



Available online at www.sciencedirect.com



Operations Research Letters 35 (2007) 773–781

**Operations
Research
Letters**

www.elsevier.com/locate/orl

A branch-and-price algorithm for the capacitated vehicle routing problem with stochastic demands

Christian H. Christiansen*, Jens Lysgaard

Department of Business Studies, Aarhus School of Business, University of Aarhus, Fuglesangs Allé 4, DK-8210 Aarhus V, Denmark

Received 8 June 2006; accepted 20 December 2006

Available online 21 February 2007

Abstract

This article introduces a new exact algorithm for the capacitated vehicle routing problem with stochastic demands (CVRPSD). The CVRPSD can be formulated as a set partitioning problem and it is shown that the associated column generation subproblem can be solved using a dynamic programming scheme. Computational experiments show promising results. © 2007 Elsevier B.V. All rights reserved.

Keywords: Routing; Stochastic programming; Logistics

1. Introduction

The deterministic capacitated vehicle routing problem (CVRP) has been widely studied in the literature, and can be described as follows. A set of customers must be provided with known quantities of a common commodity from a single depot. To make the deliveries, a fleet of identical vehicles, each with a given capacity, is available. The distance traveled by the vehicles to conduct the deliveries defines the travel cost of a route. The objective is to find a route collection of minimum total travel cost under the restrictions that (i) each route begins and ends at the depot, (ii) each customer is serviced exactly once, and (iii)

the total demand on any route does not exceed the vehicle capacity.

In this article we consider the capacitated vehicle routing problem with stochastic demands (CVRPSD). The CVRPSD differs from the CVRP with respect to the following points:

1. In the CVRPSD, the customers' demands are stochastic variables of which only the probability distribution for each customer is assumed known at the time of planning.
2. In the CVRPSD, it is the *expected* total travel cost which is subject to minimization.
3. In the CVRPSD, the total actual demand on a route may exceed the vehicle capacity. In such cases a *failure* is said to occur. A strategy is required for updating the routes in case of such an event. The actual action resulting from this strategy is called a

* Corresponding author.

E-mail address: chc@asb.dk (C.H. Christiansen).

recourse action. The particular strategy affects the expected cost of a given route, so the strategy must be known at the time of planning.

The CVRPSD has not received nearly the same level of attention as the CVRP. In the literature several reasons are given why only limited attention is paid to the CVRPSD. The most important of these might be the fact that the CVRP problem in itself is very hard to solve, and adding a stochastic dimension to the problem only increases the intractability.

Nonetheless, neglecting the stochastic nature of demands during the planning of the routes can incur substantially higher expected costs than if the stochastic demands had been explicitly included in the route planning. For the TSP this has been thoroughly illustrated by Laporte and Trudeau in [8]. Moreover, neglecting the stochastic nature of demands might result in an inoptimal solution not only as a consequence of the customer sequence on each route as illustrated in [8], but also as a consequence of an inoptimal allocation of customers among routes.

In this article we formulate the CVRPSD as a two stage stochastic program with fixed recourse and capacity constraints. The article is organized as follows. Section 2 gives a brief literature review regarding research on the CVRPSD. Section 3 focuses on the CVRPSD modeling. Section 4 presents our new algorithm. Section 5 deals with the construction of test instances and Section 6 provides computational results obtained with the proposed algorithm.

2. Literature review

To the best of our knowledge, the first to address the CVRPSD was Tillman in 1969 [20]. Tillman considered a multidepot variant of the CVRP with Poisson distributed demands. The model considered a cost tradeoff between exceeding the vehicle capacity and finishing the route with excess capacity. The solution approach was a modification of the savings algorithm originally introduced by Clarke and Wright in 1964 [4]. For further review of savings-based approaches for both constraint programming models and two stage stochastic programming models see [9].

Regarding the two stage stochastic programming approach, Bertsimas [1] formulated two widely

accepted recourse actions (A) and (B), respectively. These are based on two different assumptions regarding the time at which a customer's actual demand becomes known. Strategy (A) assumes that a customer's actual demand becomes known only upon arrival at the customer. Strategy (B), however, assumes that actual demands become known early enough to enable the vehicle to skip customers with zero actual demand. The recourse action under both strategies is to deplete the vehicle at the point of failure, return to the depot to replenish, and continue the originally planned route from the point of failure. In the particular case where a customer's demand equals the remaining load on the vehicle, the vehicle returns to the depot to replenish before visiting the next customer.

In their article in 1993, Laporte and Louveaux developed an integer L-shaped method for stochastic programs with recourse [16]. Their approach is based on adding feasibility cuts to a relaxed flow formulation of a CVRPSD until an integer feasible solution is found. If no integer solution can be found through the introduction of feasibility cuts, branching is applied. During this phase the expected failure cost is represented by a lower bound on its true value. When an integer solution is identified, the lower bound is compared to the actual expected failure cost of the identified integer solution. If a gap exists between the lower bound on the expected failure cost and the actual expected failure cost for the identified integer solution, an optimality cut is added. However, in their article no computational results were presented. The method has been applied to the CVRPSD in 1995 [12], 1998 [17], and 2002 [18]. In the latter, instances with up to 100 customers were solved.

In 1999, Hjorring and Holt applied the L-shaped algorithm to the single vehicle CVRPSD and developed a new set of optimality cuts to improve the procedure. Furthermore they developed an improved lower bound on the expected failure cost [14].

When considering the computational results obtained by the L-shaped algorithm, it is obvious that the best results are obtained for instances with small expected demands relative to the vehicle capacity. This is not surprising since the L-shaped algorithm is based on a cutting procedure. In fact, as is the case for the CVRP, the effectiveness of the cutting-based algorithms seems to diminish as the capacity constraints become more restrictive.

For the deterministic CVRP, the instances that have proven hardest to solve are characterized by restrictive vehicle capacities. This can be illustrated by considering the CVRP test instances E-n76-k7, E-n76-k8, E-n76-k10, and E-n76-k14, where the x in E-n76-kx stands for the minimum number of routes given the customer demands and the vehicle capacity, i.e., E-n76-k10 and E-n76-k14, have been solved only recently (by the algorithm in [10]), despite the fact that they were stated as early as 1969 [2]. Branch-and-cut algorithms had previously solved only E-n76-k7 and E-n76-k8.

Branch-and-price algorithms, on the other hand, benefit in general from tighter constraints, in the sense that a more restricted solution space is beneficial for the solution of the column generation subproblem. For example, branch-and-price has been the dominating approach to the VRPTW particularly in the case of tight time windows [5].

Similarly, there is reason to believe that column generation would be a promising approach for solving the type of CVRPSD instances that have proven most difficult to solve with existing (cutting-plane-based) exact algorithms.

In this article we develop a new branch-and-price algorithm for the CVRPSD. Specifically, we formulate the CVRPSD as a set partitioning problem and develop a dynamic programming algorithm for solving the associated column generation subproblem. To ensure integrality, we develop a new branching procedure which exploits the asymmetric nature of the route costs.

3. Notation and model formulation

To formally describe our model, let $G = (V, E)$ be an undirected graph, with vertex set $V = \{0, \dots, n\}$ and edge set E . Vertex 0 represents a depot, and each of the vertices in $V_c = \{1, \dots, n\}$ represents a customer. With each edge $\{i, j\}$ is associated a travel cost d_{ij} . Let Ψ represent any stochastic distribution with an accumulative property, meaning that the sum of two or more independent and Ψ distributed random variables is also a Ψ distributed random variable. This latter assumption is non-restrictive for practical purposes since it holds true for a number of well-known stochastic

distributions. This includes, e.g., the Poisson distribution, the normal distribution, and the gamma distribution. For further details on stochastic distributions satisfying this assumption see [19].

The vehicle capacity is denoted by Q . For each customer i , we let q_i denote the random variable describing the demand at customer i . We assume that the q_i 's are independent and Ψ distributed. We denote by $E[q_i] > 0$ and $V[q_i] > 0$ the expected demand and the variance of q_i , respectively.

We define a *route* as a path $(0, z_1, \dots, z_k, 0)$ where $z_1, \dots, z_k \in V_c$ and $z_i \neq z_{i+1}$ for $i = 1, \dots, k-1$. Further, we say that the route is *elementary* if z_1, \dots, z_k are all different, otherwise we say that the route is non-elementary. Moreover, we say that the (elementary or non-elementary) route is feasible if and only if $\sum_{i=1, \dots, k} E[q_{z_i}] \leq Q$. As in [18], we do not allow routes whose total expected demand exceeds Q , as such routes would systematically fail. For any feasible elementary route r , let c_r denote its expected cost. Further, let \mathfrak{R}_e denote the set of all feasible elementary routes.

Let α_{ir} be a parameter of value 1 if route r visits customer i and 0 otherwise. Further, let x_r be a variable of value 1 if route r is chosen and 0 otherwise. This leads to the following set partitioning formulation:

$$(P_{\text{org}})$$

$$\min \quad \sum_{r \in \mathfrak{R}_e} c_r x_r \quad (1)$$

$$\text{s.t.} \quad \sum_{r \in \mathfrak{R}_e} \alpha_{ir} x_r = 1 \quad \forall i \in V_c, \quad (2)$$

$$x_r \in \{0, 1\} \quad \forall r \in \mathfrak{R}_e. \quad (3)$$

Objective (1) minimizes the total expected distribution cost. Constraints (2) ensure that each customer is contained in exactly one route, whereas constraints (3) are the binary constraints on the decision variables.

The recourse that we have chosen is identical to strategy (A), with one exception. If the vehicle is exactly depleted at some customer, we assume that it continues, without an intermediate return to the depot, along the route until it, at some customer j , encounters a positive demand (or reaches the depot). Since a customer's actual demand is not known until arrival at the customer, a customer with an actual demand of zero cannot be skipped. The failure cost corresponding

to this recourse is $2d_{0j}$ (with $d_{00} = 0$). This recourse strategy was chosen to ensure comparability with the existing results obtained for the CVRPSD.

We now consider in detail how the expected cost c_r of a route $r \in \mathfrak{R}_e$ can be calculated. For notational simplicity, we assume that the route is $(0, 1, \dots, k, 0)$.

The total expected cost c_r can be decomposed into two elements. The first element (C_D) is the deterministic cost of following the route, which must be done irrespectively of actual demands. This cost element is simply $d_{01} + d_{k0} + \sum_{i=1}^{k-1} d_{i,i+1}$.

The second element (C_F) is the total extra distance expected to be traveled due to failures along the route r . More specifically, it is the expected failure cost of the route r . We note that the extra traveling represented by C_F must be done *in addition* to traveling the path represented by C_D , so that C_D and C_F are additive. Indeed, we obtain $c_r = C_D + C_F$. The complicating part of c_r is the calculation of C_F . As it turns out, however, we are able to separate C_F into k additive terms, one for each customer $1, \dots, k$. This is shown in the following.

The probability $F(i|u, Q)$ that the total actual demand on the path $(0, 1, \dots, i)$ does not exceed uQ , where u is an integer > 0 , can be calculated as follows:

$$F(i|u, Q) = P\left(\sum_{h=1}^i q_h \leq uQ\right). \quad (4)$$

Proposition 1. *The probability in (4) depends only on the total expected demand along the path to i , not on how this total expected demand is divided among the customers on the path.*

Proof. Our assumption that the demands are independent and follow the distribution Ψ implies that the expression $\sum_{h=1}^i q_h$ is in itself a Ψ distributed random variable. The expected value is $\sum_{h=1}^i E[q_h]$ and the variance is $\sum_{h=1}^i V[q_h]$. As such, the distribution of $\sum_{h=1}^i q_h$ does not depend on the individual expected demands and variances, but only on $\sum_{h=1}^i E[q_h]$ and $\sum_{h=1}^i V[q_h]$. \square

Proposition 1 is the key to our algorithm for solving the column generation subproblem in Section 4.2.

As a consequence of Proposition 1, we can for the remainder of this section leave the assumption of the

specific path $(0, \dots, i)$ since the customers' sequence on the path segment $(0, \dots, i - 1)$ is irrelevant to the calculation of the expected failure cost at customer i . We now consider any elementary path from 0 to i along which the total expected demand and variance are given values, say, μ and σ^2 , respectively. We let $\Psi(\mu, \sigma^2)$ denote a Ψ distributed random variable with an expected value of μ and a variance of σ^2 .

In addition, we define $F(\mu, \sigma^2, U)$ as the probability that the total actual demand on a path, whose total expected demand is μ and whose variance is σ^2 , does not exceed U , where U is a positive integer:

$$F(\mu, \sigma^2, U) = P(\Psi(\mu, \sigma^2) \leq U). \quad (5)$$

We now turn to consider *failures* in more detail.

Definition 1. For a given integer $u \geq 1$ and any elementary path $(0, \dots, j, i)$, we say that the u th failure occurs at customer i if and only if the total actual demand on the path $(0, \dots, j, i)$ exceeds uQ and the total actual demand on the path $(0, \dots, j)$ does not exceed uQ .

Let μ and σ^2 denote the total expected demand and the variance on the path $(0, \dots, i)$. The probability that the u th failure occurs at customer i is then $F(\mu - E[q_i], \sigma^2 - V[q_i], uQ) - F(\mu, \sigma^2, uQ)$.

For any elementary path $(0, \dots, i)$ with total expected demand μ and a variance of σ^2 , the expected number of failures $\text{FAIL}(\mu, \sigma^2, i)$ at customer i can then be calculated by summing over all possible failures:

$$\begin{aligned} \text{FAIL}(\mu, \sigma^2, i) &= \sum_{u=1}^{\infty} F(\mu - E[q_i], \sigma^2 - V[q_i], uQ) \\ &\quad - F(\mu, \sigma^2, uQ), \end{aligned} \quad (6)$$

which in practical computations is approximated by replacing ∞ with some sufficiently large number.

Since the failure cost $2d_{0i}$ is incurred for every failure at customer i , the *expected failure cost* $\text{EFC}(\mu, \sigma^2, i)$ at customer i , for any elementary path $(0, \dots, i)$ with total expected demand μ and variance σ^2 , can be calculated as follows:

$$\text{EFC}(\mu, \sigma^2, i) = 2d_{0i} \text{FAIL}(\mu, \sigma^2, i). \quad (7)$$

The significance of (7) is that it allows us to express the total expected failure cost along a given path as a sum of contributions from the visited customers. This in turn provides us with a domination criterion. In particular, among all paths characterized by (μ, σ^2, i) , the path having the lowest expected cost dominates all other paths with the same characteristics. Our column generation algorithm in Section 4.2 exploits this observation.

The calculation of $EFC(\mu, \sigma^2, i)$ is originally introduced for the CVRPSD in [8], in which the authors embed the failure cost into a standard flow formulation of the CVRP, leading to a flow formulation of the CVRPSD.

However, an integer solution to the flow formulation is needed to calculate the expected failure cost. Laporte and Louveaux handle this in the L-shaped algorithm [16] by modifying the objective function. In particular, they replace the expected failure cost by a lower bound valid for any integer solution.

We suggest a new solution method based on (P_{org}) . This formulation possesses the important property that the customer sequence on each route is known even when an integer solution to (P_{org}) is not. This enables us to calculate the expected failure cost of each route before an integer solution to (P_{org}) is known. We can then embed the expected failure cost into the expected route cost. Our solution procedure exploits this property.

4. Solution procedure

Our solution procedure is based on Dantzig–Wolfe decomposition. As is usual in set partitioning-based approaches to vehicle routing, we make a few modifications to the formulation in order to obtain a more tractable problem.

4.1. The master problem

To obtain the master problem denoted P_M , we (i) relax the integrality constraints (3), (ii) change the partitioning constraints to covering constraints in order to obtain a smaller dual solution space, (iii) enlarge the set of feasible routes by permitting non-elementary routes ($\mathfrak{R} \supseteq \mathfrak{R}_e$), and (iv) introduce the coefficient a_{ir} . This coefficient equals the number of

times that customer i is visited on route r . This leads to the following master problem:

(P_M)

$$\min \sum_{r \in \mathfrak{R}} c_r x_r \quad (8)$$

$$\text{s.t. } \sum_{r \in \mathfrak{R}} a_{ir} x_r \geq 1 \quad \forall i \in V_c, \quad (9)$$

$$x_r \geq 0 \quad \forall r \in \mathfrak{R}. \quad (10)$$

In P_M , the set \mathfrak{R} contains all feasible elementary routes as well as all feasible non-elementary routes without 2-cycles (i, j, i) . A customer i contributes $E[q_i]$ to the total expected demand on every arrival at i .

We initialize P_M by n single-customer routes and solve this LP. By solving P_M a vector of dual prices π_1, \dots, π_n is obtained related to the constraint set (9), so that the dual price associated with customer i is π_i . The dual prices are used in the subproblem in the search for one or more columns with negative reduced cost. If such columns are identified, they are added to the LP, which is then reoptimized. The steps of column generation and LP reoptimization are repeated until no further columns with negative reduced cost exist. The current solution is then optimal for P_M .

If the LP solution is integer and all constraints (9) are satisfied with equality, it is optimal for P_{org} . (If not all inequalities (9) are satisfied with equality in an integer solution, we change the inequalities to equalities, resolve the LP, and continue the iterative procedure.) If the LP solution is fractional, we resort to branching in order to eventually obtain an integer solution. The overall *branch-and-price* algorithm is a variant of a branch-and-bound algorithm in which column generation is performed at each node in the branch-and-bound tree.

4.2. Column generation

The idea of column generation was originally introduced in 1984 [7], where it was applied to the CVRP with time windows. For an extensive review of column generation, see [6].

The column generation subproblem has frequently been solved by a dynamic programming algorithm, which effectively solves a shortest path problem on a

particular acyclic network. This applies to, e.g., the approaches in [3,13]. The generated paths are invariably restricted to those without 2-cycles, which can be done without increasing the computational complexity using the idea in [15]. Our column generation approach is quite similar to this, but with the modification that expected failure costs must be taken into account. We note that the calculation of expected failure costs is not affected by allowing non-elementary routes.

For the purpose of applying dynamic programming to the column generation subproblem, we assume that both $E[q_i]$ and $V[q_i]$ are integer values, for $i = 1, \dots, n$.

Further, we calculate an upper bound V_{\max} on the total variance on any feasible elementary route. This is done by solving the following 0–1 knapsack problem, where y_i equals 1 if customer i is visited on the route and 0 otherwise:

$$V_{\max} = \max \sum_{i \in V_c} V[q_i] y_i \quad (11)$$

$$\text{s.t. } \sum_{i \in V_c} E[q_i] y_i \leq Q, \quad (12)$$

$$y_i \in \{0, 1\} \quad \forall i \in V_c. \quad (13)$$

During column generation, we permit only routes whose total variance does not exceed V_{\max} . This may implicitly eliminate some non-elementary routes and may therefore reduce the weakening which results from permitting non-elementary routes.

For the purpose of solving the column generation subproblem, we can now create the graph $G_S = (V_S, A_S)$.

V_S contains $(n+1)QV_{\max} + 1$ vertices. Each vertex $v(\mu, \sigma^2, i)$, for $\mu = 1, \dots, Q$, $\sigma^2 = 1, \dots, V_{\max}$, and $i = 0, \dots, n$, represents a particular set of paths, namely those paths without 2-cycles from 0 to i which have a total expected demand of μ and a total variance of σ^2 . All vertices with $i = 0$ represent feasible routes, except for $v(0, 0, 0)$ which is the origin of all paths.

Beginning with an empty set A_S , we add arcs to A_S as follows:

1. For $i = 1, \dots, n$, add an arc from $v(0, 0, 0)$ to $v(E[q_i], V[q_i], i)$ and set its cost to $d_{0i} + \text{EFC}(E[q_i], V[q_i], i) - \pi_i$.
2. For each ordered pair $i, j \in V_c$, $i \neq j$, each $\mu = 1, \dots, Q-1$, and each $\sigma^2 = 1, \dots, V_{\max}$, add an arc

from $v(\mu, \sigma^2, j)$ to $v(\mu+E[q_i], \sigma^2+V[q_i], i)$ (provided that $\mu+E[q_i] \leq Q$ and $\sigma^2+V[q_i] \leq V_{\max}$) and set its cost to $d_{ji} + \text{EFC}(\mu+E[q_i], \sigma^2+V[q_i], i) - \pi_i$.

3. For each $\mu = 1, \dots, Q$, each $\sigma^2 = 1, \dots, V_{\max}$, and each $j = 1, \dots, n$, add an arc from $v(\mu, \sigma^2, j)$ to $v(\mu, \sigma^2, 0)$ and set its cost to d_{j0} .

The shortest path in G_S from $v(0, 0, 0)$ to $v(\mu, \sigma^2, 0)$, for any $\mu \in \{1, \dots, Q\}$ and any $\sigma \in \{1, \dots, V_{\max}\}$, represents the route of minimum reduced cost among routes with total expected demand μ and total variance σ^2 . As such, computing the shortest path from $v(0, 0, 0)$ to $v(\mu, \sigma^2, 0)$ for all $\mu = 1, \dots, Q$ and all $\sigma^2 = 1, \dots, V_{\max}$ effectively solves the column generation subproblem. This can be done in $O(n^2 Q V_{\max})$ time, also in the case that 2-cycles are prohibited.

4.3. The branching strategies

We adopt the branching strategy proposed by Gélinas et al. in 1995 for the vehicle routing problem with time windows and backhauls [11]. They introduced a branching procedure based on the time windows. In each branching, the parent problem is split into two restricted problems each containing a restricted time window for some customer.

In a similar way, we branch on the resources. If there exists a route whose decision variable has a fractional value, then there is a customer who is either visited more than once on this route or visited on both this route and on one or more other routes.

This motivates a branching strategy based on the accumulated expected demand. In particular, let customer i be visited on two paths characterized by (μ_1, σ_1^2, i) and (μ_2, σ_2^2, i) , respectively, with $\mu_1 \neq \mu_2$. From this we create two new problems by choosing a threshold value δ such that $\mu_1 \leq \delta < \mu_2$. In the first problem, we forbid paths characterized by $(\mu > \delta, \sigma^2, i)$ and in the second problem we forbid paths characterized by $(\mu \leq \delta, \sigma^2, i)$. Clearly, the original fractional solution is infeasible in both new problems.

Branching on the accumulated expected demand is referred to as strategy A for the remainder of the article. Branching on single flow variables (as in [13]) is referred to as strategy B. Both strategies have been

included in the computational testing in order to compare their relative performance.

5. Construction of test instances

For testing our algorithm we assume that all demands are Poisson distributed. This assumption dramatically reduces the size of the graph G_S , because the Poisson distribution has the well-known property $\mu = \sigma^2$. This implies that the number of vertices in G_S is reduced to $(n + 1)Q + 1$. The vertices are then denoted by $v(0, 0)$ for the starting vertex of any route and $v(\mu, i)$ for all other vertices.

The construction of test instances for the CVRPSD was done on the basis of existing test instances for the CVRP. In particular, the instances chosen include all instances of the Augerat test sets A and P and the Christofides and Eilon test set with up to 60 customers. The data for these CVRP instances are available at www.branchandcut.org. No test results for instances with more than 60 customers are included, because preliminary testing showed that none of these was currently solvable.

To convert the CVRP instances into CVRPSD instances the following three steps were performed for each instance: (1) all customer demands are assumed Poisson distributed, (2) for each customer, we set the customer's *expected* demand equal to the customer's original deterministic demand, and (3) travel distances, node locations, and vehicle capacity are left unchanged and deterministic.

For all instances the capacity and demands are divided by their greatest common denominator. This decreases the time consumed by the column generation procedure.

6. Computational results

All instances were run on a Pentium Centrino 1500 MHz computer with 480 MB of RAM. The time limit was set at 1200 s. For each instance, data were recorded when the root node was solved, and when the algorithm terminated either due to timeout or optimality. The data collected are presented in Table 1.

Column 1 shows the name of the instance. The format of the name is '(Instance code) - (number of nodes) - (minimum number of vehicles

required)'. Column 2 shows the objective value after solving the root node. Column 3 shows the number of columns generated at the root node, and column 4 shows the number of times the master problem was solved. Column 5 shows the sum of the decision variables after having solved the root node.

Columns 6–11 show, for each of the two branching strategies, the total time in seconds spent by the algorithm (the sign '##' indicates that the algorithm has reached the time limit), the number of nodes solved in the branching tree, and the lower bound on the optimal objective value, respectively. If a lower bound is marked with an upper case (*) the solution is optimal. Column 12 shows an upper bound on the optimal objective value. This bound can originate from three different sources: (1) the optimal solution, (2) the best integer solution found during branching, and (3) the best obtainable expected cost given the optimal solution for the CVRP counterpart (column 14). Column 13 shows the sum of the decision variables. Column 14 shows the expected cost of an optimal solution to the CVRP counterpart, obtained by taking the best direction of each route in the CVRP solution from www.branchandcut.org. Finally, column 15 shows the ratio between the objective value found at the root node and the best known upper bound ((LB root)/(UB)).

When considering the instances solved, it is clear that the algorithm seems to work better on problems with tight capacity constraints.

On several instances, the number of routes in the optimal solution exceeds the minimum possible for serving all customers. For the instance P-55-15 this is particularly evident in that three extra routes are formed.

Comparing column 14 to the optimal expected solution cost shows that neglecting the stochastic nature of demands during the route planning can incur substantial cost increases.

Strategy A solves more problems and tends to search fewer nodes before reaching optimum than does strategy B. This points to the conclusion that strategy A in general performs better than strategy B.

The largest instance successfully solved by our algorithm was an instance with 60 customers and 16 routes. In comparison, the L-shaped algorithm solved instances with up to 100 customers in [18], but only with 2 vehicles. In addition, instances with up to 4

Table 1
Computational results

Inst.	LB root	Cols.	P_M	Routes root	Time A	Time B	Nodes A	Nodes B	LB A	LB B	UB	Routes opt.	Best det.	$\frac{LB Root}{UB}$
A-32-5	817.31	1221	67	5	282	##	2467	13799	853.6*	842.32	853.6	5	890.13	0.96
A-33-5	700.01	962	44	5	8	9	117	107	704.2*	704.2*	704.2	5	722.99	0.99
A-33-6	775	832	43	6.05	49	##	909	26195	793.9*	789.65	793.9	6	816.58	0.98
A-34-5	803.26	1212	62	5.31	##	##	16059	15155	825.26	823.17	827.87	—	839.95	0.97
A-36-5	838.83	1641	113	5	##	##	8035	8093	852.09	851.01	907.55	—	907.55	0.92
A-37-5	687.4	1758	95	5	##	##	9191	8773	707.54	702.99	708.34	—	709.83	0.97
A-37-6	1007.98	1237	67	6.48	##	##	16195	14965	1030.44	1021.83	1030.75	—	1069.32	0.98
A-38-5	739.19	1207	55	5.47	##	##	14499	13865	761.12	755.05	778.09	—	831.99	0.95
A-39-5	866.92	1722	73	6.07	3	4	9	23	869.18*	869.18*	869.18	6	903.26	1
A-39-6	850.09	1462	84	6.04	279	##	2431	11289	876.6*	872.28	876.6	6	960.81	0.97
A-44-6	1007.55	1871	59	6.43	##	##	11077	10053	1021.29	1017.27	1025.48	—	1047.18	0.98
A-45-6	984.38	1861	95	6.75	##	##	9313	10285	1006.88	1001.01	1096.19	—	1096.19	0.9
A-45-7	1254.23	1954	84	7.13	882	##	5365	9553	1264.83*	1263.09	1264.83	7	1302.2	0.99
A-46-7	986.39	1951	84	7	##	##	8149	9257	999.87	997.25	1002.41	—	1069.66	0.98
A-48-7	1160.52	2482	91	7.13	##	##	7729	8221	1180.22	1177.8	1248.27	—	1248.27	0.93
A-53-7	1093.64	3273	128	7.69	##	##	5385	5321	1109.34	1106.67	1180.1	—	1180.1	0.93
A-54-7	1262.49	2636	98	7.44	##	##	5925	5741	1279.93	1272.65	1342.87	—	1342.87	0.94
A-55-9	1148.4	1881	55	9.76	##	##	9979	9597	1173.56	1159.82	1264.18	—	1264.18	0.91
A-60-9	1489.82	3043	108	9.2	##	##	6889	5737	1503.65	1498.23	1608.4	—	1608.4	0.93
E-22-4	409.86	460	34	4.2	1	1	9	13	411.57*	411.57*	411.57	4	411.73	1
E-33-4	844.35	3030	148	4	86	##	73	1161	850.27*	850.07	850.27	4	850.27	0.99
E-51-5	538.75	3290	138	5.48	##	##	2771	2643	544.63	543.67	553.26	—	553.26	0.97
P-16-8	511.27	79	8	8.5	0	0	17	29	512.82*	512.82*	512.82	8	512.82	1
P-19-2	210.9	620	63	2.17	153	154	1815	1529	224.06*	224.06*	224.06	3	229.68	0.94
P-20-2	221.11	871	88	2.18	352	496	3191	3819	233.05*	233.05*	233.05	2	233.05	0.95
P-21-2	217.75	1037	100	2.14	5	7	27	45	218.96*	218.96*	218.96	2	218.96	0.99
P-22-2	223.67	1129	103	2.21	219	329	1335	1903	231.26*	231.26*	231.26	2	231.26	0.97
P-22-8	677.97	205	18	8.92	0	0	65	53	681.06*	681.06*	681.06	9	707.8	1
P-23-8	619.52	22	19	9	1	1	0	0	619.52*	619.52*	619.52	9	662.31	1
P-40-5	471.47	1905	80	5	9	6	35	29	472.5*	472.5*	472.5	5	475.45	1
P-45-5	519.03	2153	101	5.14	##	##	4561	4265	527.77	526.63	546.05	—	546.05	0.95
P-50-7	573.66	2016	67	7.13	##	##	5311	5273	581.19	579.97	606.41	—	606.41	0.95
P-50-8	659.67	1522	55	8.78	##	##	10409	10359	666.9	666.86	724.69	—	724.69	0.91
P-50-10	750.27	1320	50	11	##	##	18901	18529	756.52	757.44	792.2	—	792.2	0.95
P-51-10	802.58	1436	57	10.99	430	995	5431	11177	809.7*	809.7*	809.7	11	859.24	0.99
P-55-7	582.12	2782	102	7	##	##	2441	3303	585.47	585.46	616.44	—	616.44	0.94
P-55-10	734.69	1747	56	10.17	##	##	11257	11525	740.02	741.25	797.21	—	797.21	0.92
P-55-15	1062.67	867	21	17.18	792	1756	20027	40873	1068.05*	1068.05*	1068.05	18	1191.34	0.99
P-60-10	793.71	2102	59	10.58	##	##	4969	6961	798.63	799.58	831.24	—	831.24	0.95
P-60-15	1080.85	1175	29	16.08	##	1348	18347	19029	1085.12	1085.49*	1085.49	16	1133.3	1

vehicles were solved for 25 customers. It was concluded in [18] that problem difficulty and computational effort increase sharply with the number of vehicles, when using the L-shaped algorithm.

This result indicates that our algorithm complements the L-shaped method well and allows a broader range of problems to be solved.

Acknowledgment

Thanks are due to an anonymous referee whose helpful comments led to an improved presentation.

References

- [1] D.J. Bertsimas, A vehicle routing problem with stochastic demand, *Oper. Res.* 40 (1992) 574–585.
- [2] N. Christofides, S. Eilon, An algorithm for the vehicle-dispatching problem, *Oper. Res. Quart.* 20 (1969) 309–318.
- [3] N. Christofides, A. Mingozzi, P. Toth, Exact algorithms for the vehicle routing problem, based on spanning tree and shortest path relaxations, *Math. Programming* 20 (1981) 255–282.
- [4] G. Clarke, J.W. Wright, Scheduling of vehicles from a central depot to a number of delivery points, *Oper. Res.* 12 (1964) 568–581.
- [5] W. Cook, J.L. Rich, A parallel cutting-plane algorithm for the vehicle routing problem with time windows, Technical Report, Computational and Applied Mathematics, Rice University, 1999.
- [6] G. Desaulniers, J. Desrosiers, M.M. Solomon (Eds.), *Column Generation*, Springer, New York, 2005.
- [7] J. Desrosiers, F. Soumis, M. Desrochers, Routing with time windows by column generation, *Networks* 14 (1984) 545–565.
- [8] M. Dror, G. Laporte, P. Trudeau, Vehicle routing with stochastic demands: properties and solution frameworks, *Transportation Sci.* 23 (1989) 166–176.
- [9] M. Dror, P. Trudeau, Stochastic vehicle routing with modified savings algorithm, *European J. Oper. Res.* 23 (1986) 228–235.
- [10] R. Fukasawa, H. Longo, J. Lysgaard, M.P. de Aragão, M. Reis, E. Uchoa, R.F. Werneck, Robust branch-and-cut-and-price for the capacitated vehicle routing problem, *Math. Programming* 106 (2006) 491–511.
- [11] S. Gélinas, M. Desrochers, J. Desrosiers, M.M. Solomon, A new branching strategy for time constrained routing problems with application to backhauling, *Ann. Oper. Res.* 61 (1995) 91–109.
- [12] M. Gendreau, G. Laporte, R. Séguin, An exact algorithm for the vehicle routing problem with stochastic demands and customers, *Transportation Sci.* 29 (1995) 143–155.
- [13] E. Hadjiconstantinou, N. Christofides, A. Mingozzi, A new exact algorithm for the vehicle routing problem based on q -paths and k -shortest paths relaxations, *Ann. Oper. Res.* 61 (1995) 21–43.
- [14] C. Hjorring, J. Holt, New optimality cuts for a single-vehicle stochastic routing problem, *Ann. Oper. Res.* 86 (1999) 569–584.
- [15] D.J. Houck, J.C. Picard, M. Queyranne, R.R. Vemuganti, The travelling salesman problem as a constrained shortest path problem: theory and computational experience, *Opsearch* 17 (1980) 93–109.
- [16] G. Laporte, F.V. Louveaux, The integer L-shaped method for stochastic integer programs with complete recourse, *Oper. Res. Lett.* 13 (1993) 133–142.
- [17] G. Laporte, F.V. Louveaux, Solving stochastic routing problems with the integer L-shaped method, in: T.G. Crainic, G. Laporte (Eds.), *Fleet Management and Logistics*, Kluwer Academic Publishers, USA, 1998, pp. 159–167, (Chapter 7).
- [18] G. Laporte, F.V. Louveaux, L. Van Hamme, An integer L-shaped algorithm for the capacitated vehicle routing problem with stochastic demands, *Oper. Res.* 50 (2002) 415–423.
- [19] in: A.M. Law, W.D. Kelton (Eds.), *Simulation Modeling and Analysis*, McGraw-Hill, Singapore, 2000.
- [20] F.A. Tillman, The multiple terminal delivery problem with probabilistic demands, *Transportation Sci.* 3 (1969) 192–204.