\ell})$, where Ω_k is the set of the feasible (i, j) pairs yielding the k best savings. In the second phase the best solution identified in the first phase is decomposed into subproblems which are then reoptimized using the procedure used in the first phase.

Computational comparison of metaheuristics. Cordeau et al. (2005) provide a computational comparison of recent VRP heuristics on the 14 Christofides et al. (1979) instances ($50 \leqslant n \leqslant 199$) and on the 20 larger Li et al. (2005) instances ($200 \leqslant n \leqslant 480$). Most metaheuristics used in the comparison consistently yield solutions whose value lies within 1% of the best known value.

On the Christofides et al. (1979) instances, the best solutions are obtained by Taillard (1993), Rochat and Taillard (1995), and Mester and Bräysy (2005). If the two instance sets are taken together, the best performers, in terms of accuracy and computing time are probably Mester and Bräysy (2005), Tarantilis and Kiranoudis (2002), and Prins (2004). It should be noted that these three methods all combine population search and local search, thus allowing for a broad and deep exploration of the solution space.

As noted by Cordeau et al. (2002b) heuristics should not be judged solely on speed and accuracy. Simplicity and flexibility are also important. In this respect the Li et al. (2005) record-to-record algorithm is rather interesting: this algorithm possesses a simple structure and is capable of generating very high quality solutions. As far as flexibility is concerned, the granularity principle (Toth and Vigo, 2003) and the adaptive memory concept (Rochat and Taillard, 1995) are general and useful ideas which can easily be applied to other problems.

3 The vehicle routing problem with time windows

The Vehicle Routing Problem with Time Windows (VRPTW) is an important generalization of the classical VRP in which service at every customer i must start within a given time window $[a_i, b_i]$. A vehicle is allowed to arrive before a_i and wait until the customer becomes available, but arrivals after b_i are prohibited. The VRPTW has numerous applications in distribution management. Common examples are beverage and food delivery, newspaper delivery, and commercial and industrial waste collection (see, e.g., Golden et al., 2002).

The VRPTW is NP-hard since it generalizes the CVRP which is obtained when $a_i = 0$ and $b_i = \infty$ for every customer i. In the case of a fixed fleet size, even finding a feasible solution to the VRPTW is itself an NP-complete problem (Savelsbergh, 1985). As a result, research on the VRPTW has concentrated on heuristics. Nevertheless, when the problem is sufficiently constrained (i.e., when time windows are sufficiently narrow), realistic size instances can be solved optimally through mathematical programming techniques. This section presents a mathematical formulation of the VRPTW followed by a description of some of the most important available exact and heuristic algorithms. It is worth pointing out that while exact methods usually minimize distance, most heuristics consider a hierarchical objective which first minimizes the number of vehicles used and then distance.

3.1 Formulation of the VRPTW

The VRPTW can be defined on a directed graph $G = (V, A)$, where $|V| = n + 2$, and the depot is represented by the two vertices 0 and $n + 1$. Feasible vehicle routes then correspond to paths starting at vertex 0 and ending at vertex $n + 1$. The set of vehicles is denoted by K, with $|K| = m$. Let s_i denote the service time at i (with $s_0 = s_{n+1} = 0$) and let t_{ij} be the travel time from i to j. In addition to the time window $[a_i, b_i]$ associated with each vertex $i \in N = V \setminus \{0, n+1\}$, time windows $[a_0, b_0]$ and $[a_{n+1}, b_{n+1}]$ can also be associated with the depot vertex. If no particular restrictions are imposed on vehicle availability, one may simply set $a_0 = \min_{i\in N}\{a_i - t_{0i}\}$, $b_0 = \max_{i\in N}\{b_i - t_{0i}\}$, $a_{n+1} = \min_{i\in N}\{a_i + s_i + t_{i,n+1}\}$, and $b_{n+1} = \max_{i\in N}\{b_i + s_i + t_{i,n+1}\}$. As in the CVRP, let q_i denote the demand of customer i, and let Q be the vehicle capacity.

While several models are available for the VRPTW, this problem is often formulated as a multicommodity network flow model with time window and capacity constraints. This model involves two types of variables: binary variables x_{ij}^k, $(i, j) \in A$, $k \in K$, equal to 1 if and only if arc (i, j) is used by vehicle k, and continuous variables w_i^k, $i \in N$, $k \in K$, indicating the time at which vehicle k starts servicing vertex i. Let $\delta^+(i) = \{j\colon (i, j) \in A\}$ and $\delta^-(j) = \{i\colon (i, j) \in A\}$. The problem can then be stated as follows (see, e.g., Desrochers et al., 1988):

$$\text{minimize} \quad \sum_{k\in K} \sum_{(i,j)\in A} c_{ij} x_{ij}^k \tag{11}$$

subject to

$$\sum_{k\in K} \sum_{j\in\delta^+(i)} x_{ij}^k = 1, \quad i \in N, \tag{12}$$

$$\sum_{j\in\delta^+(0)} x_{0j}^k = 1, \quad k \in K, \tag{13}$$

$$\sum_{i\in\delta^-(j)} x_{ij}^k - \sum_{i\in\delta^+(j)} x_{ji}^k = 0, \quad k \in K, j \in N, \tag{14}$$

$$\sum_{i\in\delta^-(n+1)} x_{i,n+1}^k = 1, \quad k \in K, \tag{15}$$

$$x_{ij}^k\bigl(w_i^k + s_i + t_{ij} - w_j^k\bigr) \leqslant 0, \quad k \in K, (i, j) \in A, \tag{16}$$

$$a_i \leqslant w_i^k \leqslant b_i, \quad k \in K, i \in V, \tag{17}$$

$$\sum_{i\in N} q_i \sum_{j\in\delta^+(i)} x_{ij}^k \leqslant Q, \quad k \in K, \tag{18}$$

$$x_{ij}^k \in \{0,1\}, \quad k \in K, (i,j) \in A. \tag{19}$$

The objective function (11) minimizes the total routing cost. Constraints (12) state that each customer is visited exactly once, while constraints (13)–(15) ensure that each vehicle is used exactly once and that flow conservation is satisfied at each customer vertex. The consistency of the time variables w_i^k is ensured through constraints (16) while time windows are imposed by (17). These constraints also eliminate subtours. Finally, constraints (18) enforce the vehicle capacity restriction.

Formulation (11)–(19) is nonlinear because of constraints (16). These constraints can, however, be linearized as follows:

$$w_j^k \geqslant w_i^k + s_i + t_{ij} - M_{ij}\big(1 - x_{ij}^k\big), \quad k \in K, (i,j) \in A, \tag{20}$$

where $M_{ij} = \max\{0, b_i + s_i + t_{ij} - a_j\}$ is a constant. As suggested by Desrochers and Laporte (1991), the bounds on the time variables b_i^k can also be strengthened:

$$w_i^k \geqslant a_i + \sum_{j \in \delta^-(i)} \max\{0, a_j - a_i + s_j + t_{ji}\} x_{ji}^k, \quad k \in K, i \in V, \tag{21}$$

$$w_i^k \leqslant b_i - \sum_{j \in \delta^+(i)} \max\{0, b_i - b_j + s_i + t_{ij}\} x_{ij}^k, \quad k \in K, i \in V. \tag{22}$$

3.2 Exact algorithms for the VRPTW

As for most other vehicle routing problems, it is difficult to solve the VRPTW exactly through classical simplex-based branch-and-bound methods, even for small instances. This is in large part explained by the fact that the LP relaxation of the problem provides a weak lower bound. The first optimization algorithm for the VRPTW can be attributed to Kolen et al. (1987) who used dynamic programming coupled with state space relaxation (Christofides et al., 1981b) to compute lower bounds within a branch-and-bound algorithm. Instances with $n \leqslant 15$ were solved using this approach. Most subsequent algorithms rely either on the generation of valid inequalities to strengthen the LP relaxation or on mathematical decomposition techniques. This section reviews the three main available approaches: Lagrangian relaxation, column generation, and branch-and-cut. Additional references on the subject can also be found in the Cordeau et al. (2002a) review.

3.2.1 Lagrangian relaxation based algorithms

Lagrangian relaxation can be applied to the VRPTW in several ways. It is well known that when the subproblem obtained by relaxing some of the constraints possesses the integrality property, the best lower bound obtained by Lagrangian relaxation (i.e., the value of the Lagrangian dual) is equal to the value of the linear programming relaxation of the original problem. But as mentioned above, the LP relaxation of formulation (11)–(19) provides a weak lower bound which will usually prevent the problem from being solved by branch-and-bound. As a result, successful implementations of Lagrangian relaxation for the VRPTW should retain at least some of the complicating constraints in the subproblem.

Fisher (1994) and Fisher et al. (1997) have described Lagrangian relaxation based on m-trees (see Section 2.2.1). This approach relaxes the flow conservation constraints as well as the capacity and time window constraints. Violated capacity constraints are handled by identifying subsets of customers $S \subseteq N$ that must be visited by at least $\kappa(S)$ vehicles and imposing the constraint

$$\sum_{k \in K} \sum_{i \in V \setminus S} \sum_{j \in S} x_{ij}^k \geqslant \kappa(S). \tag{23}$$

These constraints are relaxed in a Lagrangian fashion so that the resulting problem remains an m-tree problem with modified costs. Time windows are handled similarly by identifying infeasible paths and imposing the constraint that at least one arc in the path be left out of the solution. This approach has solved a few of the Solomon (1987) test instances with $n = 100$. In addition to the m-tree relaxation method, Fisher et al. (1997) have also experimented with a variable splitting approach in which additional variables y_i^k, equal to 1 if and only if customer i is visited by vehicle k, are introduced in the formulation, and the constraints $\sum_{j \in V} x_{ij}^k = y_i^k$ $(i \in N, k \in K)$ are dualized. The Lagrangian subproblem decomposes into a semi-assignment problem in the y_i^k variables which is solvable by inspection, and a set of m elementary shortest path problems with time windows and capacity constraints.

Another possible Lagrangian relaxation consists of dualizing the demand constraints. Let $\boldsymbol{\lambda} = (\lambda_i)$ $(i \in N)$ be the vector of multipliers associated with constraints (12) requiring that each customer be visited exactly once. For given values of the multipliers, the Lagrangian subproblem $L(\boldsymbol{\lambda})$ obtained by relaxing these constraints in the objective function is

$$\min \sum_{k \in K} \sum_{(i,j) \in A} (c_{ij} - \lambda_i)\, x_{ij}^k + \sum_{i \in N} \lambda_i, \tag{24}$$

subject to constraints (13)–(19).

This subproblem does not possess the integrality property. It does, however, decompose into m disjoint elementary shortest-path problems with capacity and time window constraints. When all vehicles are identical, a single problem can be solved to compute the lower bound. The Lagrangian dual, i.e., the problem of finding optimal multipliers that maximize $L(\boldsymbol{\lambda})$, is a concave nondifferentiable maximization problem. Using subgradient and bundle methods, Kohl and Madsen (1997) were able to solve some instances with up to 100 customers. They reported optimal solutions to each of the 27 clustered and short-horizon Solomon instances.

Kallehauge et al. (2006) have developed a stabilized cutting-plane algorithm to solve the Lagrangian dual. Cutting planes are generated by solving the Lagrangian subproblem and are introduced in a master problem which imposes bounds (i.e., a trust region) on the dual variables to ensure the stability of their values from one iteration to the next. Optimizing the relaxed master problem (a maximization linear program) provides a lower bound on the value of the original problem. To obtain feasible integer solutions, the cutting-plane algorithm is embedded within a branch-and-bound algorithm and valid inequalities are introduced in the master problem. Because the relaxed master problem is stated on the dual variables, violated subtour elimination constraints and 2-path inequalities (see Section 3.2.2) are added as columns to this problem. This approach has yielded good results on the Solomon test instances and was able to solve two large instances with 400 and 1000 customers, respectively.

3.2.2 Column generation algorithms

Column generation is intimately related to constraint generation and can be seen as a special way of updating the multipliers associated with the relaxed constraints. Let Ω^k denote the set of feasible paths for vehicle $k \in K$. For each path $\omega \in \Omega^k$, let c^k_ω be the cost of this path and let θ^k_ω be a binary variable equal to 1 if and only if vehicle k uses path ω. Let also $a_{i\omega}$ be the number of times customer $i \in N$ is visited by path ω. As first suggested by Balinski and Quandt (1964), the VRPTW can be stated as follows:

$$\text{minimize} \quad \sum_{k\in K}\sum_{\omega\in\Omega^k} c^k_\omega \theta^k_\omega \tag{25}$$

subject to

$$\sum_{k\in K}\sum_{\omega\in\Omega^k} a_{i\omega}\theta^k_\omega = 1, \quad i \in N, \tag{26}$$

$$\sum_{\omega\in\Omega^k} \theta^k_\omega = 1, \quad k \in K, \tag{27}$$

$$\theta^k_\omega \in \{0, 1\}, \quad k \in K, \omega \in \Omega^k. \tag{28}$$

Because the sets Ω^k are likely to have a very large cardinality, this problem can be tackled by a branch-and-bound algorithm in which the linear relaxations are solved by column generation. At each node of the enumeration tree, a restricted column generation master problem is solved over the current set of columns. New columns of negative reduced cost are generated by solving a resource constrained shortest path problem (13)–(19) with modified arc costs reflecting the current values of the dual variables associated with the constraints of the column generation master problem. This process stops when no negative reduced cost column can be generated. Because the column generation subproblem is equivalent to the Lagrangian subproblem $L(\boldsymbol{\lambda})$, the lower bound provided by column generation is equal to the value of the Lagrangian dual. The dual of the LP relaxation of formulation (25)–(28) is, in fact, equivalent to the Lagrangian dual defined in the previous section. This formulation can also be obtained by applying the Dantzig–Wolfe decomposition principle (Dantzig and Wolfe, 1960) to the original formulation (11)–(19).

Branching must be performed at each node of the branch-and-bound tree, where the optimal solution to the linear relaxation includes fractional path variables. While it is in principle possible to branch directly on fractional θ_ω variables, this approach is difficult to implement in practice. Indeed, it is easy to set such variables equal to 1 but it is much more difficult to impose the opposite decision. In the latter case, care must be taken to ensure that the same path will not be generated more than once by the subproblem. To this purpose, one could use a modified dynamic programming algorithm to implicitly handle forbidden paths, or a p-shortest path algorithm where p is equal to the number of forbidden paths plus one. This would ensure the generation of at least one valid path of negative reduced cost whenever one exists. A more convenient branching scheme consists of making decisions on the original arc flow variables x_{ij}^k or on sums of these variables. For example, binary decisions can be made on the following sum of variables:

$$\sum_{j\in N'} \sum_{k\in K'} x_{ij}^k,$$

where $i \in N$, $N' \subseteq \delta^+(i)$, and $K' \subseteq K$. Forcing this sum to be equal to 1 requires that some vertex in subset N' be visited immediately after i by some vehicle. If $|N'| = 1$, then the corresponding vertex must be visited after i by some vehicle. If $|K'| = 1$, then vertex i is implicitly assigned to vehicle k. The special case $|N'| = 1$ and $|K'| = 1$ is equivalent to forcing $x_{ij}^k = 1$ for some given j and k. It is worth pointing out that all such decisions can be handled directly at the subproblem level through the simple elimination of arcs in the networks.

Column generation was successfully applied to the VRPTW by Desrochers et al. (1992) and by Kohl et al. (1999). The latter authors also used valid inequalities to strengthen the bounds obtained by column generation. More specifically, let

$$x(S) = \sum_{k\in K} \sum_{i\in V\setminus S} \sum_{j\in S} x_{ij}^k$$

denote the flow into set $S \subseteq N$ and denote by $\kappa(S)$ the minimum number of vehicles needed to serve all customers in S. Then the constraint

$$x(S) \geqslant \kappa(S) \tag{29}$$

is a valid inequality for the VRPTW and is called a κ-path inequality. Computing $\kappa(S)$ is a difficult problem which is equivalent to solving the VRPTW on a subset of vertices with the objective of minimizing the number of vehicles used. Kohl et al. (1999) have, in fact, restricted their attention to the case

$\kappa = 2$. Determining whether $\kappa(S) = 1$ for a particular subset S can be achieved by checking that the capacity of a single vehicle is sufficient and the corresponding TSPTW is feasible. The latter problem is NP-hard but can be solved relatively quickly by dynamic programming for small instances. The algorithm of Kohl et al. was capable of solving 70 of the 87 Solomon short-horizon instances to optimality. Cook and Rich (1999) have extended this approach to the case $\kappa \leqslant 6$ by using parallel computing and replacing the TSPTW feasibility problem with a VRPTW. They were thus able to solve 80 of the short-horizon instances. They also solved 30 of the 81 long-horizon instances.

While the constrained elementary shortest path problem is NP-hard, the relaxation obtained by allowing cycles can be solved by a pseudopolynomial labeling algorithm (see, e.g., Desrochers and Soumis, 1988). Because of time windows and capacity constraints, these cycles will nevertheless be of finite length. This relaxation will of course weaken the value of the lower bound, but cycle elimination procedures can be used to circumvent this difficulty. A procedure for eliminating 2-cycles (i.e., cycles of the form (i, j, i)) was first proposed by Houck et al. (1980). More recently, Irnich and Villeneuve (2003) developed an efficient approach to forbid cycles of length greater than 2. Experiments performed by the authors show that k-cycle elimination with $k \geqslant 3$ can substantially improve the lower bounds. Embedding this technique within column generation enabled the exact solution of 15 previously unsolved instances of the Solomon benchmark set.

Recently, Chabrier (2006) proposed a modified labeling algorithm to handle the constrained elementary shortest path problem and thus obtain improved lower bounds. In this algorithm, both exact and heuristic dominance rules are considered. Whenever the heuristic approach cannot find a path of negative reduced cost, the exact but slower implementation is used. This approach has allowed the author to find the optimal solution to 17 previously unsolved long-horizon instances from the Solomon benchmark set.

Promising results were also reported by Danna and Le Pape (2003) who developed a cooperation scheme between column generation and local search applied to the VRPTW. During the branch-and-price process, local search is regularly applied from the best known integer solution. This often results in an improved upper bound that can then be used to prune nodes in the enumeration tree. Furthermore, columns associated with solutions identified during local search can be fed into the restricted master problem. The branch-and-price algorithm thus benefits from local search by being provided at an early stage with high quality upper bounds, resulting in a smaller search tree. In turn, local search benefits from branch-and-price by working with a variety of different initial solutions, resulting in an effective form of diversification.

3.2.3 A branch-and-cut algorithm

A branch-and-cut algorithm for the VRPTW was developed by Bard et al. (2002). As in most such algorithms for the VRP, the problem is formulated using two-index variables x_{ij} equal to 1 if and only if a vehicle travels directly from vertex i to vertex j. The algorithm incorporates five types of inequalities: subtour elimination constraints, capacity constraints, comb inequalities, incompatible pair inequalities, and incompatible path inequalities. At each node of the search tree an upper bound is computed by means of the Greedy Randomized Adaptive Search Procedure (GRASP) described by Kontoravdis and Bard (1995).

Incompatible pair inequalities rely on the existence of vertex pairs that cannot belong to the same vehicle route. If i and j denote two incompatible vertices and $\mathcal{P} = (i, h_1, \dots, h_{|\mathcal{P}|-2}, j)$ is a path, then the following inequality is valid:

$$x_{i_1,h_1} + x_{h_1,i} + \cdots + x_{h_{|\mathcal{P}|-2},j} + x_{j,h_{|\mathcal{P}|-2}} \leqslant |\mathcal{P}| - 2. \tag{30}$$

Incompatible path inequalities are similar to infeasible pair inequalities but take arc orientations into account. If i and j are two vertices such that i cannot precede j in a feasible vehicle route then the following inequality is valid for any path $\mathcal{P}$ between i and j:

$$x_{i,h_1} + x_{h_1,h_2} + \cdots + x_{h_{|\mathcal{P}|-2},j} \leqslant |\mathcal{P}| - 2. \tag{31}$$

The authors present four separation heuristics to identify violated capacity constraints. The first is based on the computation of minimum cuts in G. The second applies a graph shrinking heuristic similar to that proposed by Araque et al. (1994) for the VRP. The third consists of identifying connected components in G that do not contain the depot. Finally, the fourth is a heuristic proposed by Kohl et al. (1999) to identify violated 2-path inequalities. Heuristic separation algorithms are also described for the identification of violated comb inequalities, incompatible path inequalities, and incompatible pair inequalities. The branch-and-cut algorithm of Bard, Kontoravdis, and Yu has obtained good results on the Solomon test instances: all 50-customer instances and several 100-customer instances were solved optimally.

3.3 *Heuristics for the VRPTW*

Because of the difficulty of the VRPTW and its high practical relevance, there is a genuine need to develop fast algorithms capable of producing good quality solutions in short computing times. Heuristics can also be used to provide upper bounds for the exact algorithms described in the previous section. This section describes the three main classes of heuristics for the VRPTW: construction heuristics, improvement heuristics, and metaheuristics.

3.3.1 *Construction heuristics*

Route construction algorithms work by inserting customers one at a time into partial routes until a feasible solution is obtained (see Section 2.3.1). Routes can either be constructed sequentially or in parallel. Construction algorithms are mainly distinguished by the order in which customers are selected and by the method used to determine where a customer should be inserted.

Several sequential insertion heuristics for the VRPTW were proposed by Solomon (1987). Among these heuristics, the most efficient, called I1, consists of first selecting the farthest customer from the depot as a seed customer. The remaining customers are then inserted one at a time into the current route by selecting at each iteration the customer that maximizes a saving measure, taking into account the distance from the depot and the cost of insertion in the current route. The customer is then inserted in the position minimizing a weighted combination of extra distance and extra time required to visit the customer. The process is repeated until all customers have been inserted or it is no longer possible to insert additional customers without violating either the capacity or time window constraints. At this point a new route is initialized by selecting a new seed customer and the process repeats itself until no customers remain.

A parallel version of this heuristic was later developed by Potvin and Rousseau (1993) who proposed a generalized regret measure to select the next customer for insertion. This measure reflects the cost increase likely to result if a customer is not assigned to the route minimizing the insertion cost. Further improvements to the sequential heuristic of Solomon (1987) were also described by Ioannou et al. (2001) who proposed modifying the criteria for customer selection and insertion to take into account the impact of the insertion on all routed and unrouted customers.

3.3.2 Improvement heuristics

Improvement heuristics iteratively improve an initial feasible solution by performing exchanges while maintaining feasibility. The process normally stops when no further exchange can be made without deteriorating the solution. Improvement heuristics are mainly characterized by the type of exchanges considered at each iteration. These define the neighborhood of a solution, i.e., the set of solutions reachable from the current solution by performing a single exchange.

The first improvement heuristics for the VRPTW (see, e.g., Russell, 1977; Baker and Schaffer, 1986) were adaptations of the 2-opt (Croes, 1958), 3-opt (Lin, 1965), and Or-opt (Or, 1976) edge exchange mechanisms originally introduced for the TSP. Because of time windows, checking whether a given exchange maintains feasibility of the solution can be rather time consuming. Starting with the work of Savelsbergh (1985), several attempts have been made to develop efficient implementations of neighborhood evaluation procedures for λ-exchanges (see also Solomon et al., 1988; Savelsbergh, 1990, 1992). A comparison of 2-opt, 3-opt, and Or-opt exchange heuristics for the VRPTW was performed by Potvin and Rousseau (1995) who also introduced a new exchange, called 2-opt*, a special case of 2-opt that maintains the orientation of the subroutes involved in the exchange. This is accomplished by removing the last n_1 customers from a route k_1, inserting them after the first n_2 customers of a route k_2, and reconnecting the initial part of route k_1 with the terminal part of route k_2. Another exchange mechanism was described by Thompson and

Psaraftis (1993) who proposed transferring sets of customers in a cyclic fashion between routes.

Several attempts have also been made to integrate construction and improvement heuristics. Russell (1995) developed a procedure that embeds route improvement within the solution construction process. More precisely, customers can be switched between routes, and routes can be eliminated during the construction of the solution which is performed by a procedure similar to that of Potvin and Rousseau (1993). More recently, Cordone and Wolfler Calvo (2001) have proposed a composite heuristic in which a set of initial solutions is first constructed by means of Solomon's I1 insertion heuristic and an improvement procedure is then applied to each of them. This procedure applies 2-opt and 3-opt exchanges and attempts to reduce the number of routes by relocating customers. To escape from local optima, the heuristic alternates between an objective minimizing total distance and an objective minimizing total route duration (the primary objective being in both cases to minimize the number of routes). Several deterministic local search heuristics were also proposed by Bräysy (2002), based on a new three-phase approach. In a first phase, an initial solution is created with one of two proposed route construction heuristics (a cheapest insertion-based heuristic with periodic route improvements and a parallel savings heuristic). The second phase attempts to reduce the number of routes by applying a local search operator based on ejection chains (see, e.g., Glover, 1992). Finally, the third phase applies Or-opt exchanges to reduce the total length of the routes.

3.3.3 Metaheuristics

Most of the recent research on approximate algorithms for the VRPTW has concentrated on the development of metaheuristics. Unlike classical improvement methods, metaheuristics usually incorporate mechanisms to continue the exploration of the search space after a local minimum is encountered.

Tabu search heuristics. Some of the first applications of tabu search to the VRPTW can be attributed to Semet and Taillard (1993) and to Potvin et al. (1996) who combined Solomon's insertion heuristics with improvement schemes based on vertex and chain exchange procedures.

A more sophisticated algorithm was later developed by Taillard et al. (1997) for the VRP with soft time windows in which vehicles are allowed to arrive late at customer locations but time window violations are penalized in the objective function. This heuristic relies on the concept of adaptive memory introduced by Rochat and Taillard (1995) and on the decomposition and reconstruction procedure developed by Taillard (1993) for the classical VRP. An adaptive memory is a pool of routes extracted from the best solutions found during the search. This memory is first initialized with routes produced by a randomized insertion heuristic. At each iteration of the metaheuristic, a solution is constructed from the routes belonging to the adaptive memory and is improved through tabu search. The routes of the resulting solution are then stored in the adaptive memory if this solution improves upon the worst solution already stored. The tabu search heuristic uses an exchange operator, called CROSS exchange, which swaps sequences of consecutive customers between two routes. Individual routes are also optimized by removing two edges from a route and moving the segment between these two edges to another location within the route. A parallel computing implementation of this approach is described in Badeau et al. (1997).

A metaheuristic embedding reactive tabu search (see, e.g., Battiti and Tecchiolli, 1994) within the parallel construction heuristic of Russell (1995) was developed by Chiang and Russell (1997). In this implementation, the tabu list length is increased if identical solutions occur too frequently and is decreased if no feasible solution can be found. Using a variety of customer ordering rules and criteria for measuring the best insertion points, the metaheuristic first constructs six different initial solutions by gradually inserting customers and repeatedly applying tabu search to the partial solutions. The best solution obtained after this step is further improved through tabu search. Exchanges are performed by using some of the λ-interchanges of Osman (1993): switch a customer from one route to another and swap two customers belonging to different routes.

More recently, a tabu search heuristic was developed by Cordeau et al. (2001) for the VRPTW and two of its generalizations: the periodic VRPTW and the multidepot VRPTW (see also Cordeau et al., 1997). In this heuristic, an initial solution is obtained by means of a modified sweep heuristic. Infeasible solutions are allowed during the search and violations of capacity, duration or time window constraints are penalized in the objective function through dynamically updated penalty factors. At each iteration of the tabu search, a customer is removed from its current route and inserted into a different route by using a least cost insertion criterion. A continuous diversification mechanism that penalizes frequently made exchanges is used to drive the search process away from local optima. Finally, a post-optimizer based on a specialized TSPTW heuristic (Gendreau et al., 1998) is applied to individual routes. An improvement to this heuristic for the handling of route duration constraints was recently described by Cordeau et al. (2004). The heuristic was also extended by Cordeau and Laporte (2001) to handle heterogeneous vehicles. Other tabu search algorithms for the VRPTW were proposed by Brandão (1998), Schulze and Fahle (1999), and Lau et al. (2003).

Genetic algorithms. Homberger and Gehring (1999) have described two evolution strategies for the VRPTW. Both are based on the (μ, λ) strategy: starting from a population with μ individuals, subsets of individuals are randomly selected and recombined to yield a total of $\lambda > \mu$ offspring. Each offspring is then subjected to a mutation operator, and the μ fittest are selected to form the new population. In the first method, new individuals are generated directly through mutations and no recombination takes place. Mutations are obtained by performing one or several moves from the 2-opt, Or-opt, and

1-interchange families. In the second method, offspring are generated through a two-step recombination procedure in which three individuals are involved. In both methods, the fitness of an individual depends first on the number of vehicles used, and second on the total distance traveled. Gehring and Homberger (2002) later proposed a two-phase metaheuristic in which the first phase minimizes the number of vehicles through an evolution strategy, while the second one minimizes the total distance through tabu search. A parallelization strategy is also used to run several concurrent searches of the solution space with differently configured metaheuristics cooperating through the exchange of solutions.

Berger et al. (2003) have developed a genetic algorithm that concurrently evolves two distinct populations pursuing different objectives under partial constraint relaxation. The first population aims to minimize the total distance traveled while the second one focuses on minimizing the violations of the time window constraints. The maximum number of vehicles imposed in the first population is equal to $k_{\min}$ whereas the second population is allowed only $k_{\min} - 1$ vehicles, where $k_{\min}$ refers to the number of routes in the best known feasible solution. Whenever a new feasible solution emerges from the second population, the first population is replaced with the second and the value of $k_{\min}$ is updated accordingly. Two recombination operators and five mutation operators are used to evolve the populations. This approach has proved to be rather efficient in minimizing the number of vehicles used.

More recently, Mester and Bräysy (2005) have developed an iterative metaheuristic that combines guided local search and evolution strategies. An initial solution is first created by an insertion heuristic. This solution is then improved by the application of a two-stage procedure. The first stage consists of a guided local search procedure in which 2-opt* and Or-opt exchanges are performed together with 1-interchanges. This local search is guided by penalizing long arcs appearing often in local minima. The second stage iteratively removes a selected set of customers from the current solution and reinserts the removed customers at minimum cost. These two stages are themselves repeated iteratively until no further improvement can be obtained. Very good results are reported by the authors on large-scale instances. According to Bräysy and Gendreau (2005b), the three approaches just described seem to produce the best results among genetic algorithms. Other such algorithms have also been proposed by a number of researchers including Potvin and Bengio (1996), Thangiah and Petrovic (1998), and Tan et al. (2001).

Other metaheuristics. Kontoravdis and Bard (1995) have described a two-phase GRASP for the VRPTW. A number of routes are first initialized by selecting seed customers. The remaining customers are then gradually inserted in the routes by using a randomized least insertion cost procedure. During this process, periodic attempts are made to improve the routes by local search. In this phase certain routes may be eliminated by means of a deterministic procedure that attempts to relocate the customers to a different route. To estimate the required number of routes, the authors have proposed three lower bounds for fleet size. Two are based on bin packing structures generated by the capacity or time window constraints. The other is derived from the associated graph created by pairs of customers having incompatible demands or time windows.

A guided local search algorithm for the VRPTW was introduced by Kilby et al. (1998). In guided local search, the objective function is augmented with a penalty term reflecting the proximity of the current solution value to that of previously encountered local minima. The method is used to drive a local search heuristic that modifies the current solution by performing one of four moves: 2-opt exchanges within a route, switching a customer from one route to another, exchanging customers belonging to two different routes, and swapping the ends of two routes. All customers are first assigned to a virtual vehicle and the routes for the actual vehicles are left empty. Because a penalty is associated with not visiting a customer, a feasible solution will be constructed in the process of minimizing cost. The local search algorithm starts from this solution and performs a series of exchanges until a local minimum is reached. The objective function is then modified by adding a term penalizing the presence of the arcs used in this solution. The search iterates by finding new local minima and accumulating penalties until a stopping criterion is met. This approach was later coupled with tabu search and embedded within a constraint programming framework by De Backer et al. (2000).

Gambardella et al. (1999) have developed an ant colony optimization algorithm for the VRPTW which associates an attractiveness measure to the arcs. Artificial ants represent parallel processes whole role is to construct feasible solutions. To deal with the hierarchical objective of first minimizing the number of vehicles and then minimizing distance, two ant colonies are used, each dedicated to the optimization of a different objective. These colonies cooperate by exchanging information through pheromone updating. Whenever a feasible solution with a smaller number of vehicles is found, both colonies are reactivated with the reduced number of vehicles.

Bent and Van Hentenryck (2004) have described a two-stage hybrid algorithm that first minimizes the number of routes by simulated annealing and then minimizes total distance traveled by using a large neighborhood search (Shaw, 1998) which may relocate a large number of customers. The first stage uses a lexicographic evaluation function to minimize the number of routes, maximize the sum of the squares of the route sizes, and minimize the minimal delay (a measure of time window tightness) of the solution. The neighborhood used in this stage consists of 2-opt, Or-opt, relocating, exchange, and crossover moves. In the second stage, subsets of customers are removed from their current route and reinserted in possibly different routes. Customers selected for removal are randomly chosen but the algorithm favors customers that are geographically close to each other and belong to different routes. A branch-and-bound algorithm is then used to reinsert these customers.

A four-phase metaheuristic based on a modification of the variable neighborhood search was described by Bräysy (2003). In the first phase, an initial solution is created by using route construction heuristics. During this process, the partial routes are periodically reoptimized through Or-opt exchanges. In the second phase, an attempt is made to reduce the number of routes by applying a route elimination operator based on ejection chains. In the third phase, four local search procedures embedded within a variable neighborhood search (see, e.g., Mladenović and Hansen, 1997) are applied to reduce the total distance traveled. These procedures are based on modifications to the CROSS exchanges of Taillard et al. (1997) and cheapest insertion heuristics. In the fourth phase, a modified objective function considering waiting time is used by the local search operators in the hope of further improving the solution.

More recently, a local search algorithm with restarts was also proposed by Li and Lim (2003). This algorithm first constructs an initial solution by using an insertion heuristic. Local search is then performed from this solution using three exchange operators that move segments of customers either between routes or within the same route. Whenever a local minimum is reached, multiple restarts are performed starting from the best known solution, and a tabu list is used to prevent cycling.

A large number of other metaheuristics based on various paradigms have been described in recent years. For additional references on approximate algorithms for the VRPTW as well as detailed computational experiments, the reader is referred to recent surveys by Bräysy and Gendreau (2005a, 2005b).

4 The inventory routing problem

The Inventory Routing Problem (IRP) is an important extension of the VRP which integrates routing decisions with inventory control. The problem arises in environments where Vendor Managed Inventory (VMI) resupply policies are employed. These policies allow a vendor to choose the timing and size of deliveries. In exchange for this freedom, the vendor agrees to ensure that its customers do not run out of product. In a more traditional relationship, where customers call in their orders, large inefficiencies can occur due to the timing of customers' orders (resulting in high inventory and distribution costs). Realizing the cost savings opportunities of vendor managed inventory policies, however, is not a simple task, particularly with a large number and variety of customers. The inventory routing problem achieves this goal by determining a distribution strategy that minimizes *long term* distribution costs. This description of the inventory routing problem focuses primarily on distribution. Inventory control is restricted to ensuring that no stockouts occur at the customers. Inventory control takes a more prominent role when inventory holding costs are considered. In the inventory control literature, the resulting environment is usually referred to as a *one warehouse multiretailer system*.

Inventory routing problems are very different from VRPs. Vehicle routing problems occur when customers place orders and the vendor, on any given day, assigns the orders for that day to routes for vehicles. In IRPs, the delivery company, not the customer, decides how much to deliver to which customers each day. There are no customer orders. Instead, the delivery company operates under the restriction that its customers are not allowed to run out of product. Another key difference is the planning horizon. Vehicle routing problems typically deal with a single day, the only requirement being that all orders have to be delivered by the end of the day. Inventory routing problems are defined on a longer horizon. Each day the delivery company makes decisions about which customers to visit and how much to deliver to each of them, while keeping in mind that decisions made today have an impact on what has to be done in the future. The objective is to minimize the total cost over the planning horizon while ensuring that no customer runs out of product.

4.1 Definition of the IRP

The deterministic IRP is concerned with the repeated distribution of a single product from a single facility, to a set of n customers over a planning horizon of length T, possibly infinity. Customer i consumes the product at a rate u_i (say volume per day) and can maintain a local inventory of product of up to a maximum of C_i. The inventory at customer i is I_i^0 at time 0. A fleet of m homogeneous vehicles, with capacity D, is available for the distribution of the product. If a quantity d_i is delivered at customer i, the vendor earns a reward equal to $r_i d_i$. It takes a vehicle a time t_{ij} to traverse arc (i, j) of the distribution network and a cost c_{ij} is incurred when doing so. The objective is to maximize the profit (revenue minus cost) over the planning horizon, without causing stockouts at any of the customers. (Note that because product usage is assumed to be deterministic and no stockouts are allowed, long run revenues are fixed and the key is to reduce delivery costs.) A dispatcher has to decide when to serve a customer, how much to deliver, and which delivery routes to use to serve customers.

In the stochastic IRP customer demands are defined at discrete time instants t by means of random variables. Let $U_t = (U_{1t}, \dots, U_{nt})$ denote the vector of random customer demands at time t. Customer demands on different days are independent random vectors with a joint probability distribution F that does not change with time; that is, $U_0, U_1, \dots$ is an independent and identically distributed sequence, and F is the probability distribution of each U_t. The probability distribution F is known to the decision maker. The vendor can measure the inventory level X_{it} of each customer i at any time t. At each time instant t, the vendor makes a decision that controls the routing of vehicles and the replenishment of customer inventories. Because demand is uncertain, there is often a positive probability that a customer will run out of stock, and thus shortages cannot always be prevented. Shortages result in a penalty $p_i s_i$ if the unsatisfied demand on day t at customer i is s_i. Unsatisfied demand is treated as lost demand. The objective is to construct a distribution policy maximizing the expected discounted profit over an infinite time horizon.

4.2 Motivating example

To illustrate the difficulty of inventory routing problems, we reproduce a small deterministic example introduced by Fisher et al. (1982) and Bell et al. (1983). The relevant optimal tour costs can be derived from the network shown in Figure 1, e.g., the optimal tour costs for visiting customers 1 and 2, denoted by $C_{1,2}$, is equal to \$210. The vehicle capacity is 5000 gallons and customer tank capacity and usage data, in gallons, are as follows:

Customer i	d_i	u_i
1	5000	1000
2	3000	3000
3	2000	2000
4	4000	1500

A simple schedule jointly replenishes customers 1 and 2 as well as customers 3 and 4 on a daily basis. This schedule is natural because customers 1 and 2 (3 and 4, respectively) are near each other. Each customer i receives a quantity equal to its daily consumption u_i. The long-run average cost of this schedule is 420 miles per day. An improved schedule consists of a cycle that repeats itself every two days. On the first day, one trip replenishes 3000 gallons to customer 2 and 2000 gallons to customer 3, at a cost of 340 miles. On the second day, two trips are made. The first trip replenishes 2000 gallons to customer 1 and 3000 gallons to customer 2. The second trip replenishes 2000 gallons to

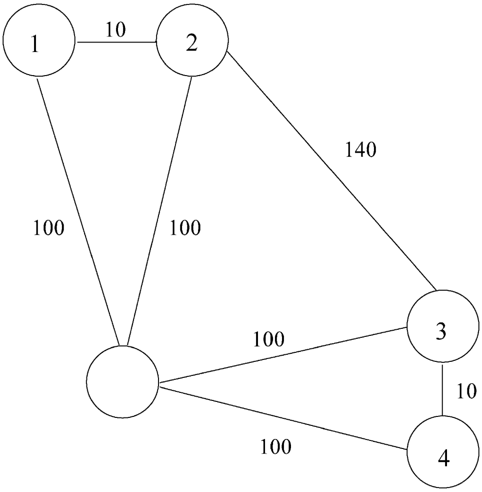

Fig. 1. A four-customer example with distances shown on edges.

customer 3 and 3000 gallons to customer 4. Each trip costs 210 miles. The average cost of this schedule is 380 miles per day, which is nearly 10% lower than the first schedule.

4.3 Observations on the IRP

Before describing solution approaches, we present some general observations concerning inventory routing problems and some common elements found in most solution approaches.

The IRP is a long-term dynamic control problem which is extremely difficult to solve. Therefore, most of the available algorithms solve only a short-term planning problem. In early publications, it was often just a single day but later, short-term was expanded to a few days. Two key issues need to be resolved with such approaches: how to model the long-term effect of short-term decisions, and which customers to consider in the short-term planning period. A short-term approach that only minimizes costs has the tendency to defer as many deliveries as possible to future planning periods, which may lead to an undesirable situation in the future. Therefore, a proper incorporation of the long-term objective into the short-term planning problem is essential. The long-term effect of short-term decisions needs to capture the costs and benefits of delivering to a customer earlier than necessary. This usually means delivering less and may lead to higher future distribution costs, but reduces the risk of a stockout and may thus reduce future shortage costs. Decisions regarding which customers need to be considered in the short-term planning period are usually guided by some measure of the urgency to make a delivery to a customer and the quantity that can be delivered. Usually, it is assumed that customers considered in the short-term planning period may actually be visited, but the decision whether or not to actually visit them still has to be made.

When the short-term planning problem consists of a single day, the problem can be viewed as an extension of the VRP and solution techniques for the VRP can be adapted. For example, Campbell and Savelsbergh (2004c) have discussed efficient implementations of insertion heuristics to handle situations where the delivery amount has to lie between a lower and an upper bound, as opposed to being fixed. In related work, Campbell and Savelsbergh (2004b) have studied the problem of determining an optimal delivery schedule for a route, i.e., given a sequence of customer visits, determine the timing of the visits so as to maximize the total amount of the product delivered on the route. Because single day approaches usually base decisions on the latest inventory measurement and a predicted usage for that day, they avoid the difficulty of forecasting long-term usage, which makes the problem much simpler.

4.4 Single customer analysis

It is insightful to analyze the “simple” situation in which there is only a single customer. The results of this type of analysis can be used effectively to guide decisions on which customers to consider in a short term planning horizon. The material presented in this subsection is primarily based on Jaillet et al. (2002), although much of it dates back to the work of Dror and Ball (1987). We first consider the deterministic case. For ease of notation, let the usage rate of the customer be u, the storage capacity of the customer be C, the initial inventory level be I^0, the delivery cost to the customer be c, and the vehicle capacity be Q. It is easy to see that an optimal policy is to fill up the storage space precisely at the time when it becomes empty. Therefore the cost v_T for a planning period of length T is

$$v_T = \max\left\{0, \left\lceil \frac{Tu - I^0}{\min\{C, Q\}} \right\rceil\right\} c.$$

Now consider the stochastic case in which one decides daily whether to make a delivery to the customer or not. The demand U between consecutive decision points, i.e., the demand per day, is a random variable with known probability distribution and finite mean. Assuming that the storage capacity at each customer is at least as large as the vehicle capacity and the vendor can only monitor the inventory in the storage space at the time of a delivery, it can be shown that for the infinite horizon case, there exists an optimal policy that fills up the storage space at each delivery and, following any scheduled or stockout delivery, plans the next delivery d days after. The optimal replenishment interval d is a constant chosen to minimize the expected daily cost.

A d-day policy makes a delivery to the customer every d days and delivers as much as possible, unless a stockout occurs earlier. In such a case, the vehicle is sent right away, which generates a cost S. It is assumed that deliveries are instantaneous, so that no additional stockout penalties are incurred. Furthermore, assume that initially the storage space is full. Let p_j be the probability that a stockout first occurs on day j $(1 \leqslant j \leqslant d-1)$. Then $p = p_1 + p_2 + \cdots + p_{d-1}$ is the probability that there is a stockout in period $[1, \ldots, d-1]$. Furthermore, let $v_T(d)$ be the expected total cost of this policy over a planning period of length T. We now have for $d > T$

$$v_T(d) = \sum_{1 \leqslant j \leqslant T} p_j \big(v_{T-j}(d) + S\big)$$

and for $d \leqslant T$

$$v_T(d) = \sum_{1 \leqslant j \leqslant d-1} p_j \big(v_{T-j}(d) + S\big) + (1-p)\big(v_{T-d}(d) + c\big).$$

As a consequence, the expected total cost of filling up a customer's tank every d days over a T-day period $(T \geqslant d)$ is given by

$$v_T(d) = \alpha(d) + \beta(d)T + f(T, d),$$

where $\alpha(d)$ is a constant depending only on d, $f(T, d)$ is a function that tends to zero exponentially fast as T tends to infinity, and

$$\beta(d) = \frac{pS + (1 - p)c}{\sum_{1 \leqslant j \leqslant d} j p_j},$$

with $p_d = 1 - p$. The value $\beta(d)$ is the long-run average cost per day. To determine the best policy in this class, we need to minimize $v_T(d)$ which for large T means finding a value of d minimizing $\beta(d)$.

4.5 The two-customer IRP

When more than one customer is served, the problem becomes significantly harder. Not only is it necessary to decide which customers to visit next, but one must also determine how to combine them into vehicle tours, and how much to deliver to each of them. Even if there are only two customers, these decisions may not be easy. The material in the remainder of this section is primarily based on Campbell et al. (1998).

If the two customers are visited together, it is intuitively clear that given the amount delivered at the first customer, it is optimal to deliver as much as possible at the second one (determined by the remaining amount in the vehicle, and the remaining capacity at the second customer). Thus the problem of deciding how much to deliver to each customer involves a single decision. However, making that decision may not be easy, as the following two-customer stochastic IRP example shows.

Assume the product is delivered and consumed in discrete units and that each customer has a storage capacity of 20 units. The daily demands of the customers are independent and identically distributed (across customers as well as across time), with $P(U = 0) = 0.4$ and $P(U = 10) = 0.6$. The shortage penalty is $s_1 = 1000$ per unit at customer 1 and $s_2 = 1005$ per unit at customer 2. The vehicle capacity is 10 units. At the beginning of each day the inventory at the two customers is measured, and the decision maker determines how much to deliver to each customer. There are three possible vehicle tours, namely tours exclusively to customers 1 and 2, of cost 120 each, and a tour to both customers 1 and 2, of cost 180. Only one vehicle tour can be completed per day. This situation can be modeled as an infinite horizon Markov decision process, with the objective of minimizing the expected total discounted cost. Because of the small size of the state space, it is possible to compute the optimal expected value and an optimal policy.

Figure 2 shows the expected value (total discounted cost) as a function of the amount delivered at customer 1 (and therefore also at customer 2), when the inventory at each customer is 7, and both customers are to be visited in the next vehicle tour (which is the optimal decision in the given state). The figure shows that the objective function is not unimodal, with a local minimum at 3, and a global minimum at 7. Consequently, deciding just how much to deliver to each customer may require solving a nonlinear optimization problem with

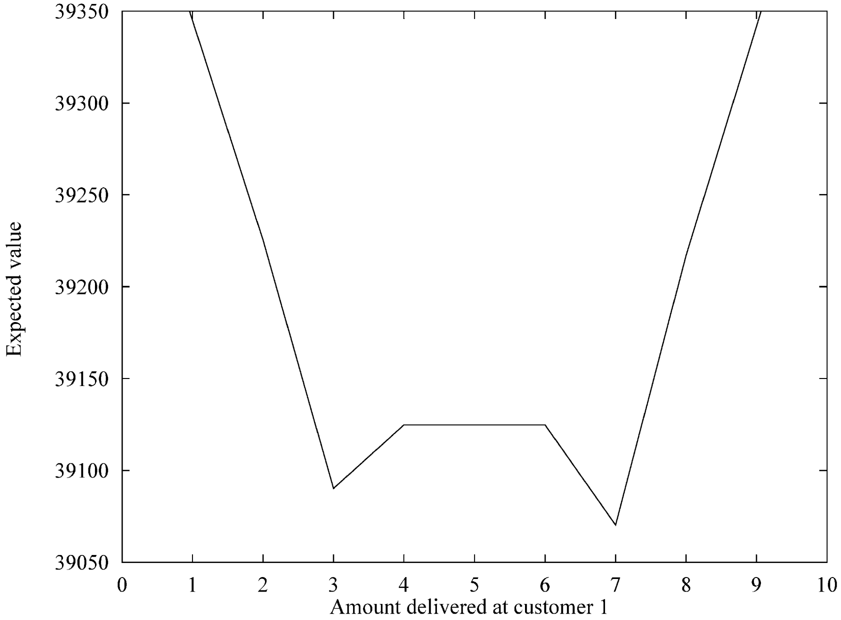

Fig. 2. Nonunimodal objective function for determining the optimal delivery quantity.

a nonunimodal objective function. This is a hard problem for which available search methods may not converge to an optimal solution.

4.6 Literature review on the IRP

Rather than providing a comprehensive review of the IRP literature, we discuss several research streams representing a variety of solution approaches that have been proposed and investigated. We encourage the reader to examine the referenced papers for more elaborate and precise coverage.

A first stream of research uses time-discretized integer programming models to determine the set of customers to be visited in a short-term planning horizon as well as the amount of product to deliver to them. In order to accurately reflect costs and time related aspects, the integer linear programs work with a set of potential delivery routes. Fisher et al. (1982) and Bell et al. (1983) pioneered this approach when they studied the IRP at Air Products, a producer of industrial gases. Their formulation determines the delivery volumes to customers, the assignment of customers to routes, the assignment of vehicle to routes, and the start times of routes. The core structure of their model is presented below, where the variable x_{irtv} represents the amount of product delivered to customer $i \in N$ on route $r \in R$ starting at time $t \in T$, the variable y_{rt} is 1 if route r starts at time t and 0 otherwise, S_r the set of customers visited on route r, p_i the value of delivering a unit of product to customer i, F_r the fixed cost of executing route r, $\underline{q}_{it}$ a lower bound on the cumulative amount delivered to customer i by time t, and $\bar{q}_{it}$ an upper bound on the cumulative amount that can be delivered to customer i by time t:

$$\begin{aligned}
\text{maximize} \quad & \sum_{r\in R}\sum_{t\in T}\Bigl(\sum_{i\in S_r} p_i x_{irt} - F_r y_{rt}\Bigr)\\
\text{subject to} \quad & \\
& \underline{q}_{it} \leqslant \sum_{r\in R}\sum_{s\leqslant t} x_{irs} \leqslant \bar{q}_{it}, \quad i \in N, t \in T,\\
& \sum_{i\in S_r} x_{irt} \leqslant Q y_{rt}, \quad r \in R, t \in T,\\
& x_{irt} \geqslant 0, \quad y_{rt} \in \{0, 1\}.
\end{aligned}$$

In the model, the per unit value of a delivery to a customer is used to represent the effect of decisions on events occurring beyond the planning horizon of the model. In the short-term planning period considered by the model, there is considerable discretion in the amount of product to deliver. In the long run this amount is determined by customer usage. Hence, each unit scheduled for delivery to a customer within the planning horizon reduces the amount to be delivered in the future. This is accounted for by setting the unit value to an estimate of the cost of delivering to a customer at a point in time outside the planning horizon of the model. Furthermore, rather than explicitly incorporating customer usage rates into the model, lower and upper bounds on the cumulative amount to be delivered to each customer in each time period in the planning horizon are used. It is simple, of course, to convert customer usage rates into bounds, i.e., $\underline{q}_{it} = \max\{0, tu_i - I_i^0\}$ and $\bar{q}_{it} = tu_i + C_i - I_i^0$. Lagrangian relaxation was a central tool in developing an effective heuristic for solving the integer program. The size of the integer programs to be solved depends on the chosen time discretization as well as on the size of set of routes.

Campbell and Savelsbergh (2004a) use an integer linear program with a similar structure to determine which customers to visit in the next few days (even though the integer program covers several weeks) and to *suggest* quantities and delivery times to these customers. However, then, in a second phase, they use modified insertion heuristics to determine the actual delivery routes and quantities. The advantage of such a two-phase approach is that a higher degree of accuracy (in terms of timing of events) can be provided in the second phase and other practical details, such as drivers shifts, can be considered. The delivery quantities and times specified by the solution to the integer program are good from a long-term perspective; they may need to be modified somewhat to also be good from a short-term perspective. When constructing the actual delivery routes, Campbell and Savelsbergh consider delivering more to the customers than the quantity suggested by the integer program (and slightly altering the delivery time if needed) since this may result in higher vehicle utilization and thus higher revenues. As in Bard et al. (1998b) their approach is embedded into a rolling horizon framework.

A second stream of research is based on the single customer analysis presented above. This approach was pioneered by Dror et al. (1985) and Dror and Ball (1987). The optimal replenishment day t_i^* minimizing the expected total cost for customer i is used to determine the set of customers considered in a short-term planning problem for the next $\bar{t}$ days. If $t_i^* \leqslant \bar{t}$, then the customer will be included and will definitely be visited. A value c_t is computed for each day of the planning period to reflect the expected increase in future cost if the delivery is made on day t instead of t^*. If $t^* \geqslant T$, i.e., the optimal replenishment day falls outside the short-term planning period, then a future benefit g_t can be computed for making an early delivery to the customer on day t of the short-term planning period. These computed values reflect the long term effects of short term decisions. An integer linear program is subsequently solved to assign customers to a vehicle and a day, or just a day, that minimizes the sum of these costs plus the transportation costs. (It was shown by Adelman (2004) that this objective function is in fact equivalent to that used by Fisher et al. (1982).) The delivery amount to a customer on a specific day is fixed and set to the quantity needed to fill up the storage tank on that day. This leaves either TSPs or VRPs to be solved in the second stage. These ideas are extended and improved in Trudeau and Dror (1992). The most recent work along these lines is that of Bard et al. (1998a, 1998b) who work with a rolling horizon approach in which a short term planning problem is defined for a two-week period, but only the decisions for the first week are implemented. In addition, satellite facilities are considered, i.e., locations other than the depot where vehicles can be refilled.

A third stream of research focuses on the asymptotic analysis of delivery policies. Anily and Federgruen (1990, 1991, 1993) analyze fixed partition policies for the IRP with an unlimited number of vehicles. Customers within the same partition are divided into regions so as to make the demand of each region roughly equal to a vehicle load. A customer may appear in more than one region, but then a certain percent of his demand is allocated to each region. When one customer in a region is also visited, all other customers in that region are also visited. The authors determine lower and upper bounds on the minimum long-run average cost over all fixed partition policies, and propose a heuristic, called modified circular regional partitioning, to choose a fixed partition. Using similar ideas, Gallego and Simchi-Levi (1990) evaluate the long-run effectiveness of direct deliveries (one customer on each route). Direct shipping is shown to be at least 94% effective over all inventory routing strategies whenever the minimal economic lot size is at least 71% of vehicle capacity. This indicates that direct shipping becomes an undesirable and costly policy when many customers require significantly less than a vehicle load, making more complicated routing policies the appropriate choice. Another adaptation of these ideas can be found in Bramel and Simchi-Levi (1995) who consider a variant of the IRP in which customers can hold an unlimited amount of inventory. To obtain a solution, they transform the problem into a Capacitated Concentrator Location Problem (CCLP), solve it, transform the solution back into a solution to the IRP, and heuristically improve it. The CCLP solution will partition the customers into disjoint sets, which in the IRP will become the fixed partitions. Chan et al. (1998) analyze zero-inventory ordering policies, in which a customer's inventory is replenished only when it has been depleted, in combination with fixed partitioning routing policies and derive asymptotic worst-case bounds on their performance. Gaur and Fisher (2004) consider an IRP with time varying demand. They propose a randomized heuristic to find a fixed partition policy with periodic deliveries. Their method was implemented for a supermarket chain.

The fourth stream of research is based on formulating the stochastic IRP as a Markov decision process and thus explicitly incorporating demand uncertainty. This approach was pioneered by Minkoff (1993) who proposed a decomposition heuristic to overcome the computational difficulties caused by large state spaces. The heuristic solves a linear program to allocate joint transportation costs to individual customers and then solves individual customer subproblems. The value functions of the subproblems are added to approximate the value function of the original problem. The main limitation of the proposed approach is that it assumes the availability of a set of delivery routes with fixed delivery quantities for the customers on a route and the dispatcher only has to decide which of the delivery routes to use at each decision point. This limitation is removed in the work of Kleywegt et al. (2002, 2004) on approximate dynamic programming approaches and in that of Adelman (2003a, 2004) on price-directed approaches. Let state $x = (x_1, x_2, \dots, x_n)$ represent the current inventory at each customer, and let $\mathcal{A}(x)$ denote the set of all feasible decisions when the process is in state x. A decision $a \in \mathcal{A}(x)$ specifies which customer inventories to replenish, how much to deliver at each customer location, and how to combine customers into vehicle routes. Let Q be the Markov transition function according to which transitions occur. Let $g(x, a)$ denote the expected single stage net reward if the process is in state x at time t and decision $a \in \mathcal{A}(x)$ is implemented. The objective is to maximize the expected total discounted value over an infinite horizon. Let $V^*(x)$ denote the optimal expected value given that the initial state is x. Then, for any state x,

$$V^*(x) = \sup_{a \in \mathcal{A}(x)} \left\{ g(x, a) + \alpha \int V^*(y) Q[\mathrm{d}y|x, a] \right\}. \tag{32}$$

A policy π^* is called optimal if $V^{\pi^*} = V^*$, where V^π represents the value function of policy π. Solving a Markov decision process involves computing the optimal value function V^* and an optimal policy π^* by solving the optimality equation (32). This requires performing the following major computational tasks:

(1) The computation of the optimal value function V^*. Most algorithms for computing V^* involve the computation of successive approximations to $V^*(x)$ for every state x. These algorithms are practical only if the state space is small.

(2) The estimation of the expected value (the integral in (32)). For the stochastic IRP, this is a high dimensional integral. Conventional numerical integration methods are not practical for the computation of such high-dimensional integrals.

(3) The maximization problem on the right-hand side of (32) has to be solved to determine the optimal decision for each state. For the stochastic IRP, this means solving a complex variant of the VRP.

Kleywegt, Nori, and Savelsbergh develop approximation methods to efficiently perform these computational tasks. Furthermore, their approach has the ability to handle a finite fleet of vehicles, whereas in other Markov decision process based approaches it is assumed that there exists an infinite fleet of vehicles. The optimal value function V^* is approximated by $\widehat{V}$ as follows. First, the stochastic IRP is decomposed into subproblems defined for specific subsets of customers. Each subproblem is also a Markov decision process. The subsets of customers do not necessarily partition the set of customers, but must cover it. The idea is to define each subproblem so that it provides an accurate representation of the overall process as experienced by the subset of customers. To do so, the parameters of each subproblem are determined by simulating the overall stochastic IRP process, and by constructing simulation estimates of subproblem parameters. Next, each subproblem is solved optimally. Finally, for any given state x, the approximate value $\widehat{V}(x)$ is determined by choosing a partition of the customers and by setting $\widehat{V}(x)$ equal to the sum of the optimal value functions of the subproblems corresponding to the partition at states corresponding to x. The partition is chosen to maximize $\widehat{V}(x)$. Randomized methods, incorporating variance reduction techniques to limit the required sample size, are used to estimate the expected value on the right-hand side of (32). Action determination involves deciding which customers to visit on a route and how much to deliver to them. This is achieved through a heuristic. An initial solution consisting of only direct delivery routes is constructed. This is followed by a local search procedure that examines the benefit of adding a customer to an existing route and modifying the delivery quantities. Using their approach Kleywegt, Nori, and Savelsbergh can solve problems involving up to 50 customers.

More recently, Adelman (2003a, 2004) proposed a price-directed operating policy based on a simple economic mechanism to determine routing and delivery decisions for a given inventory state. Suppose management specifies a value V_i for replenishing one unit of product at customer i. A dispatcher can now evaluate a feasible delivery route as follows. If a set $S = \{s_1, \dots, s_n\}$ of customers is visited, quantities $d_1, \dots, d_n$ are delivered, and a cost c_S is incurred. Then the net value of the route equals $\sum_{i \in S} V_i d_i - c_S$. The dispatcher has to choose delivery routes so as to maximize his total net value without stockouts at customers. This mechanism motivates the dispatcher to replenish a customer i whose current inventory level is low, because then d_i can be set large. When faced with the option of expanding the set S of customers to visit on a route which does not yet use the full vehicle capacity, the dispatcher will consider the incremental cost $c_{S\cup\{k\}} - c_S$ and determine if a quantity d_k can be replenished that is large enough to justify it, i.e., whether $d_k V_k - (c_{S\cup\{k\}} - c_S) > 0$ or $d_k \geqslant (c_{S\cup\{k\}} - C_S)/V_k$.

The key to success in solving management's problem is to set the V_i's in such a way that the dispatcher is motivated to (ideally) minimize the long-run time average replenishment costs. If the dispatcher's total net value is regularly positive, then his performance exceeds management's long range expectations. Management should decrease the V_i's to make them consistent with actual performance. On the other hand, if the dispatcher's total net value is regularly negative, then the V_i's impose unrealistic expectations on the dispatcher and management should increase them. Ideally, management should set the V_i's equal to the lowest achievable marginal costs.

Starting from a dynamic control model of the inventory routing problem, Adelman (2003b) derives the following nonlinear programming relaxation, which computes a long run "average" solution to the inventory routing problem. Let z_R be a decision variable representing the rate at which a subset R of customers is visited together. Furthermore, let $d_{i,R}$ for all $i \in R$ be a decision variable representing the average quantity delivered to customer i on a delivery route visiting subset R. This yields the following formulation:

$$\text{(NLP)} \quad \text{minimize} \quad \sum_{R\subseteq N} C_R z_R \tag{33}$$

subject to

$$\sum_{R\subseteq N} d_{i,R} z_R = u_i, \quad i \in N, \tag{34}$$

$$\sum_{i\in R} d_{i,R} \leqslant Q, \quad R \subseteq N, \tag{35}$$

$$d_{i,R} \leqslant C_i, \quad R \subseteq N, i \in R, \tag{36}$$

$$z_R, d_{i,R} \geqslant 0, \quad R \subseteq N, i \in R. \tag{37}$$

The objective (33) minimizes the long run average replenishment cost. Constraints (34) state that for each customer i the rate at which quantities are replenished must equal the rate at which they are consumed. Constraints (35) state that on average vehicle capacity is satisfied, and constraints (36) state that on average the quantity delivered at customer i is less than the storage capacity. Consider the following linear program

$$\text{(D)} \quad \text{maximize} \quad \sum_{i\in N} u_i V_i \tag{38}$$

subject to

$$\sum_{i\in R} d_{i,R} V_i \leqslant C_R, \quad R \subseteq N, \tag{39}$$

with decision variables V_i. Adelman shows that this semi-infinite linear program is dual to the nonlinear program in that there is no duality gap between them and a version of complementary slackness holds. In (NLP) $d_{i,R}$ is a decision variable while in (D) it is part of the input. The decision variables V_i at optimality are the marginal costs associated with satisfying constraints (34) of (NLP). This means that at optimality $u_i V_i$ is the total allocated cost rate for replenishing customer i in an optimal solution to (NLP). Each V_i can be interpreted as the payment management transfers to the dispatcher for replenishing one unit of product of customer i. Hence, the objective (38) maximizes the total transfer rate, subject to the constraint (39) that the payments can be no larger than the cost of any replenishment. NLP can be solved effectively by means of column generation techniques.

We have opted to focus on only a few research streams with an emphasis on more recent efforts. However, many other researchers have contributed to the inventory routing literature, including Federgruen and Zipkin (1984), Golden et al. (1984), Burns et al. (1985), Larson (1988), Chien et al. (1989), Webb and Larson (1995), Barnes-Schuster and Bassok (1997), Herer and Roundy (1997), Viswanathan and Mathur (1997), Christiansen and Nygreen (1998a, 1998b), Christiansen (1999), Reimann et al. (1999), Waller et al. (1999), Çetinkaya and Lee (2000), Lau et al. (2002), Bertazzi et al. (2002), Savelsbergh and Song (2005), and Song and Savelsbergh (2005).

5 Stochastic vehicle routing problems

Stochastic Vehicle Routing Problems (SVRPs) are extensions of the deterministic VRP in which some components are random. The three most common cases are:

(1) stochastic customers: customer i is present with probability p_i and absent with probability $1 - p_i$;
(2) stochastic demands (to be collected, say): the demand ξ_i of customer i is a random variable;
(3) stochastic times: the service time s_i of customer i and the travel time t_{ij} of edge (i, j) are random variables.

Because some of the data are random it is no longer required to satisfy the constraints for all realizations of the random variables, and new feasibility and optimality concepts are required. With respect to their deterministic counterparts, SVRPs are considerably more difficult to solve. Not only is the notion of a solution different, but some of the properties that were valid in a deterministic context no longer hold in the stochastic case (see, e.g., Dror et al., 1989; Gendreau et al., 1996).

Applications of SVRP arise in a number of settings such as the delivery of meals on wheels (Bartholdi et al., 1983) or of home heating oil (Dror et al., 1985), sludge disposal (Larson, 1988), forklift routing in warehouses

(Bertsimas, 1992), money collection in bank branches (Lambert et al., 1993), and general pickup and delivery operations (Hvattum et al., 2006).

Stochastic VRPs can be formulated and solved in the context of stochastic programming: a first stage or a priori solution is computed, the realizations of the random variables are then disclosed and, in a second stage, a *recourse* or corrective action is applied to the first stage solution. The recourse action usually generates a cost or a saving which may be taken into account when designing the first stage solution. To illustrate, consider a planned vehicle route in an SVRP with stochastic demands. Because demands are stochastic, the vehicle capacity may be attained or exceeded at some customer j before the route is completed. In this case several possible recourse policies are possible. For example, the vehicle could return to the depot to unload and resume collections at customer j (if the vehicle capacity was exceeded at j) or at the successor of j on the route (if the vehicle capacity was attained exactly at j). Another policy would be to plan preventive return trips to the depot in the hope of avoiding higher costs at a later stage (see, e.g., Laporte and Louveaux, 1990; Dror et al., 1993; Yang et al., 2000). A more radical policy would be to reoptimize the route segment following j upon arrival at the depot (see, e.g., Bastian and Rinnooy Kan, 1992; Secomandi, 1998; Haughton, 1998, 2000). The best choice of a recourse policy depends on the time at which information becomes available. For example, information about a customer demand may only be available upon arriving at that customer or when visiting the previous customer, thus allowing for a wider range of recourse actions, such as returning to the depot in anticipation of failure or postponing the visit of a high demand customer. An extensive discussion of recourse policies in the context of availability of information is provided in Dror et al. (1989).

There exist two main solution concepts in stochastic programming. In Chance Constrained Programming (CCP) the first stage problem is solved under the condition that the constraints are satisfied with some probability. For example, one could impose a failure threshold α, i.e., planned vehicle routes should fail with probability at most equal to α. The cost of failure is typically disregarded in this approach. Stewart and Golden (1983) have proposed the first CCP formulation for the VRP with stochastic demands. Using a three-index model they showed that probabilistic constraints could be transformed into a deterministic equivalent form. Laporte et al. (1989) later proposed a similar transformation for a two-index model. The interest of such transformations is that the chance constrained SVRP can then be solved using any of the algorithms available for the deterministic case. In Stochastic Programming with Recourse (SPR) two sets of variables are used: first-stage variables characterize the solution generated before the realization of the random variables, while second-stage variables define the recourse action. The solution cost is defined as the sum of the cost of the first-stage solution and that of the recourse action. The aim of SPR is to design a first-stage solution of least expected total cost.

Stochastic VRPs are usually modeled and solved with the framework of a priori optimization (Bertsimas et al., 1990) or as Markov decision processes (Dror et al., 1989). A priori optimization computes a first-stage solution of least expected cost under a given recourse policy. The most favored a priori optimization methodology is the *integer L-shaped method* (Laporte and Louveaux 1993, 1998) which belongs to the same class as Benders decomposition (Benders, 1962) and the L-shaped method for continuous stochastic programming (Van Slyke and Wets, 1969). While route reoptimization is preferable to a priori optimization from a solution cost point of view, it is computationally more cumbersome. In contrast, a priori optimization entails solving only one instance of an NP-hard problem and produces a more stable and predictable solution (Bertsimas et al., 1990). It is also superior to solving a deterministic VRP instance with expected demands (Louveaux, 1998).

The integer L-shaped method is essentially a variant of branch-and-cut. It operates on a current problem obtained by relaxing integrality requirements and subtour elimination constraints, and by replacing the cost of recourse $Q(x)$ of first-stage solution x by a lower bound θ on its value. Integrality and subtour elimination constraints are gradually satisfied as is commonly done in branch-and-cut algorithms for the deterministic VRP (see, e.g., Naddef and Rinaldi, 2002) while lower bounding functionals on θ, called *optimality cuts*, are introduced into the problem at integer or fractional solutions. The method assumes that a lower bound L on θ is available. In the following description x_{ij} is a binary variable equal to 1 if and only if edge (i, j) is used in the first stage solution.

Step 0. Set the iteration count $\nu := 0$ and introduce the bounding constraint $\theta \geqslant L$ into the current problem. Set the value $\bar{z}$ of the best known solution equal to ∞. At this stage, the only active node corresponds to the initial current problem.

Step 1. Select a pendent node from the list. If none exists stop.

Step 2. Set $\nu := \nu+1$ and solve the current problem. Let (x^ν, θ^ν) be an optimal solution.

Step 3. Check for any subtour elimination constraint violation. If at least one violation can be identified, introduce a suitable number of subtour elimination constraints into the current problem, and return to Step 2. Otherwise, if $cx^\nu + \theta^\nu \geqslant \bar{z}$, fathom the current node and return to Step 1.

Step 4. If the solution is not integer, branch on a fractional variable. Append the corresponding subproblems to the list of pendent nodes and return to Step 1.

Step 5. Compute $Q(x^\nu)$ and set $z^\nu := cx^\nu + Q(x^\nu)$. If $z^\nu < \bar{z}$, set $\bar{z} := z^\nu$.

Step 6. If $\theta^\nu \geqslant Q(x^\nu)$, then fathom the current node and return to Step 1. Otherwise, impose the optimality cut

$$\theta \geqslant L + \big(Q(x^\nu) - L\big)\Bigg(\sum_{1<i<j, x_{ij}^\nu=1} x_{ij} - \sum_{1<i<j} x_{ij}^\nu + 1\Bigg) \tag{40}$$

into the current problem and return to Step 2.

The optimality cut (40) uses the fact that a feasible solution is fully characterized by the x_{ij} variables associated with edges nonincident to the depot. They state that either the current solution must be maintained, in which case the cut becomes $\theta \geqslant Q(x^\nu)$, or a new solution must be identified, in which case the cut becomes $\theta \geqslant L$ or less and is thus redundant.

Markov decision models are defined on a state space. The system is observed at various transition times corresponding to moments at which a new customer is visited, and new decisions are taken at these moments. The state of the system at a given transition time is described by the set of customers already visited by the vehicle and by its current load. Because the state space is typically very large, this approach can only be applied to relatively small scale instances.

Heuristics for SVRPs are adaptations of methods originally designed for the deterministic case, which can be rather intricate because of the probability computations involved. In particular, computing the expected cost of a vehicle route is itself complicated and it may be advisable to use approximations if such computations are to be performed repeatedly within a search process (see, e.g., Gendreau et al., 1996). In what follows we study some particular classes of SVRPs.

5.1 The vehicle routing problem with stochastic customers

In vehicle routing problems with stochastic customers each vertex i is present with probability p_i. A first-stage solution consists of a set of vehicle routes visiting the depot and each customer exactly once. The set of absent customers is then revealed and the second-stage solution consists of following the first-stage routes while skipping the absent vertices. Jaillet (1985) laid the foundations of this line of research in his study of the Traveling Salesman Problem with Stochastic Customers (TSPSC). He proposed mathematical models and bounds, and he investigated a number of properties of the problem. For example, he showed that the solution of a deterministic TSP can be arbitrarily bad for the TSPSC. Also, even if the TSPSC is defined in a plane with Euclidean distances, an optimal cycle may cross itself, contrary to what happens for the TSP (Flood, 1956). Jézéquel (1985) and Rossi and Gavioli (1987) have proposed a number of heuristics for the TSPSC based on adaptations of the Clarke and Wright (1964) savings principle. Bertsimas (1988) and Bertsimas and Howell (1993) later investigated further properties of the TSPSC and proposed new heuristics, namely methods based on space filling curves (Bartholdi and Platzman, 1982) and on a 2-opt edge interchange mechanism. The first exact algorithm for the TSPSC is an integer L-shaped algorithm developed by Laporte et al. (1994) and capable of solving instances involving up to 50 customers. An extension of the TSPSC, called the Pickup and Delivery Traveling Salesman Problem with Stochastic Customers (PDTSPSC), was recently investigated by Beraldi et al. (2005). In this problem there are n requests, each consisting of a pickup location and of a delivery location, but request i only materializes with probability p_i. The authors show how to efficiently implement a low complexity interchange heuristic for this problem.

The Vehicle Routing Problem with Stochastic Customers (VRPSC) has been mostly studied in the context of unit demand customers. As in the TSPSC, vehicles follow the first-stage routes while skipping the absent customers and return to the depot to unload when their capacity is reached. This problem was first studied by Jézéquel (1985), Jaillet (1987), and Jaillet and Odoni (1988). The latter reference states two interesting properties of the VRPSC:

(1) even if travel costs are symmetric the overall solution cost is dependent on the direction of travel;
(2) larger vehicle capacities may yield larger solution costs.

Bertsimas' PhD thesis (Bertsimas, 1988) is an excellent source of information on this problem. It describes several properties, bounds and heuristics. Waters (1989) has studied the case of general integer demands and has compared three simple heuristics for this problem.

5.2 The vehicle routing problem with stochastic demands

The Vehicle Routing Problem with Stochastic Demands (VRPSD) has been the most studied stochastic VRP. In this problem customer demands are random and usually (but not always) independent. Tillman (1969) was probably the first to study this problem in a multidepot context. He proposed a savings based heuristic for its solution. The first, major study of the VRPSD can be attributed to Golden and Stewart (1978) who presented a chance constrained model and two recourse models. In the first of these a penalty proportional to vehicle overcapacity is imposed; in the second, the penalty is proportional to the expected demand in excess of the vehicle capacity. Several basic heuristics were implemented and tested. Dror and Trudeau (1986) developed further heuristics and showed that for this problem expected travel cost depends on the direction of travel even in the symmetric case. Again, Bertsimas' thesis (1988) constitutes a major contribution to the study of the VRPSD. It proposes several bounds, asymptotic results and properties for the case where ξ_i is equal to 1 with probability p_i, and equal to 0 otherwise. In their survey paper, Dror et al. (1989) have shown that some properties established by Jaillet (1985, 1988) and Jaillet and Odoni (1988) extend to the VRPSC, namely (1) in an optimal solution a vehicle route may intersect itself; (2) in a Euclidean problem customers are not necessarily visited in the order in which they appear on the convex hull of vertices; (3) segments of an optimal route are not necessarily optimal when considered separately. The latter property can have a major impact on the design of a dynamic algorithm for the VRPSD.

Laporte et al. (1989) proposed a two-index chance constrained model for the VRPSC as well as an associated branch-and-cut algorithm capable of solving instances with $n \leqslant 30$. They also introduced a bounded penalty model in which the cost of recourse associated with a given route cannot exceed a preset proportion of the first-stage route cost. The best exact solution approach for the VRPSD is again the integer L-shaped algorithm. Séguin (1994) and Gendreau et al. (1995) proposed the first implementation of this method for the solution of the VRPSD and were able to solve instances of up to 70 vertices. The most difficult case arises when the expected filling rate f of the vehicles is large. For example, when $f = 0.3$ instances with $n = 70$ can be solved optimally, but when $f = 1.0$ instances with $n = 10$ can rarely be solved. Using a similar approach, Hjorring and Holt (1999) solved one-vehicle instances ($m = 1$) with $0.95 \leqslant f \leqslant 1.05$ and $n = 90$. Laporte et al. (2002) imposed an additional restriction, namely that the expected demand of a route does not exceed the vehicle capacity, and they also exploited properties of the demand under known distributions (Poisson and normal) in the generation of lower bounding functionals on the cost of recourse. This enabled them to solve larger instances: for Poisson demands they solved instances with $f = 0.9$, $m = 4$, and $n = 25$, or with $m = 2$ and $n = 100$; for normal demands they solved instances with $f = 0.9$, $m = 3$, and $n = 50$.

Dynamic programming was applied by Secomandi (1998) to the VRP with stochastic demands. The largest instance solved to optimality with this method contained only 10 customers. The author also developed Neuro-Dynamic Programming (NDP) algorithms (Secomandi, 1998, 2000, 2003) for the same problem. Neuro-dynamic programming (see, e.g., Bertsekas, 1995) is a heuristic approach used to solve large-scale dynamic programs. It replaces the "cost-to-go" computations by proxies based on simulation and parametric function approximations. Secomandi (2000) compared two NDP implementations for the VRP with stochastic demands: an optimistic approximate iteration policy in which a neural network methodology is used to compute the approximations, and a rollout policy in which the cost-to-go is approximated by means of a heuristic. Computational results show that the second of these two policies is consistently and substantially superior to the first.

5.3 The vehicle routing problem with stochastic customers and demands

The VRP with stochastic customers and demands combines two difficult cases. This problem was first mentioned by Jézéquel (1985), Jaillet (1987), Jaillet and Odoni (1988), and was later formally defined by Bertsimas (1992). A first-stage solution visiting all customers is first constructed, the set of present customers is then revealed and their demand becomes known upon the arrival of the vehicle at the customer's location, routes are followed as planned but absent customers are skipped and the vehicle returns to the depot to unload whenever its capacity becomes attained. Benton and Rossetti (1992) proposed an algorithm which performs route reoptimizations whenever demands are revealed. One major difficulty in solving this problem lies in the computation of the objective function value. Recursions, bounds, asymptotic results, and a comparison of various reoptimization policies are provided by

Bertsimas (1992). Séguin (1994) and Gendreau et al. (1995) developed the first exact algorithm for this problem, based again on the integer L-shaped approach. They solved instances involving up to 46 customers and concluded that stochastic customers are a far more complicating factor than are stochastic demands. In a different study, Gendreau et al. (1996) developed a tabu search algorithm which uses an approximation of the objective function cost in order to ease computations. On a set of 825 instances with $6 \leqslant n \leqslant 46$ for which the optimum was known, an optimal solution was identified in 89.45% of the cases and the average optimality gap was 0.38%.

5.4 The vehicle routing problem with stochastic travel times

In the Vehicle Routing Problem with Stochastic Travel Times (VRPSTT) travel times on the edges and service times at the vertices are random variables. Vehicles follow their planned routes and may incur a penalty if the route duration exceeds a given deadline. It is natural to make this penalty proportional to the elapsed route duration in excess of the deadline (Laporte et al., 1992). Another possibility is to define a penalty proportional to the uncollected demand within the time limit, as is the case in a money collection application studied by Lambert et al. (1993).

The simplest case of the VRPSTT is the Traveling Salesman Problem with Stochastic Travel Time (TSPSTT) in which there is only one vehicle. It was first studied by Leipälä (1978) who computed the expected length of tours with random travel times. A common version of the TSPSTT is the case where the objective is to design a tour having the largest probability of being completed within the deadline. Kao (1978) proposed two heuristics for this problem: one based on dynamic programming and the other on implicit enumeration. Sniedovich (1981) has shown that dynamic programming applied to the same problem can be suboptimal because the monotonicity property required for this method is not satisfied in the TSPSTT. Carraway et al. (1989) later developed a so-called generalized dynamic programming algorithm that overcomes this difficulty. Kenyon and Morton (2003) have shown that an optimal TSPSTT can be identified by solving a deterministic TSP in which the travel and service times are replaced by their mean values. Verweij et al. (2003) have developed a heuristic for the case where a penalty proportional to route duration in excess of the deadline is incurred. The method uses a sample average approximation technique in which a sample of instance realizations is drawn and each is solved optimally by means of a deterministic technique. By repeating the method with different samples a statistical estimate of the optimality gap can be computed.

Laporte et al. (1992) were probably the first to provide exact algorithms for the VRPSTT. They formulated the chance constrained version of the problem, and they modeled a recourse version of the problem in which a penalty proportional to route duration in excess of the deadline is incurred. The problem was solved optimally by means of an integer L-shaped algorithm for $10 \leqslant n \leqslant 20$

and two to five travel time scenarios (each scenario corresponds to a different travel speed for the entire network). In a more recent study, Kenyon and Morton (2003) have investigated properties of VRPSTT solutions and have developed bounds on the objective function value. They have developed a heuristic that combines branch-and-cut and Monte Carlo simulation which, if run to completion, terminates with a solution value within a preset percentage of the optimum.

Finally, vehicle routing with stochastic travel time is frequently encountered in pickup and delivery problems such as those arising in truckload operations. Wang and Regan (2001) have proposed models for this class of problems under the presence of time windows.

Acknowledgements

This work has been supported by the Canadian Natural Sciences and Engineering Research Council under Grants 227837-00 and OGP0039682, by the Ministero dell'Università e della Ricerca (MIUR), and by the Consiglio Nazionale delle Ricerche (CNR), Italy. This support is gratefully acknowledged.

References

Achuthan, N.R., Caccetta, L., Hill, S.P. (1996). A new subtour elimination constraint for the vehicle routing problem. *European Journal of Operational Research* 91, 573–586.

Achuthan, N.R., Caccetta, L., Hill, S.P. (2003). An improved branch and cut algorithm for the capacitated vehicle routing problem. *Transportation Science* 37, 153–169.

Adelman, D. (2003a). Price-directed replenishment of subsets: Methodology and its application to inventory routing. *Manufacturing & Service Operations Management* 5, 348–371.

Adelman, D. (2003b). Internal transfer pricing for a decentralized operation with a shared supplier. Working paper, Graduate School of Business, The University of Chicago.

Adelman, D. (2004). A price-directed approach to stochastic inventory/routing. *Operations Research* 52, 499–514.

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

Altinkemer, K., Gavish, B. (1991). Parallel savings based heuristic for the delivery problem. *Operations Research* 39, 456–469.

Anily, S., Federgruen, A. (1990). One warehouse multiple retailer systems with vehicle routing costs. *Management Science* 36, 92–114.

Anily, S., Federgruen, A. (1991). Rejoinder to "One warehouse multiple retailer systems with vehicle routing costs". *Management Science* 37, 1497–1499.

Anily, S., Federgruen, A. (1993). Two-echelon distribution systems with vehicle routing costs and central inventories. *Operations Research* 41, 37–47.

Araque, J.R., Hall, L., Magnanti, T.L. (1990). Capacitated trees, capacitated routing and associated polyhedra. Discussion Paper 90-61, CORE, University of Louvain-la-Neuve, Belgium.

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

Augerat, P., Belenguer, J.M., Benavent, E., Corberán, A., Naddef, D., Rinaldi, G. (1995). Computational results with a branch and cut code for the capacitated vehicle routing problem. Technical Report RR 949-M, Université Joseph Fourier, Grenoble.

Augerat, P., Belenguer, J.M., Benavent, E., Corberán, A., Naddef, D. (1999). Separating capacity inequalities in the CVRP using tabu search. *European Journal of Operational Research* 106, 546–557.

Badeau, P., Gendreau, M., Guertin, F., Potvin, J.-Y., Taillard, É.D. (1997). A parallel tabu search heuristic for the vehicle routing problem with time windows. *Transportation Research C* 5, 109–122.

Baker, E., Schaffer, J. (1986). Computational experience with branch exchange heuristics for vehicle routing problems with time window constraints. *American Journal of Mathematical and Management Sciences* 6, 261–300.

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

Baldacci, R., Bodin, L., Mingozzi, A. (2006). The multiple disposal facilities and multiple inventory locations rollon–rolloff vehicle routing problem. *Computers & Operations Research* 33, 2667–2702.

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

Bard, J.F., Huang, L., Dror, M., Jaillet, P. (1998a). A branch and cut algorithm for the VRP with satellite facilities. *IIE Transactions* 30, 831–834.

Bard, J.F., Huang, L., Jaillet, P., Dror, M. (1998b). A decomposition approach to the inventory routing problem with satellite facilities. *Transportation Science* 32, 189–203.

Bard, J.F., Kontoravdis, G., Yu, G. (2002). A branch-and-cut procedure for the vehicle routing problem with time windows. *Transportation Science* 36, 250–269.

Barnes-Schuster, D., Bassok, Y. (1997). Direct shipping and the dynamic single-depot/multi-retailer inventory system. *European Journal of Operational Research* 101, 509–518.

Bartholdi, J.J., Platzman, L.K. (1982). An $N \log N$ planar traveling salesman heuristic based on space-filling curves. *Operations Research Letters* 1, 121–125.

Bartholdi, J.J., Platzman, L.K., Collins, R.L., Warden, W.H. (1983). A minimal technology routing system for meals on wheels. *Interfaces* 13 (3), 1–8.

Bastian, C., Rinnooy Kan, A.H.G. (1992). The stochastic vehicle routing problem revisited. *European Journal of Operational Research* 56, 407–412.

Battiti, R., Tecchiolli, G. (1994). The reactive tabu search. *ORSA Journal on Computing* 6, 126–140.

Bean, J.C. (1994). Genetic algorithms and random keys for the sequencing and optimization. *ORSA Journal on Computing* 6, 154–160.

Beasley, J.E. (1983). Route-first cluster-second methods for vehicle routing. *Omega* 11, 403–408.

Bell, W., Dalberto, L., Fisher, M.L., Greenfield, A., Jaikumar, R., Kedia, P., Mack, R., Prutzman, P. (1983). Improving the distribution of industrial gases with an on-line computerized routing and scheduling optimizer. *Interfaces* 13 (6), 4–23.

Benders, J.F. (1962). Partitioning procedures for solving mixed variables programming problems. *Numerische Mathematik* 4, 238–252.

Bent, R., Van Hentenryck, P. (2004). A two-stage hybrid local search for the vehicle routing problem with time windows. *Transportation Science* 38, 515–530.

Benton, W.C., Rossetti, M.D. (1992). The vehicle scheduling problem with intermittent customer demands. *Computers & Operations Research* 19, 521–531.

Beraldi, P., Ghiani, G., Laporte, G., Musmanno, G. (2005). Efficient neighbourhood search for the probabilistic pickup and delivery travelling salesman problem. *Networks* 46, 195–198.

Berger, J., Barkaoui, M. (2004). A new hybrid genetic algorithm for the capacitated vehicle routing problem. *Journal of the Operational Research Society* 54, 1254–1262.

Berger, J., Barkaoui, M., Bräysy, O. (2003). A route-directed hybrid genetic approach for the vehicle routing problem with time windows. *INFOR* 41, 179–194.

Bertazzi, L., Paletta, G., Speranza, M.G. (2002). Deterministic order-up-to level policies in an inventory routing problem. *Transportation Science* 36, 119–132.

Bertsekas, D.P. (1995). *Dynamic Programming and Optimal Control*. Athena Scientific, Belmont, MA.

Bertsimas, D.J. (1988). Probabilistic combinatorial optimization problems. PhD thesis, Operations Research Center, Massachusetts Institute of Technology, Cambridge, MA.

Bertsimas, D.J. (1992). A vehicle routing problem with stochastic demand. *Operations Research* 40, 574–585.

Bertsimas, D.J., Howell, L.H. (1993). Further results on the probabilistic traveling salesman problem. *European Journal of Operational Research* 65, 68–95.

Bertsimas, D.J., Simchi-Levi, D. (1996). A new generation of vehicle routing research: Robust algorithms addressing uncertainty. *Operations Research* 44, 286–304.

Bertsimas, D.J., Jaillet, P., Odoni, A.R. (1990). A priori optimisation. *Operations Research* 38, 1019–1033.

Blasum, U., Hochstättler, W. (2000). Application of the branch and cut method to the vehicle routing problem. Technical Report ZPR2000-386, ZPR, Universität zu Köln. Available at *http://www.zaik.uni-koeln.de/~paper*.

Bozkaya, B., Erkut, E., Laporte, G. (2003). A tabu search algorithm and adaptive memory procedure for political districting. *European Journal of Operational Research* 144, 12–26.

Bramel, J., Simchi-Levi, D. (1995). A location based heuristic for general routing problems. *Operations Research* 43, 649–660.

Bramel, J., Simchi-Levi, D. (1997). On the effectiveness of set covering formulations for the vehicle routing problem with time windows. *Operations Research* 45, 295–301.

Bramel, J., Simchi-Levi, D. (2002). Set-covering-based algorithms for the capacitated VRP. In: Toth, P., Vigo, D. (Eds.), *The Vehicle Routing Problem*. *SIAM Monographs on Discrete Mathematics and Applications*. SIAM, Philadelphia, pp. 85–108.

Brandão, J. (1998). Metaheuristic for the vehicle routing problem with time windows. In: Voss, S., Martello, S., Osman, I.H., Roucairol, C. (Eds.), *Meta-Heuristics: Advances and Trends in Local Search Paradigms for Optimization*. Kluwer Academic, Boston, pp. 19–36.

Bräysy, O. (2002). Fast local searches for the vehicle routing problem with time windows. *INFOR* 40, 319–330.

Bräysy, O. (2003). A reactive variable neighborhood search for the vehicle routing problem with time windows. *INFORMS Journal on Computing* 15, 347–368.

Bräysy, O., Gendreau, M. (2005a). Vehicle routing problem with time windows, Part I: Route construction and local search algorithms. *Transportation Science* 39, 104–118.

Bräysy, O., Gendreau, M. (2005b). Vehicle routing problem with time windows, Part II: Metaheuristics. *Transportation Science* 39, 119–139.

Burns, L.D., Hall, R.W., Blumenfeld, D.E., Daganzo, C.F. (1985). Distribution strategies that minimize transportation and inventory costs. *Operations Research* 33, 469–490.

Campbell, A.M., Savelsbergh, M.W.P. (2004a). Delivery volume optimization. *Transportation Science* 38, 210–223.

Campbell, A.M., Savelsbergh, M.W.P. (2004b). A decomposition approach for the inventory-routing problem. *Transportation Science* 38, 488–502.

Campbell, A.M., Savelsbergh, M.W.P. (2004c). Efficient insertion heuristics for vehicle routing and scheduling problems. *Transportation Science* 38, 269–378.

Campbell, A.M., Clarke, L., Kleywegt, A.J., Savelsbergh, M.W.P. (1998). The inventory routing problem. In: Crainic, T.G., Laporte, G. (Eds.), *Fleet Management and Logistics*. Kluwer Academic, Boston, pp. 95–112.

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

Carraway, R.L., Morin, T.L., Moskowitz, H. (1989). Generalized dynamic programming for stochastic combinatorial optimization. *Operations Research* 37, 819–829.

Çetinkaya, S., Lee, C.Y. (2000). Stock replenishment and shipment scheduling for vendor managed inventory systems. *Management Science* 46, 217–232.

Chabrier, A. (2006). Vehicle routing problem with elementary shortest path based column generation. *Computers & Operations Research* 33, 2972–2990.

Chan, L.M., Federgruen, A., Simchi-Levi, D. (1998). Probabilistic analyses and practical algorithms for inventory-routing models. *Operations Research* 46, 96–106.

Chiang, W.-C., Russell, R.A. (1997). A reactive tabu search metaheuristic for the vehicle routing problem with time windows. *INFORMS Journal on Computing* 9, 417–430.

Chien, T., Balakrishnan, A., Wong, R. (1989). An integrated inventory allocation and vehicle routing problem. *Transportation Science* 23, 67–76.

Christiansen, M. (1999). Decomposition of a combined inventory and time constrained ship routing problem. *Transportation Science* 33, 3–16.

Christiansen, M., Nygreen, B. (1998a). A method for solving ship routing problems with inventory constraints. *Annals of Operations Research* 81, 357–378.

Christiansen, M., Nygreen, B. (1998b). Modelling path flows for a combined ship routing and inventory management problem. *Annals of Operations Research* 82, 391–412.

Christofides, N., Mingozzi, A. (1989). Vehicle routing: Practical and algorithmic aspects. In: Van Rijn, C.F.H. (Ed.), *Logistics: Where Ends Have to Meet*. Pergamon, Oxford, pp. 30–48.

Christofides, N., Mingozzi, A., Toth, P. (1979). The vehicle routing problem. In: Christofides, N., Mingozzi, A., Toth, P., Sandi, C. (Eds.), *Combinatorial Optimization*. Wiley, Chichester, pp. 315–338.

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

Christofides, N., Mingozzi, A., Toth, P. (1981b). State-space relaxation procedures for the computation of bounds to routing problems. *Networks* 11, 145–164.

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

Cook, W., Rich, J.L. (1999). A parallel cutting-plane algorithm for the vehicle routing problem with time windows. Technical Report TR99-04, Computational and Applied Mathematics Department, Rice University, TX.

Cordeau, J.-F., Laporte, G. (2001). A tabu search algorithm for the site dependent vehicle routing problem with time windows. *INFOR* 39, 292–298.

Cordeau, J.-F., Laporte, G. (2004). Tabu search heuristics for the vehicle routing problem. In: Rego, C., Alidaee, B. (Eds.), *Metaheuristic Optimization via Memory and Evolution: Tabu Search and Scatter Search*. Kluwer Academic, Boston, pp. 145–163.

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

Cordeau, J.-F., Laporte, G., Mercier, A. (2001). A unified tabu search heuristic for vehicle routing problems with time windows. *Journal of the Operational Research Society* 52, 928–936.

Cordeau, J.-F., Desaulniers, G., Desrosiers, J., Solomon, M.M., Soumis, F. (2002a). VRP with Time Windows. In: Toth, P., Vigo, D. (Eds.), *The Vehicle Routing Problem*. *SIAM Monographs on Discrete Mathematics and Applications*. SIAM, Philadelphia, pp. 157–193.

Cordeau, J.-F., Gendreau, M., Laporte, G., Potvin, J.-Y., Semet, F. (2002b). A guide to vehicle routing heuristics. *Journal of the Operational Research Society* 53, 512–522.

Cordeau, J.-F., Laporte, G., Mercier, A. (2004). An improved tabu search algorithm for the handling of route duration constraints in vehicle routing problems with time windows. *Journal of the Operational Research Society* 55, 542–546.

Cordeau, J.-F., Gendreau, M., Hertz, A., Laporte, G., Sormany, J.-S. (2005). New heuristics for the vehicle routing problem. In: Langevin, A., Riopel, D. (Eds.), *Logistics Systems: Design and Optimization*. Springer-Verlag, New York, pp. 279–297.

Cordone, R., Wolfler Calvo, R. (2001). A heuristic for the vehicle routing problem with time windows. *Journal of Heuristics* 7, 107–129.

Cornuéjols, G., Harche, F. (1993). Polyhedral study of the capacitated vehicle routing problem. *Mathematical Programming* 60, 21–52.

Croes, A. (1958). A method for solving traveling salesman problems. *Operations Research* 6, 791–812.

Danna, E., Le Pape, C. (2003). Accelerating branch-and-price with local search: A case study on the vehicle routing problem with time windows. Technical Report 03-006, ILOG.

Dantzig, G.B., Ramser, J.M. (1959). The truck dispatching problem. *Management Science* 6, 81–91.

Dantzig, G.B., Wolfe, P. (1960). Decomposition principle for linear programming. *Operations Research* 8, 101–111.

De Backer, B., Furnon, V., Kilby, P., Prosser, P., Shaw, P. (2000). Solving vehicle routing problems using constraint programming and metaheuristics. *Journal of Heuristics* 6, 501–523.

Dell'Amico, M., Toth, P. (2000). Algorithms and codes for dense assignment problems: The state of the art. *Discrete Applied Mathematics* 100, 17–48.

Desrochers, M., Laporte, G. (1991). Improvements and extensions to the Miller–Tucker–Zemlin subtour elimination constraints. *Operations Research Letters* 10, 27–36.

Desrochers, M., Soumis, F. (1988). A generalized permanent labeling algorithm for the shortest path problem with time windows. *INFOR* 26, 191–212.

Desrochers, M., Verhoog, T.W. (1989). A matching based savings algorithm for the vehicle routing problem. Les Cahiers du GERAD G–89–04, HEC Montréal.

Desrochers, M., Lenstra, J.K., Savelsbergh, M.W.P., Soumis, F. (1988). Vehicle routing with time windows: Optimization and approximation. In: Golden, B.L., Assad, A.A. (Eds.), *Vehicle Routing: Methods and Studies*. North-Holland, Amsterdam, pp. 65–84.

Desrochers, M., Desrosiers, J., Solomon, M.M. (1992). A new optimization algorithm for the vehicle routing problem with time windows. *Operations Research* 40, 342–354.

Dorigo, M., Di Caro, G., Gambardella, L.M. (1999). Ant algorithms for discrete optimization. *Artificial Life* 5, 137–172.

Drezner, Z. (2003). A new genetic algorithm for the quadratic assignment problem. *INFORMS Journal on Computing* 15, 320–330.

Dror, M., Ball, M.O. (1987). Inventory/routing: Reduction from an annual to a short period problem. *Naval Research Logistics Quarterly* 34, 891–905.

Dror, M., Trudeau, P. (1986). Stochastic vehicle routing with modified savings algorithm. *European Journal of Operational Reserach* 23, 228–235.

Dror, M., Ball, M.O., Golden, B.L. (1985). A computational comparison of algorithms for the inventory routing problem. *Annals of Operations Research* 4, 3–23.

Dror, M., Laporte, G., Trudeau, P. (1989). Vehicle routing with stochastic demands: Properties and solution frameworks. *Transportation Science* 23, 166–176.

Dror, M., Laporte, G., Louveaux, F.V. (1993). Vehicle routing with stochastic demands and restricted failures. *Zeitschrift für Operations Research* 37, 273–283.

Dueck, G. (1990). New optimization heuristics, the great deluge algorithm and the record-to-record travel. Technical report, IBM Germany, Heidelberg Scientific Center.

Dueck, G. (1993). New optimization heuristics: The great deluge algorithm and the record-to-record travel. *Journal of Computational Physics* 104, 86–92.

Ergun, Ö., Orlin, J.B., Steele-Feldman, A. (2003). Creating very large scale neighborhoods out of smaller ones by compounding moves: A study on the vehicle routing problem. MIT Sloan working Paper 4393-02, Massachusetts Institute of Technology, Cambridge, MA.

Federgruen, A., Zipkin, P. (1984). A combined vehicle routing and inventory allocation problem. *Operations Research* 32, 1019–1036.

Fischetti, M., Toth, P. (1989). An additive bounding procedure for combinatorial optimization problems. *Operations Research* 37, 319–328.

Fischetti, M., Toth, P., Vigo, D. (1994). A branch-and-bound algorithm for the capacitated vehicle routing problem on directed graphs. *Operations Research* 42, 846–859.

Fisher, M.L. (1994). Optimal solution of vehicle routing problems using minimum k-trees. *Operations Research* 42, 626–642.

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

Fisher, M.L., Greenfield, A., Jaikumar, R., Kedia, P. (1982). Real-time scheduling of a bulk delivery fleet: Practical application of Lagrangean relaxation. Technical report, The Wharton School, University of Pennsylvania.

Fisher, M.L., Jörnsten, K.O., Madsen, O.B.G. (1997). Vehicle routing with time windows – two optimization algorithms. *Operations Research* 45, 488–492.

Flood, M.M. (1956). The travelling salesman problem. *Operations Research* 4, 61–75.

Foster, B.A., Ryan, D.M. (1976). An integer programming approach to the vehicle scheduling problem. *Operations Research* 27, 367–384.

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

Gallego, G., Simchi-Levi, D. (1990). On the effectiveness of direct shipping strategy for the one-warehouse multi-retailer *r*-systems. *Management Science* 36, 240–243.

Gambardella, L.M., Taillard, É.D., Agazzi, G. (1999). MACS-VRPTW: A multiple ant colony system for vehicle routing problems with time windows. In: Corne, D., Dorigo, M., Glover, F. (Eds.), *New Ideas in Optimization*. McGraw-Hill, London, pp. 63–76.

Gaur, V., Fisher, M.L. (2004). A periodic inventory routing problem at a supermarket chain. *Operations Research* 52, 813–822.

Gehring, H., Homberger, J. (2002). Parallelization of a two-phase metaheuristic for routing problems with time windows. *Journal of Heuristics* 8, 251–276.

Gendreau, M., Hertz, A., Laporte, G. (1992). New insertion and post-optimization procedures for the traveling selesman problem. *Operations Research* 40, 1083–1094.

Gendreau, M., Hertz, A., Laporte, G. (1994). A tabu search heuristic for the vehicle routing problem. *Management Science* 40, 1276–1290.

Gendreau, M., Laporte, G., Séguin, R. (1995). An exact algorithm for the vehicle routing problem with stochastic customers and demands. *Transportation Science* 29, 143–155.

Gendreau, M., Laporte, G., Séguin, R. (1996). A tabu search algorithm for the vehicle routing problem with stochastic demands and customers. *Operations Research* 44, 469–477.

Gendreau, M., Hertz, A., Laporte, G., Stan, M. (1998). A generalized insertion heuristic for the traveling salesman problem with time windows. *Operations Research* 43, 330–335.

Gendreau, M., Laporte, G., Potvin, J.-Y. (2002). Metaheuristics for the capacitated VRP. In: Toth, P., Vigo, D. (Eds.), *The Vehicle Routing Problem*. *SIAM Monographs on Discrete Mathematics and Applications*. SIAM, Philadelphia, pp. 129–154.

Ghaziri, H. (1993). Algorithmes connexionistes pour l'optimisation combinatoire. Thèse de doctorat, École Polytechnique Fédérale de Lausanne, Switzerland.

Ghiani, G., Laporte, G., Semet, F. (2006). The black and white traveling salesman problem. *Operations Research* 54, 366–378.

Gillett, B.E., Miller, L.R. (1974). A heuristic algorithm for the vehicle-dispatch problem. *Operations Research* 21, 340–349.

Glover, F. (1992). New ejection chain and alternating path methods for traveling salesman problems. In: Balci, O., Sharda, R., Zenios, S. (Eds.), *Computer Science and Operations Research: New Developments in Their Interfaces*. Pergamon, Oxford, pp. 449–509.

Golden, B.L., Stewart, W.R. (1978). Vehicle routing with probabilistic demands. In: Hogben, D., Fife, D. (Eds.) *Computer Science and Statistics: Tenth Annual Symposium on the Interface. NBS Special Publication*, vol. 503, pp. 252–259.

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

Golden, B.L., Assad, A.A., Dahl, R. (1984). Analysis of a large scale vehicle routing problem with an inventory component. *Large Scale Systems* 7, 181–190.

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

Golden, B.L., Assad, A.A., Wasil, E.A. (2002). Routing vehicles in the real world: Applications in the solid waste, beverage, food, dairy, and newspaper industries. In: Toth, P., Vigo, D. (Eds.), *The Vehicle Routing Problem*. *SIAM Monographs on Discrete Mathematics and Applications*. SIAM, Philadelphia, pp. 245–286.

Gouveia, L. (1995). A result on projection for the vehicle routing problem. *Journal of Operational Research Society* 85, 610–624.

Hadjiconstantinou, E., Christofides, N., Mingozzi, A. (1995). A new exact algorithm for the vehicle routing problem based on *q*-paths and *k*-shortest paths relaxations. *Annals of Operations Research* 61, 21–43.

Haimovich, M., Rinnooy Kan, A.H.G. (1985). Bounds and heuristics for capacitated routing problems. *Mathematics of Operations Research* 10, 527–542.

Haughton, M.A. (1998). The performance of route modification and demand stabilization strategies in stochastic vehicle routing. *Transportation Research* 32, 551–566.

Haughton, M.A. (2000). Quantifying the benefits of route reoptimisation under stochastic customer demands. *Journal of the Operational Research Society* 51, 320–332.

Held, M., Karp, R.M. (1971). The traveling salesman problem and minimum spanning trees: Part II. *Mathematical Programming* 1, 6–25.

Held, M., Wolfe, P., Crowder, M.P. (1974). Validation of the subgradient optimization. *Mathematical Programming* 6, 62–88.

Herer, Y., Roundy, R. (1997). Heuristics for a one-warehouse multiretailer distribution problem with performance bounds. *Operations Research* 45, 102–115.

Hjorring, C., Holt, J. (1999). New optimality cuts for a single-vehicle stochastic routing problem. *Annals of Operations Research* 86, 569–585.

Homberger, J., Gehring, H. (1999). Two evolutionary metaheuristics for the vehicle routing problem with time windows. *INFOR* 37, 297–318.

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

Hvattum, L.M., Løkketangen, A., Laporte, G. (2006). Solving a dynamic and stochastic vehicle routing problem with a sample scenario hedging heuristic. *Transportation Science*, in press.

Ioannou, G., Kritikos, M., Prastacos, G. (2001). A greedy look-ahead heuristic for the vehicle routing problem with time windows. *Journal of the Operational Research Society* 52, 523–537.

Irnich, S., Villeneuve, D. (2003). The shortest path problem with resource constraints and k-cycle elimination for $k \geqslant 3$. Technical report, Rheinisch-Westfälische Technische Hochschule, Aachen, Germany.

Jaillet, P. (1985). Probabilistic traveling salesman problem. PhD thesis, Operations Research Center, Massachusetts Institute of Technology, Cambridge, MA.

Jaillet, P. (1987). Stochastic routing problems. In: Andreatta, G., Mason, F., Serafini, P. (Eds.), *Stochastics in Combinatorial Optimization*. World Scientific, Singapore, pp. 197–213.

Jaillet, P. (1988). A priori solution of a traveling salesman problem in which a random subset of the customers are visited. *Operations Research* 36, 929–936.

Jaillet, P., Odoni, A.R. (1988). The probabilistic vehicle routing problem. In: Golden, B.L., Assad, A.A. (Eds.), *Vehicle Routing: Methods and Studies*. North-Holland, Amsterdam, pp. 293–318.

Jaillet, P., Bard, J.F., Huang, L., Dror, M. (2002). Delivery cost approximations for inventory routing problems in a rolling horizon framework. *Transportation Science* 3, 292–300.

Jézéquel, A. (1985). Probabilistic vehicle routing problems. MSc dissertation, Department of Civil Engineering, Massachusetts Institute of Technology, Cambridge, MA.

Kallehauge, B., Larsen, J., Madsen, O.B.G. (2006). Lagrangean duality applied to the vehicle routing with time windows. *Computers & Operations Research* 33, 1464–1487.

Kao, E.P.C. (1978). A preference order dynamic program for a stochastic traveling salesman problem. *Operations Research* 26, 1033–1045.

Kenyon, A.S., Morton, D.P. (2003). Stochastic vehicle routing with random travel times. *Transportation Science* 37, 69–82.

Kilby, P.J., Prosser, P., Shaw, P. (1998). Guided local search for the vehicle routing problem with time windows. In: Voss, S., Martello, S., Osman, I.H., Roucairol, C. (Eds.), *Meta Heuristics: Advances and Trends in Local Search Paradigms for Optimisation*. Kluwer Academic, Boston, pp. 473–486.

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

Kleywegt, A.J., Nori, V., Savelsbergh, M.W.L. (2002). The stochastic inventory routing problem with direct deliveries. *Transportation Science* 36, 94–118.

Kleywegt, A.J., Nori, V., Savelsbergh, M.W.L. (2004). Dynamic programming approximations for a stochastic inventory routing problem. *Transportation Science* 38, 42–70.

Kohl, N., Madsen, O.B.G. (1997). An optimization algorithm for the vehicle routing problem with time windows based on Lagrangean relaxation. *Operations Research* 45, 395–406.

Kohl, N., Desrosiers, J., Madsen, O.B.G., Solomon, M.M., Soumis, F. (1999). 2-path cuts for the vehicle routing problem with time windows. *Transportation Science* 33, 101–116.

Kolen, A.W.J., Rinnooy Kan, A.H.G., Trienekens, H.W.J.M. (1987). Vehicle routing with time windows. *Operations Research* 35, 256–273.

Kontoravdis, G., Bard, J.F. (1995). A GRASP for the vehicle routing problem with time windows. *ORSA Journal on Computing* 7, 10–23.

Lambert, V., Laporte, G., Louveaux, F.V. (1993). Designing collection routes through bank branches. *Computers & Operations Research* 20, 783–791.

Laporte, G., Louveaux, F.V. (1990). Formulations and bounds for the stochastic capacitated vehicle routing problem with uncertain supplies. In: Gabzewicz, J., Richard, J.-F., Wolsey, L. (Eds.), *Economic Decision Making: Games, Econometrics and Optimisation*. North-Holland, Amsterdam, pp. 443–455.

Laporte, G., Louveaux, F.V. (1993). The integer L-shaped method for stochastic integer programs with complete recourse. *Operations Research Letters* 13, 133–142.

Laporte, G., Louveaux, F.V. (1998). Solving stochastic routing problems with the integer L-shaped method. In: Crainic, T.G., Laporte, G. (Eds.), *Fleet Management and Logistics*. Kluwer Academic, Boston, pp. 159–167.

Laporte, G., Nobert, Y. (1987). Exact algorithms for the vehicle routing problem. *Annals of Discrete Mathematics* 31, 147–184.

Laporte, G., Semet, F. (2002). Classical heuristics for the capacitated VRP. In: Toth, P., Vigo, D. (Eds.), *The Vehicle Routing Problem*. *SIAM Monographs on Discrete Mathematics and Applications*. SIAM, Philadelphia, pp. 109–128.

Laporte, G., Nobert, Y., Desrochers, M. (1985). Optimal routing under capacity and distance restrictions. *Operations Research* 33, 1050–1073.

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

Laporte, G., Louveaux, F.V., Mercure, H. (1989). Models and exact solutions for a class of stochastic location-routing problems. *European Journal of Operational Research* 39, 71–78.

Laporte, G., Louveaux, F.V., Mercure, H. (1992). The vehicle routing problem with stochastic travel times. *Transportation Science* 26, 161–170.

Laporte, G., Louveaux, F.V., Mercure, H. (1994). A priori optimization of the probabilistic traveling salesman problem. *Operations Research* 42, 543–549.

Laporte, G., Louveaux, F.V., Van hamme, L. (2002). An integer L-shaped algorithm for the capacitated vehicle routing problem with stochastic demands. *Operations Research* 50, 415–423.

Larson, R.C. (1988). Transportation of sludge to the 106-mile site: An inventory routing problem for fleet sizing and logistic system design. *Transportation Science* 22, 186–198.

Lau, H.C., Liu, Q., Ono, H. (2002). Integrating local search and network flow to solve the inventory routing problem. *American Association for Artificial Intelligence* 2, 9–14.

Lau, H.C., Sim, M., Teo, K.M. (2003). Vehicle routing problem with time windows and a limited number of vehicles. *European Journal of Operational Research* 148, 559–569.

Leipälä, T. (1978). On the solutions of stochastic traveling salesman problems. *European Journal of Operational Research* 2, 291–297.

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

Li, F., Golden, B.L., Wasil, E.A. (2005). Very large-scale vehicle routing: New test problems, algorithms and results. *Computers & Operations Research* 32, 1165–1179.

Li, H., Lim, A. (2003). Local search with annealing-like restarts to solve the VRPTW. *European Journal of Operational Research* 150, 115–127.

Lin, S. (1965). Computer solutions of the travelling salesman problem. *Bell System Technical Journal* 44, 2245–2269.

Louveaux, F.V. (1998). An introduction to stochastic transportation models. In: Labbé, M., Laporte, G., Tanczos, K., Toint, P. (Eds.), *Operations Research and Decision Aid Methodologies in Traffic and Transportation Management*. *NATO ASI Series F: Computer and Systems Sciences*, vol. 166. Springer-Verlag, Berlin/Heidelberg, pp. 244–263.

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

Martinhon, C., Lucena, A., Maculan, N. (2000). A relax and cut algorithm for the vehicle routing problem. Technical Report RT-05/00, Universidade Federal Fluminense, Niterói, Brasil.

Mester, D., Bräysy, O. (2005). Active guided evolution strategies for large scale vehicle routing problem with time windows. *Computers & Operations Research* 32, 1593–1614.

Miller, D.L. (1995). A matching based exact algorithm for capacitated vehicle routing problems. *ORSA Journal on Computing* 7, 1–9.

Miller, D.L., Pekny, J.F. (1995). A staged primal-dual algorithm for perfect *b*-matching with edge capacities. *ORSA Journal on Computing* 7, 298–320.

Mingozzi, A., Christofides, N., Hadjiconstantinou, E. (1994). An exact algorithm for the vehicle routing problem based on the set partitioning formulation. Technical report, Department of Mathematics, University of Bologna, Italy.

Minkoff, A. (1993). A Markov decision model and decomposition heuristic for dynamic vehicle dispatching. *Operations Research* 41, 77–90.

Mladenović, N., Hansen, P. (1997). Variable neighborhood search. *Computers & Operations Research* 24, 1097–1100.

Mole, R.H., Jameson, S.R. (1976). A sequential route-building algorithm employing a generalized savings criterion. *Operational Research Quarterly* 27, 503–511.

Moscato, P., Cotta, C. (2003). A gentle introduction to memetic algorithms. In: Glover, F., Kochenberger, G.A. (Eds.), *Handbook of Metaheuristics*. Kluwer Academic, Boston, pp. 105–144.

Naddef, D., Rinaldi, G. (2002). Branch-and-cut algorithms for the capacitated VRP. In: Toth, P., Vigo, D. (Eds.), *The Vehicle Routing Problem*. *SIAM Monographs on Discrete Mathematics and Applications*. SIAM, Philadelphia, pp. 53–84.

Nelson, M.D., Nygard, K.E., Griffin, J.H., Shreve, W.E. (1985). Implementation techniques for the vehicle routing problem. *Computers & Operations Research* 12, 273–283.

Or, I. (1976). Traveling salesman-type combinatorial problems and their relation to the logistics of blood banking. PhD thesis, Department of Industrial Engineering and Management Science, Northwestern University, Evanston, IL.

Osman, I.H. (1993). Metastrategy simulated annealing and tabu search algorithms for the vehicle routing problem. *Annals of Operations Research* 41, 421–451.

Paessens, H. (1988). The savings algorithm for the vehicle routing problem. *European Journal of Operational Research* 34, 336–344.

Potvin, J.-Y. (1996). Genetic algorithms for the traveling salesman problem. *Annals of Operations Research* 63, 339–370.

Potvin, J.-Y., Bengio, S. (1996). The vehicle routing problem with time windows – Part II: Genetic search. *INFORMS Journal on Computing* 8, 165–172.

Potvin, J.-Y., Rousseau, J.-M. (1993). A parallel route building algorithm for the vehicle routing and scheduling problem with time windows. *European Journal of Operational Research* 66, 331–340.

Potvin, J.-Y., Rousseau, J.-M. (1995). An exchange heuristic for routing problems with time windows. *Journal of the Operational Research Society* 46, 1433–1446.

Potvin, J.-Y., Kervahut, T., Garcia, B.L., Rousseau, J.-M. (1996). The vehicle routing problem with time windows – Part I: Tabu search. *INFORMS Journal on Computing* 8, 158–164.

Prins, C. (2004). A simple and effective evolutionary algorithm for the vehicle routing problem. *Computers & Operations Research* 31, 1985–2002.

Ralphs, T.K., Kopman, L., Pulleyblank, W.R., Trotter, Jr., L.E. (2003). On the capacitated vehicle routing problem. *Mathematical Programming B* 94, 343–359.

Rechenberg, I. (1973). *Evolutionsstrategie*. Fromman-Holzboog, Stuttgart, Germany.

Rego, C. (1998). A subpath ejection method for the vehicle routing problem. *Management Science* 44, 1447–1459.

Rego, C., Roucairol, C. (1996). A parallel tabu search algorithm using ejection chains for the vehicle routing problem. In: Osman, I.H., Kelly, J.P. (Eds.), *Meta-Heuristics: Theory and Applications*. Kluwer Academic, Boston, pp. 661–675.

Reimann, M., Rubio, R., Wein, L.M. (1999). Heavy traffic analysis of the dynamic stochastic inventory-routing problem. *Transportation Science* 33, 361–380.

Reimann, M., Doerner, K., Hartl, R.F. (2004). D-ants: Savings based ants divide and conquer for the vehicle routing problem. *Computers & Operations Research* 31, 563–591.

Renaud, J., Boctor, F.F., Laporte, G. (1996a). A fast composite heuristic for the symmetric traveling salesman problem. *INFORMS Journal on Computing* 8, 134–143.

Renaud, J., Boctor, F.F., Laporte, G. (1996b). An improved petal heuristic for the vehicle routing problem. *Journal of the Operational Research Society* 47, 329–336.

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

Rossi, F.A., Gavioli, I. (1987). Aspects of heuristic methods in the "Probabilistic traveling salesman problem". In: Andreatta, G., Mason, F., Serafini, P. (Eds.), *Stochastics in Combinatorial Optimization*. World Scientific, Singapore, pp. 214–227.

Russell, R.A. (1977). An effective heuristic for the M-tour traveling salesman problem with some side conditions. *Operations Research* 25, 517–524.

Russell, R.A. (1995). Hybrid heuristics for the vehicle routing problem with time windows. *Transportation Science* 29, 156–166.

Ryan, D.M., Hjorring, C., Glover, F. (1993). Extensions of the petal method for vehicle routing. *Journal of Operational Research Society* 44, 289–296.

Savelsbergh, M.W.P. (1985). Local search in routing problems with time windows. *Annals of Operations Research* 4, 285–305.

Savelsbergh, M.W.P. (1990). En efficient implementation of local search algorithms for constrained routing problems. *European Journal of Operational Research* 47, 75–85.

Savelsbergh, M.W.P. (1992). The vehicle routing problem with time windows: Minimizing route duration. *ORSA Journal on Computing* 4, 146–154.

Savelsbergh, M.W.P., Song, J.-H. (2005). Inventory routing with continuous moves. *Computers & Operations Research*, in press.

Schulze, J., Fahle, T. (1999). A parallel algorithm for the vehicle routing problem with time window constraints. *Annals of Operations Research* 86, 585–607.

Secomandi, N. (1998). Exact and heuristic dynamic programming algorithms for the vehicle routing problem with stochastic demands. PhD dissertation, Faculty of the College of Business Administration, University of Houston, TX.

Secomandi, N. (2000). Comparing neuro-dynamic programming algorithms for the vehicle routing problem with stochastic demands. *Computers & Operations Research* 27, 1201–1225.

Secomandi, N. (2003). Analysis of a rollout approach to sequencing problems with stochastic routing applications. *Journal of Heuristics* 9, 321–352.

Séguin, R. (1994). Problèmes stochastiques de tournées de véhicules. PhD thesis, Département d'informatique et de recherche opérationnelle, Université de Montréal, Canada.

Semet, F., Taillard, É.D. (1993). Solving real-life vehicle routing problems efficiently using tabu search. *Annals of Operations Research* 41, 469–488.

Shaw, P. (1998). Using constraint programming and local search methods to solve vehicle routing problems. In: Maher, M., Puget, J.-F. (Eds.), *Principles and Practice of Constraint Programming*. Springer-Verlag, New York, pp. 417–431.

Sniedovich, M. (1981). Analysis of a preference order traveling salesman problem. *Operations Research* 29, 1234–1237.

Solomon, M.M. (1987). Algorithms for the vehicle routing and scheduling problems with time window constraints. *Operations Research* 35, 254–265.

Solomon, M.M., Baker, E.K., Schaffer, J.R. (1988). Vehicle routing and scheduling problems with time window constraints: Efficient implementations of solution improvement procedures. In: Golden, B.L., Assad, A.A. (Eds.), *Vehicle Routing: Methods and Studies*. North-Holland, Amsterdam, pp. 85–106.

Song, J.-H., Savelsbergh, M.W.P. (2005). Performance measurement for inventory routing. *Transportation Science*, in press.

Stewart, W.R., Golden, B.L. (1983). Stochastic vehicle routing: A comprehensive approach. *European Journal of Operational Research* 14, 371–385.

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

Taillard, É.D., Badeau, P., Gendreau, M., Guertin, F., Potvin, J.-Y. (1997). A tabu search heuristic for the vehicle routing problem with soft time windows. *Transportation Science* 31, 170–186.

Tan, K.C., Lee, L.H., Ou, K. (2001). Hybrid genetic algorithms in solving vehicle routing problems with time window constraints. *Asia-Pacific Journal of Operational Research* 18, 170–186.

Tarantilis, C.-D., Kiranoudis, C.T. (2002). Bone route: Adaptive memory method for effective fleet management. *Annals of Operations Research* 115, 227–241.

Thangiah, S.R., Petrovic, P. (1998). Introduction to genetic heuristics and vehicle routing problems with complex constraints. In: *Advances in Computational and Stochastic Optimization, Logic Programming, and Heuristic Search*. *Operations Research/Computer Science Interfaces*, vol. 9. Kluwer Academic, Boston, pp. 253–286.

Thompson, P.M., Psaraftis, H.N. (1993). Cyclic transfer algorithms for multi-vehicle routing and scheduling problems. *Operations Research* 41, 935–946.

Tillman, F. (1969). The multiple terminal delivery problem with probabilistic demands. *Transportation Science* 3, 192–204.

Toth, P., Vigo, D. (1995). An exact algorithm for the capacitated shortest spanning arborescence. *Annals of Operations Research* 61, 121–142.

Toth, P., Vigo, D. (1997). An exact algorithm for the vehicle routing problem with backhauls. *Transportation Science* 31, 372–385.

Toth, P., Vigo, D. (1998). Exact algorithms for vehicle routing. In: Crainic, T., Laporte, G. (Eds.), *Fleet Management and Logistics*. Kluwer Academic, Boston, pp. 1–31.

Toth, P., Vigo, D. (Eds.) (2002a). *The Vehicle Routing Problem*. *SIAM Monographs on Discrete Mathematics and Applications*. SIAM, Philadelphia.

Toth, P., Vigo, D. (2002b). An overview of vehicle routing problems. In: Toth, P., Vigo, D. (Eds.), *The Vehicle Routing Problem*. *SIAM Monographs on Discrete Mathematics and Applications*. SIAM, Philadelphia, pp. 1–26.

Toth, P., Vigo, D. (2002c). Branch-and-bound algorithms for the capacitated VRP. In: Toth, P., Vigo, D. (Eds.), *The Vehicle Routing Problem*. *SIAM Monographs on Discrete Mathematics and Applications*. SIAM, Philadelphia, pp. 29–51.

Toth, P., Vigo, D. (2002d). Models, relaxations and exact approaches for the capacitated vehicle routing problem. *Discrete Applied Mathematics* 123, 487–512.

Toth, P., Vigo, D. (2003). The granular tabu search and its application to the vehicle routing problem. *INFORMS Journal on Computing* 15, 333–346.

Trudeau, P., Dror, M. (1992). Stochastic inventory routing: Route design with stockouts and route failures. *Transportation Science* 26, 171–184.

Van Breedam, A. (1994). 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. PhD dissertation, University of Antwerp, Belgium.

Van Slyke, R., Wets, R.J.-B. (1969). *L*-shaped programs with applications to optimal control and stochastic programming. *SIAM Journal of Applied Mathematics* 17, 638–653.

Verweij, B., Ahmed, S., Kleywegt, A.J., Nemhauser, G.L., Shapiro, A. (2003). The sample average approximation method applied to stochastic routing problems: A computational study. *Computational Optimization and Applications* 24, 289–333.

Vigo, D. (1996). A heuristic algorithm for the asymmetric capacitated vehicle routing problem. *European Journal of Operational Research* 89, 108–126.

Viswanathan, S., Mathur, K. (1997). Integrating routing and inventory decisions in one-warehouse multiretailer multiproduct distribution systems. *Management Science* 43, 294–312.

Volgenant, A., Jonker, R. (1983). The symmetric traveling salesman problem and edge exchange in minimal 1-trees. *European Journal of Operational Research* 12, 395–403.

Voudouris, C. (1997). Guided local search for combinatorial problems. Dissertation, University of Essex, United Kingdom.

Waller, M., Johnson, M.E., Davis, T. (1999). Vendor-managed inventory in the retail supply chain. *Journal of Business Logistics* 20, 183–203.

Wang, X., Regan, A.C. (2001). Assignment models for local truckload trucking problems with stochastic service times and time window constraints. *Transportation Research Record* 1171, 61–68.

Wark, P., Holt, J. (1994). A repeated matching heuristic for the vehicle routing problem. *Journal of Operational Research Society* 45, 1156–1167.

Waters, C.D.J. (1989). Vehicle-scheduling problems with uncertainty and omitted customers. *Journal of the Operational Research Society* 40, 1099–1108.

Webb, R., Larson, R. (1995). Period and phase of customer replenishment: A new approach to the strategic inventory/routing problem. *European Journal of Operational Research* 85, 132–148.

Willard, J.A.G. (1989). Vehicle routing using *r*-optimal tabu search. MSc dissertation, The Management School, Imperial College, London.

Wren, A. (1971). *Computers in Transport Planning and Operation*. Ian Allan, London.

Wren, A., Holliday, A. (1972). Computer scheduling of vehicles from one or more depots to a number of delivery points. *Operational Research Quarterly* 23, 333–344.

Xu, J., Kelly, J.P. (1996). A network flow-based tabu search heuristic for the vehicle routing problem. *Transportation Science* 30, 379–393.

Yang, W.H., Mathur, K., Ballou, R.H. (2000). Stochastic vehicle routing with restocking. *Transportation Science* 34, 99–112.