Journal of the Operational Research Society (2011), 1–13 0160-5682/11

www.palgrave-journals.com/jors/

Robust vehicle routing problem with deadlines and travel time/demand uncertainty

C Lee[1], K Lee[2] and S Park[1]*

[1]ETRI, Daejeon, Republic of Korea; and [2]Hankuk University of Foreign Studies, Gyeonggi-do, Republic of Korea

In this article, we investigate the vehicle routing problem with deadlines, whose goal is to satisfy the requirements of a given number of customers with minimum travel distances while respecting both of the deadlines of the customers and vehicle capacity. It is assumed that the travel time between any two customers and the demands of the customer are uncertain. Two types of uncertainty sets with adjustable parameters are considered for the possible realizations of travel time and demand. The robustness of a solution against the uncertain data can be achieved by making the solution feasible for any travel time and demand defined in the uncertainty sets. We propose a Dantzig-Wolfe decomposition approach, which enables the uncertainty of the data to be encapsulated in the column generation subproblem. A dynamic programming algorithm is proposed to solve the subproblem with data uncertainty. The results of computational experiments involving two well-known test problems show that the robustness of the solution can be greatly improved.

Journal of the Operational Research Society advance online publication, 7 December 2011
doi:10.1057/jors.2011.136

Keywords: vehicle routing; robust optimization; column generation

Introduction

The vehicle routing problem (VRP) calls for the determination of the optimal set of routes to be performed by a fleet of vehicles to serve a given set of customers (Toth and Vigo, 2002). It can be considered to be one of the more important problems—both in theory and practice—in the fields of transportation, distribution, and logistics. Generally speaking, the goal of VRP is to find the minimum cost routes visiting (or serving) a given number of customers while respecting various resource constraints, including deadlines, vehicle capacities, and number of available vehicles. Several variations of the VRP exist. In the VRP with deadlines (VRPD), a deadline is imposed on each customer, with the requirement that the service to the customers must be provided before the deadline. Each customer has a given amount of demand and is visited by a vehicle exactly once. Each vehicle should have sufficient capacity to serve the demands of all of the customers it visits/serves.

Since the VRPD is a generalization of the VRP, which is a generalization of the TSP (travelling salesman problem), VRPD is $\mathcal{NP}$-hard. The VRPD is a special case of the VRP with time windows (VRPTW). Therefore, any solution algorithm for the VRPTW can be used to solve the VRPD. Kolen *et al* (1987) presented an optimization method for the VRPTW using a dynamic programming approach. The problem size was, however, limited to 15 customers. Fisher *et al* (1997) and Kohl and Madsen (1997) proposed Lagrangian relaxation approaches which were efficient for solving problems involving up to 100 customers. Ioachim *et al* (1998) developed a dynamic programming algorithm for the shortest path problem with time windows and linear node costs. At the present time, the most dominant and widely studied optimization approach for the VRPTW is the column generation-based approach. This approach was first presented by Desrochers (1986), who applied it to problems with as many as 100 customers. Fukasawa *et al* (2006) developed a branch-and-price-cut algorithm for the capacitated VRP (CVRP) by incorporating the cutting plain method into the branch-and-price algorithm. Many heuristic methods have also been developed in the context of the VRP, including the local search method (Lin and Kernighan, 1973; Dethloff, 2002), the tabu-search algorithm (Gendreau *et al*, 1994, 1996; Rego, 1998; Cordeau *et al*, 2002), the genetic algorithm (Berger and Barkaoui, 2003), and the ant colony algorithm (Dorigo *et al*, 1996; Yu *et al*, 2010).

*Correspondence: *S Park, Department of Industrial Engineering, KAIST, 291 Daehak-ro, Yuseong-gu, Daejeon 305-701, Republic of Korea.*
E-mail: sspark@kaist.ac.kr

Although there have been many advances in the optimization method for the VRP, most studies have not considered the uncertainty of the data. In practice, however, uncertainty in customer demand and/or travel time is inevitable. The feasibility of the solution obtained may not be guaranteed unless the uncertainty is incorporated directly in the optimization methodology. The *robust optimization* methodology deals directly with the *robustness* of the solution by finding a solution which is immune to variations in the data. The robust optimization approach differs from that of the stochastic optimization in that with the former it is not required to know the probability distribution of uncertain data *a priori*. In many cases, it may be very difficult—or even impossible—to estimate fairly accurate probability distributions of the data. In the robust optimization, instead of estimating the probability distributions, an uncertainty set is introduced to control the robustness of the solution; see Bertsimas and Sim (2004) for details.

Here, we propose an optimization method for the VRPD under travel time and customer demand uncertainty. In terms of the VRP with data uncertainty, the most studied problem may be the VRP with stochastic demands (VRPSD) (Gendreau *et al*, 1995, 1996a, b; Christiansen and Lysgaard, 2007; Tan *et al*, 2007). Bertsimas and Simchi-Levi (1996) developed a set of analytical results for the VRP with random demands and proposed several heuristic algorithms. In the VRPSD, the customer demands are assumed to be uncertain and to have stochastic properties, characterized by certain probability distributions. A number of studies on the travel time uncertainty in the VRP have used the stochastic programming approach to handle the uncertainty of travel time (Jula *et al*, 2006; Chang *et al*, 2009). The basic assumption of such studies is that the stochastic properties of travel time are known in advance; the goal is to obtain a solution with the best expected cost (distance or travel time), while the solution guarantees certain service levels. The functions of the expected arrival times (and/or penalties) at the customers' locations are defined and a number of methods for efficiently calculating the expected arrival time proposed. Since the exact calculation of the expected arrival times is often quite complicated, in many cases the calculation is done approximately (Jula *et al*, 2006; Chang *et al*, 2009) or heuristically (Cheung and Hang, 2003). Russell and Urban (2008) derived closed-form expressions of a penalty function for the Erlang travel times and developed a tabu-search-based algorithm. A number of papers have appeared on handling the recent changes in data in which the problem was iteratively solved to reflect these recent changes. To hedge for possible future changes, certain expected costs were calculated at each stage, so that the routes would be iteratively updated based on the best expected costs. Hvattum *et al* (2006) assumed that the presence of each customer is uncertain—a customer might place orders, or not, at any time during the planning horizon. They developed a multi-stage heuristic method to construct the routes gradually by considering future customer demands defined by pre-estimated probability distributions. Campbell and Thomas (2008, 2009) considered a similar problem but with customer deadlines. They proposed several different models to measure the penalty function with violations of the deadlines. In their settings, however, the uncertainties originate from the stochastic presence of the customers, not from the travel times. The stochastic approach is limited to cases in which the stochastic properties of uncertainty can be measured precisely, which may be exceedingly problematic, especially when data are scarce. Moreover, in many cases, the service level is expressed by the nonlinear (and nonconvex) chance constrained model, which can make the problem hard to solve.

Defining the uncertainty set in the robust optimization approach has a number of practical advantages over estimating probability distributions in the stochastic optimization approach. Firstly, in many cases, it is easier to define the uncertainty set than to estimate the probability distributions. For example, we simply can take all past realizations of data as the uncertainty set. Secondly, under certain conditions, the robust approach does not significantly escalate the complexity of the problem. For example, Bertsimas and Sim (2004) demonstrate that the robust counterpart problem of a linear programming problem is also a linear programming problem with a polynomially bounded problem size. Lastly, even though we already have some knowledge of the probability distributions, the uncertainty set can be easily constructed from these. For example, the confidence intervals of random data can be used as the intervals of uncertain data in the uncertainty set. Nevertheless, there have been only a few studies on the robust approach to the VRP. This scarcity of research on this topic may be due to (1) the VRP on its own being a hard problem and (2) the existing solution approaches for the deterministic problem being no longer valid for the robust version of the problem.

The contributions of this paper to research on the VRP are as follows: (1) proposal of a robust optimization approach to the VRPD that produces robust solutions under the uncertainty of travel time and demand; (2) demonstration that the robust version of the problem can be solved by the well-known branch-and-price algorithm, while consideration of the uncertainty is solely encapsulated in the column generation subproblem; (3) proposal of a new uncertainty-aware dominance rule for the labelling algorithm that enables the column generation subproblem to be solved efficiently; (4) reporting of an extensive computational study, which shows that in many cases the gains in robustness are rather large with small penalties in the objective values.

Problem description and formulation

In this section, a mathematical formulation for the case with no uncertainty (deterministic case) is presented. We then extend the formulation to the case of uncertain data. The problem is defined with the following parameters:

N	$\{1, \ldots, n\}$, set of customers
N_0	$N \cup \{0, n+1\}$, where 0 and $n+1$ are depots
M	$\{1, \ldots, m\}$, set of vehicles
Q	capacity of a vehicle
r_j	demand of customer $i \in N$
A	$\{(i,j) \mid i,j \in N_0$ and $i \neq j\}$, set of arcs.
c_{ij}	travel distance from i to j, where $(i,j) \in A$
t_{ij}	travel time from i to j, where $(i,j) \in A$
b_i	deadline for delivery at customer i, where $i \in N$
x_{ij}^k	decision variable. 1 if vehicle k travels from i to j, and 0 otherwise
s_i^k	decision variable. Arrival time of vehicle k at customer i

Without any loss of generality, it can be assumed that the service time for customer i is included in the travel time t_{ij}. Any vehicle should depart from depot 0 and arrive at depot $n+1$ after visiting a subset of customers. The deadline of customer i can also be represented as time window $[0, b_i]$. We may assume that the deadlines for the depots 0 and $n+1$ are 0 and ∞, which are equivalent to time windows $[0, 0]$ and $[0, \infty]$, respectively, and implying that a vehicle departs depot 0 at time 0 and can arrive at depot $n+1$ at any time. The arrival time at customer i cannot be greater than deadline b_i. The travel cost c_{i0} for any customer i is very large in order to prevent the vehicle from returning to depot 0. Similarly, $c_{n+1,i}$ for any customer i is very large, to prevent the vehicle from leaving from depot $n+1$. The mathematical model has two types of decision variables. The first type of variable determines the routes of the vehicles, that is, visiting sequences of the customers. Let x_{ij}^k be 1 if vehicle k travels from i to j, and 0 otherwise. The route of vehicle k is determined by the variables x_{ij}^k, $\forall (i,j) \in A$. The second type of variable determines when a vehicle arrives at each customer. Let s_i^k be the arrival time of vehicle k at customer i. Obviously, s_i^k must be less than or equal to b_i if vehicle k visits i; it is meaningless otherwise. Since all vehicles depart from depot 0 at time 0, s_0^k is 0 for all $k \in M$. In addition, s_{n+1}^k is the arrival time of vehicle k at the depot $n+1$.

The (deterministic) VRPD can be stated as follows:

$$\text{(VRPD)} \min \sum_{k \in M} \sum_{i \in N_0} \sum_{j \in N_0} c_{ij} x_{ij}^k \quad (1)$$

$$\text{subject to} \sum_{k \in M} \sum_{j \in N} x_{ij}^k = 1 \quad \forall i \in N, \quad (2)$$

$$\sum_{j \in N_0} x_{0j}^k = 1 \quad \forall k \in M, \quad (3)$$

$$\sum_{j \in N_0} x_{ij}^k - \sum_{j \in N_0} x_{ji}^k = 0 \quad \forall i \in N, k \in M, \quad (4)$$

$$\sum_{i \in N_0} x_{i,n+1}^k = 1 \quad \forall k \in M, \quad (5)$$

$$\sum_{i \in N} r_i \sum_{j \in N_0} x_{ij}^k \leqslant Q \quad \forall k \in M, \quad (6)$$

$$s_i^k + t_{ij} - K(1 - x_{ij}^k) \leqslant s_j^k \quad \forall i,j \in N_0, k \in M, \quad (7)$$

$$0 \leqslant s_i^k \leqslant b_i \quad \forall i \in N_0, k \in M, \quad (8)$$

$$x_{ij}^k \in \{0, 1\} \quad \forall i,j \in N_0, k \in M, \quad (9)$$

where K is a large number. The objective function is the sum of all routing distances of every vehicle. Constraints (2) ensure that each customer is served by exactly one vehicle. Constraints (6) are the capacity constraints for the vehicles; constraints (3), (4), and (5) are the flow conservation constraints, which ensure that each vehicle′s route should start from depot 0 and end at depot $n+1$; constraints (7) and (8) together guarantee that each customer is served before the deadline.

Robust VRPD

When a real-world logistics company uses the VRPD for designing its operational planning system, it must compile VRPD data, such as customer demands, vehicle capacities, deadlines, and vehicle travel times and distances. It is common practice that data for deadlines and vehicle capacities are *specified* or *given*, while those for travel times and customer demands are *estimated* or *forecasted.* In general, requested or specified data can be considered to be more accurate than estimated data. Moreover, in terms of the graph of all customers and depots, one should estimate the travel times for all pairs of nodes, since the graph is complete. Even when the estimations are derived via statistical methods, the estimated *nominal* values may be poor representations of true values when the variances in the data are considerably large.

There are a number of published studies on the VRP with travel time uncertainty. Jula *et al* (2006) considered a nonstationary stochastic travelling salesman problem. They call a route *acceptable* if the probability of visiting every node on a route before its deadline time is greater than a given constant, which is called the service level at the node. Given the probability distribution of travel times, they propose a simplified way to calculate the expected arrival time at each node of a route by approximating the expected arrival times. Although their approach does not exploit the probability distribution directly, the approximating procedure depends on the probability distribution assumption, requiring that the probability distributions be precisely determined, which may be a hard task to accomplish.

The aim of the robust optimization is to obtain the solution that is feasible for all realizations of uncertain data. In this approach, the probability distributions for uncertain data are assumed to be unknown, and only the *nominal* and *maximum possible deviation* values are specified. The uncertainty of data is represented by the uncertainty set, which contains all possible realizations of random data. To obtain a robust—but not too conservative—solution, it is necessary to introduce some parameters to control for the degree of robustness (the reader is referred to Bertsimas *et al* (2004) for details). Sungur *et al* (2008) considered robust capacitated VRP (CVRP) with uncertain demand. They modified the original CVRP formulation to incorporate the demand uncertainty and solved the problem directly by an off-the-shelf mixed integer programming (MIP) solver. Their results demonstrate that the robust optimization approach is attractive as it produces a much more robust solution with only a small penalty in the objective value.

In terms of the travel time uncertainty, it is highly unlikely that every segment on a route is delayed; in fact, it is much more likely that some segments are delayed while others are not. This observation indicates that we may restrict the number of delayed segments on a route so that we can control how much of the route should be robust. In other words, we want to protect the routes of vehicles against the given number of delays in the travel time, which yields the following definition of the uncertainty set of travel time.

Definition 1. Model of Travel time Uncertainty set U_t. For each arc $(i,j)\in A$, the travel time takes values in $[\hat{t}_{ij}, \hat{t}_{ij}+d_{ij}]$, where d_{ij} represents the maximum deviation from the nominal travel time $\hat{t}_{ij}$. We introduce a nonnegative integer Γ as a parameter for controlling the degree of robustness for the travel time uncertainties. Then, the uncertainty set of travel time data is given as

$$U_t = \left\{ \tilde{t} \in R^{|A|} | \tilde{t}_{ij} = \hat{t}_{ij} + d_{ij} v_{ij}, \quad \sum_{(i,j)\in A} v_{ij} \leqslant \Gamma, 0 \leqslant v_{ij} \leqslant 1 \forall (i,j) \in A \right\}.$$

Similarly, we define the uncertainty set of demand as follows.

Definition 2. Model of Demand Uncertainty set U_r. For each customer $i\in N$, the demand takes values in $[\hat{r}_i, \hat{r}_i+o_i]$; where o_i represents the maximum deviation from the nominal demand value $\hat{r}_i$. We introduce a nonnegative integer Λ as a parameter for controlling the degree of robustness for the demand uncertainties. Then, the uncertainty set of demand data is given as

$$U_r = \left\{ \tilde{r} \in R^{|N|} | \tilde{r}_i = \hat{r}_i + o_i w_i, \quad \sum_{i\in N} w_i \leqslant \Lambda, 0 \leqslant w_i \leqslant 1, \ \forall i \in N \right\}.$$

The robust version of VRPD can then be formulated as follows:

$$\text{(RCVRPD)} \ \min \sum_{k\in M} \sum_{i\in N_0} \sum_{j\in N_0} c_{ij} x_{ij}^k \tag{10}$$

subject to (2), (3), (4), (5)

$$\sum_{i\in N} r_i \sum_{j\in N_0} x_{ij}^k + \Lambda z_k + \sum_{i\in N} p_k^i \leqslant Q \quad \forall k \in M, \tag{11}$$

$$z_k + p_k^i \geqslant o_i \sum_{j\in N_0} x_{ij}^k \quad \forall k \in M, i \in N, \tag{12}$$

$$(s_i^k)^B + \hat{t}_{ij} + d_{ij} u_{ij}^B - K(1 - x_{ij}^k) \leqslant (s_j^k)^B \quad \forall (i,j) \in A, k \in M, \forall B \subseteq A, |B| \leqslant \Gamma, \tag{13}$$

$$0 \leqslant (s_i^k)^B \leqslant b_i \quad \forall i \in N_0, k \in M, \forall B \subseteq A, |B| \leqslant \Gamma, \tag{14}$$

$$p_k^i \geqslant 0 \quad \forall i \in N, k \in M, \tag{15}$$

$$z_k \geqslant 0 \quad \forall k \in M, \tag{16}$$

$$x_{ij}^k \in \{0,1\} \quad \forall (i,j) \in A, k \in M, \tag{17}$$

where u_{ij}^B is an indicator function which is 1 if $(i,j)\in B$, 0 otherwise. Constraints (11) and (12) and variables (15) and (16) are obtained from the reformulation of $\sum_{i\in N} r_i \sum_{j\in N_0} x_{ij}^k + \max_{S\subseteq N, |S|\leqslant\Lambda} \sum_{i\in S} o_i \sum_{j\in N_0} x_{ij}^k \leqslant Q$ (see Bertsimas and Sim, 2003). Note that a vehicle might arrive early or late at a certain customer location on the same vehicle route due to the uncertainty of the travel time. Therefore, we introduce additional variables, namely, $(s_i^k)^B$, $\forall B\subseteq A$, $|B|<\Gamma$, for different realizations of travel time. It should be noted that we assume the absence of uncertainty in the distance data c_{ij} and that we still try to minimize the sum of travel distances. The underlying motivation of this formulation is to distribute the risks of late arrivals over all of the routes as evenly as possible.

This can be done by distributing the timely tight visits to several different vehicles and making the vehicles visit those customers whose deadlines are tight at the early stages of the routes.

The large number of constraints (13) and (14) makes the formulation intractable. Therefore, instead of using the formulation (RCVRPD), which is based on the edge variables, we consider the path-based formulation using the Dantzig-Wolfe style decomposition scheme (Barnhart *et al*, 1998). We say that a route is robustly feasible if it remains feasible for all realizations of the uncertainty sets U_t and U_r. Formally speaking, a robustly feasible route is a path $(0, i_1, \ldots, i_l, n+1)$ which meets the deadline and capacity constraints at each customer location, while most Γ (travel times) and Λ (demands) of uncertain data can be at their maximum deviations, where $i_1, \ldots, i_l \in N$. Let R denote the set of all robustly feasible routes. The path-based formulation for the robust VRPD can then be given as:

$$\text{(Path-RCVRPD)} \ \min \sum_{r \in R} c_r x_r \tag{18}$$

$$\text{subject to} \sum_{r \in R} \delta_{ir} x_r \geqslant 1 \quad \forall i \in N, \tag{19}$$

$$\sum_{r \in R} x_r \leqslant m, \tag{20}$$

$$x_r \in \{0, 1\} \quad \forall r \in R, \tag{21}$$

where δ_{ir} is the number of visits to customer i in route r, m is the number of vehicles, and c_r is the travel distance of route $r \in R$, defined as the sum of the distances of the arcs of the route. Since the number of robustly feasible routes of R can be exponentially large, we use a column generation method. In the column generation method, the linear relaxation of the above set covering model with a restricte d set of routes is solved, and the *column generation subproblem* is solved to find a column which has a negative reduced cost. When no column has a negative reduced cost, the column generation procedure is terminated. One advantage of (Path-RCVRPD) is that the same deterministic column generation method can be used as long as the subproblem correctly identifies robustly feasible routes with negative reduced costs. In the following section, we present an algorithm for finding those robustly feasible routes with negative reduced costs.

Solution methodology

Let $R' \subset R$ be the restricted set of robustly feasible routes, and (RM), which is the restricted master problem, denote the linear relaxation of this problem after R has been replaced with R'.

$$\text{(RM)} \ \min \sum_{r \in R'} c_r x_r \tag{22}$$

$$\text{subject to} \sum_{r \in R'} \delta_{ir} x_r \geqslant 1 \quad \forall i \in N, \tag{23}$$

$$-\sum_{r \in R'} x_r \geqslant -m, \tag{24}$$

$$x_r \geqslant 0 \quad \forall r \in R'. \tag{25}$$

Note that we do not need $x_r \leqslant 1$ constraints, since at optimality, x_r cannot be greater than one because we minimize the objective function.

Column generation subproblem

Using the restricted set of routes $R' \subset R$, we now attempt to find the new routes—columns—entering (RM) by *pricing* their reduced costs. Let π_i and π_0 denote the dual variables associated with constraint i of (23) and (24), respectively. Based on the linear programming theory, the reduced cost of route r is given as follows:

$$rc(r) = c_r - \sum_{i \in N} \delta_{ir} \pi_i + \pi_0. \tag{26}$$

Finding routes with negative reduced cost is thereby reduced to finding the shortest route with the following arc cost and respecting the deadline and vehicle capacity constraints.

$$\bar{c}_{0,i} = c_{0,i} - \pi_i \quad \forall i \in N, \tag{27}$$

$$\bar{c}_{ij} = c_{ij} - \pi_j \quad \forall i \in N, j \in N, \tag{28}$$

$$\bar{c}_{i,n+1} = c_{i,n+1} + \pi_0 \quad \forall i \in N, \tag{29}$$

$$\bar{c}_{0,n+1} = 0. \tag{30}$$

The travel time and vehicle capacity can be generalized as the *resources* consumed (or accumulated) in the route. When a vehicle visits customer j, the resource of vehicle capacity is consumed by the amount of the demand r_j. Similarly, when a vehicle moves to the location of customer j from that of customer i, the resource of time is accumulated by the travel time t_{ij}. The vehicle can move to j if the consumed (or accumulated) resources are not greater than the resource constraints at customer j. Generally speaking, more resources can be defined, and the shortest path problem respecting the resource constraints is often referred to as the *shortest path problem with resource constraints* (SPPRC) (Irnich and Desaulniers, 2004). Here we introduce our robust version of SPPRC (RSPPRC). In RSPPRC, the amount of resource to be used is uncertain, and possible resource usages are defined by the uncertainty sets. A path is robustly feasible if—and only if—all of the resource constraints at every customer are satisfied at all realizations of the resource usages defined by the uncertainty sets.

We consider a graph G whose nodes are customers (N) and depots $(0, n+1)$, and (directed) arcs have costs of (27), (28), (29), and (30). In the standard algorithm for the SPPRC, each possible partial path is associated with a label, which represents the consumption (or accumulation) of the resources of the partial path. At the *extending* stage, all new partial paths are extended toward every possible successor node. At the *elimination* stage, a label is eliminated if it is *dominated* by some other label. A brief description of Desrochers' labelling algorithm and our adaptation to the RSPPRC are given in the following section.

Desrochers' labelling algorithm

For a partial path p which ends at node i, we associate a label $E_p = [c_p, R_p^1, R_p^2, \ldots, R_p^L]$ with the path, where L is the number of resources. Let c_p be the cost of path p. The resources can include vehicle capacity, travel time, and distance, among others. Here R_p^l represents the accumulated value of resource/at the last node of path p. Let $v(p)$ denote the last node of path p, that is $v(p) = i$. A path q is a *feasible extension* of p if the path $(p, q) \equiv \{0, p_1, p_2, \ldots, v(p), q_1, q_2, \ldots, v(q)\}$ satisfies all resource constraints at every node in the path. Let $\mathcal{E}(p)$ be the set of all feasible extensions of p. The nonnegative value r_{ij}^l is defined as the usage value of needed resource l when we travel from i to j, that is $R_p^l = \sum_{(i,j) \in A(p)} r_{ij}^l$, where $A(p)$ is the set of arcs of path p. Let p and q be two distinct paths from 0 to i with associated label E_p and E_q, respectively. We say p *dominates* q if and only if $c_p \leqslant c_q$, $R_p^l \leqslant R_q^l$, for all $l = 1, \ldots, L$. The domination is *strict* if $c_p < c_q$ or there exists l such that $R_p^l < R_q^l$. We denote these by $E_p < E_q$ and $E_p < E_q$, respectively. According to these definitions, the *dominance rule* is established as follows:

Definition 3. Dominance Rule for SPPRC. Given two distinct paths p and q such that $v(p) = v(q)$, the path q can be discarded if p strictly dominates q, that is, $E_p < E_q$. Either p or q can be discarded if they dominate each other, that is, $E_p \leqslant E_q$ and $E_p \geqslant E_q$.

The validity of the dominance rule can be easily demonstrated. When path p dominates q, clearly $\mathcal{E}(p) \supseteq \mathcal{E}(q)$ and for any $q^+ \in \mathcal{E}(q)$, there is a $p^+ \in \mathcal{E}(p)$ such that (p, p^+) dominates (q, q^+). The labelling algorithm is an iterative process of the *path extending procedure* (Extending) and the *label elimination procedure* (Eliminate). Note that the algorithm is quite general, in the sense that many types of resource constraints can be modelled within this framework. This algorithm allows multiple visits to a node, so the resulting path may contain cycles.

Uncertain resources case

The dominance rule of Definition 3 is no longer valid when the resource usage value r_{ij}^l is uncertain. There are two types of uncertainty sets, namely, U_t and U_r, in our setting. From now on, we only consider the travel-time uncertainty set U_t, since the following line of reasoning on U_t can be readily extended to demand uncertainty set U_r. For any resource l, we apply the definition of travel-time uncertainty set U_t (Definition 1). For a path p and given Γ, let $\tilde{R}_p^l = \sum_{(i,j) \in A(p)} \hat{r}_{ij}^l + \max_{\{S \subset A(p) | |S| \leqslant \Gamma\}} \sum_{(i,j) \in S} d_{ij}^l$ where $\hat{r}_{ij}^l$ is the nominal value of the consumption of resource l when we move from node i to j, and d_{ij}^l is the value of the maximum deviation. A path $p = (0, p_1, p_2, \ldots, p_o, n+1)$ is robustly feasible if and only if, for all $l \in \{1, \ldots, L\}$, $\tilde{R}_{p^k}^l$ satisfies the resource constraints for all partial paths $p^k = (0, p_1, p_2, \ldots, p_k)$, $\forall k \in \{0, 1, \ldots, o, n+1\}$.

Without any loss of generality, we assume that the resource 1 is subject to uncertainty and the degree of robustness Γ is given. For a partial path p, such that $v(p) = i$, an associated label is defined as follows:

$$\tilde{E}_p = \left[c_p, \hat{R}_p^1, \overbrace{D_p^1, \ldots, D_p^2, \ldots, D_p^\Gamma}^{\Gamma}, R_p^2, \ldots, R_p^L \right],$$

where $D_p^k = \max_{\{S \subset A(p) | |S| \leqslant k} \sum_{(i,j) \in S} d_{ij}^1$, $k = 1, \ldots, \Gamma$, that is, the sum of k largest d_{ij}^1s on path p, $\hat{R}_p^1 = \sum_{(i,j) \in A(p)} \hat{r}_{ij}^1$ and L the number of resources.

Proposition 1. For two distinct paths p and q, such that $v(p) = v(q)$, $\tilde{E}_p \leqslant \tilde{E}_q$ if and only if p dominates q.

Proof For the sufficient condition ($\tilde{E}_p \leqslant \tilde{E}_q \Rightarrow p$ dominates q), we have to show that $\mathcal{RE}(p) \supseteq \mathcal{RE}(q)$ when $\tilde{E}_p \leqslant \tilde{E}_q$, where $\mathcal{RE}(p)$ is the set of all *robustly* feasible extensions of p. Consider a robustly feasible path $Q = (q, q^+)$. By definition, $q^+ \in \mathcal{RE}(q)$. Let $A^+ = A(Q) \backslash A(q)$ and $S_k^A = \{S \subset A | |S| \leqslant k\}$, then it is easily seen that the following holds:

$$\max_{S \in S_\Gamma^{A(Q)}} \sum_{(i,j) \in S} d_{ij}^1 = \max_{\substack{k=0,\ldots,\Gamma, \\ S \in S_k^{A(q)} \cup S_{\Gamma-k}^{A+}}} \sum_{(i,j) \in S} d_{ij}^1 = \max_{k=0,\ldots,\Gamma} \max_{S \in S_k^{A(q)} \cup S_{\Gamma-k}^{A+}} \sum_{(i,j) \in S} d_{ij}^1 = \max_{k=0,\ldots,\Gamma} (D_q^k + D_{q^+}^{\Gamma-k}).$$

Now consider a path P constructed as follows:

$$P = \overbrace{(0, p_1, p_2, \ldots, v(p)}^{P}, \underbrace{q_1^+, q_2^+, \ldots, v(q^+))}_{q^+}.$$

By assumption, we have $\max_{k=0,\ldots,\Gamma}(D_p^k + D_{q^+}^{\Gamma-k}) \leqslant \max_{k=0,\ldots,\Gamma}(D_q^k + D_{q^+}^{\Gamma-k})$ and $\hat{R}_p^1 \leqslant \hat{R}_q^1$ which imply that P is also robustly feasible, that is, $q^+ \in \mathcal{RE}(p)$. So, $\mathcal{RE}(p) \supseteq \mathcal{RE}(q)$ and p dominates q.

For the necessary condition, assume p dominates q. Then we have $c_p \leqslant c_q$, $\hat{R}_p^1 \leqslant \hat{R}_q^1$, $D_p^\Gamma \leqslant D_q^\Gamma$, and $R_p^l \leqslant R_q^l$, $\forall l = 2, \ldots, L$; otherwise it easily derives a contradiction. Assume, for a contradiction, that there exists γ, such that $D_p^\gamma > D_q^\gamma$ and $1 \leqslant \gamma \leqslant \Gamma - 1$. By definition, $D_p^\gamma = d_p^{(1)} + d_p^{(2)} + \cdots + d_p^{(\gamma)}$, where $d_p^{(i)}$ is the ith largest deviation of uncertain resource 1 in path p. There then exists k, such that $d_p^{(k)} > d_q^{(k)}$ and $1 \leqslant k \leqslant \gamma$. Consider robustly feasible paths $P = (p, r)$ and $Q = (q, r)$, where Γ largest deviations are given as follows:

$$P : d_p^{(1)}, \ldots, d_p^{(k)}, \overbrace{d_r^{(1)}, \ldots, d_r^{(1)}}^{\Gamma-\gamma}, \ldots d_p^{(\gamma)}.$$

$$Q : d_q^{(1)}, \ldots, \overbrace{d_r^{(1)}, \ldots, d_r^{(1)}}^{\Gamma-\gamma}, \ldots, d_q^{(k)}, \cdots d_q^{(\gamma)},$$

with $d_p^{(k)} = d_r^{(1)}$. We then have $D_P^\Gamma > D_Q^\Gamma$, which implies $\mathcal{RE}(p) \not\supseteq \mathcal{RE}(\varsigma)$ and derives a contradiction. This completes the proof. □

Definition 4. Dominance Rule for RSPPRC. Given two distinct paths p and q such that $v(p) = v(q)$, path q can be discarded if p strictly dominates q, that is, $\tilde{E}_p < \tilde{E}_q$. Either p or q can be discarded if they dominate each other, that is, $\tilde{E}_p \leqslant \tilde{E}_q$ and $\tilde{E}_p \geqslant \tilde{E}_q$.

The modified dominance rule above allows the use of the deterministic SPPRC algorithm with a number of modifications. For a label of path p, a fixed size priority queue is used to store Γ largest deviations of the resource 1 on path p. In the procedure ELIMINATE, the dominance relation between any two labels $\tilde{E}_p$ and $\tilde{E}_q$ can be easily checked by comparing all components of the labels, including D_p^k and D_q^k for all $k = 1, \ldots, \Gamma$, which are the sums of k top elements in the priority queues. In the procedure EXTENDING, label $\tilde{E}^*_{(p,i)}$ is a feasible extension of label $\tilde{E}_p$, if $\hat{R}^1_{(p,i)} + D^\Gamma_{(p,i)} \leqslant b_i^1$ and $\hat{R}^l_{(p,i)} \leqslant b_i^l$, $l = 2, \ldots, L$, where b_i^l is the upper limit of resource l at customer i. See Algorithm 1 for details. When we associate VRPD with travel time and demand uncertainty, the resource windows for arrival time and vehicle load are $[0, b_i]$, $\forall i \in N$ and $[0, Q]$, respectively. This allows us to define the label for the robust VRPD. Under the given parameters Γ and Λ and with a partial path p such that $v(p) = i$, an associated label is defined as follows:

$$\tilde{E}_p = \left[c_p, \hat{t}_p, \overbrace{D_p^1, \ldots, D_p^2, \ldots, D_p^\Gamma}^{\Gamma}, \hat{r}_p \overbrace{O_p^1, \ldots, O_p^2, \ldots, O_p^\Lambda}^{\Lambda} \right],$$

where c_p is the sum of arc costs of path p. The sum of nominal travel times is $\hat{t}_p = \sum_{(i,j)\in A(p)} \hat{t}_{ij}$, and D_p^k is the sum of k largest travel time deviations d_{ij} on path p. For demand, $\hat{r}_p$ and O_p^k are defined in a similar manner.

Algorithm 1 Labeling algorithm for RSPPRC

```
procedure RSPPRC
    for i ∈ N_0 do                          ▷ Initialization
        Ω_i ← ∅                             ▷ Ω_i is the set of labels for node i
    end for
    Ω_0 ← [0, ..., 0]                       ▷ node 0 is the depot
    U ← {0}                                 ▷ All paths start from node 0
    repeat
        Select i ∈ U
        U ← U \ {i}
        for j ∈ {Successors of i} do
            for E ∈ Ω_i do
                E* ← EXTENDING(E, j)        ▷ Extending procedure
                if E* is feasible then      ▷ Check the resource constraints at node j
                    Ω_j ← Ω_j ∪ {E*}        ▷ Add newly created label
                end if
            end for
            Ω_j ← ELIMINATE(Ω_j)            ▷ Remove dominated labels by using Definition 4.
            U ← U ⋃ {j}
        end for
    until U = ∅∅
    return the path with smallest cost in Ω_{n+1}    ▷ Return shortest path
end procedure
```

Note that the algorithm permits negative cycles in the solution paths. In the deterministic case there is an integer optimal solution that does not contain cycles when certain mild conditions are met (Chabrier, 2006). In the robust case, however, the existence of an integer optimal solution without cycles cannot be guaranteed, since a shorter route may not be robustly feasible while a longer route with cycles is. We simply take branching when the optimal solution at a branch-and-price node gives integer paths with cycles, as branching excludes the integer paths with cycles from the search space.

Branching scheme

The column generation procedure is terminated when there are no routes with negative reduced cost. If the current optimal solution to (RM) is integer without a cycle, the solution is optimal. When the optimal solution to the current master problem is fractional or when a cycle is present in any route used in the solution, branching is required. We used a branching scheme based on the dichotomy of the arcs incident to the divergence node, which was originally proposed for the integer multi-commodity flow problem (Barnhart *et al*, 2000). We first find any positive-valued (either fractional or integral) route with cycles at the current solution. Let us suppose that route r is chosen; it can then be easily seen that there exist two arcs, namely, $(i^*, j^*) \in A(r)$ and $(i^*, k^*) \in A(r)$ sharing the same tail node i^*, which is the divergence node. Let $\delta^+(i^*) := \{(i,j) \in A | i = i^*\}$. We first construct two disjoint sets δ_1^+ and δ_2^+ such that $\delta_1^+ \bigcup \delta_2^+ = \delta^+(i^*)$, $(i^*, j^*) \in \delta_1^+$, and $(i^*, k^*) \in \delta_2^+$, and then we create two nodes in the branch-and-bound tree. In one node, the arcs in δ_1^+ are forbidden, while in the other node, the arcs in δ_2^+ are forbidden. Note that an arc can be forbidden readily by imposing a large penalty cost on the arc at the column generation subproblem. If all positive-valued routes are cycle-free, we find node i^*, which is shared by any two fractional-valued routes r_1 and r_2, and two distinct arcs $(j^*, i^*) \in A(r_1)$ and $(k^*, i^*) \in A(r_2)$ exist. We consider $\delta^-(i^*) := \{(j,i) \in A | i = i^*\}$ and construct two disjoint sets, δ_1^- and δ_2^-, such that $\delta_1^- \bigcup \delta_2^- = \delta^-(i^*)$, $(j^*, i^*) \in \delta_1^-$, and $(k^*, i^*) \in \delta_2^-$. In a similar manner, we create two nodes in the branch-and-bound tree. Arcs in δ_1^- (δ_2^-) are forbidden in one node (the other node) of the branch-and-bound tree.

Computational results

In this section, we present our computational results for our newly proposed algorithm for the robust VRPD. All of the computational tests presented here were performed on an AMD X2 2.9 GHz PC with 4GB RAM. The algorithm was implemented with C#, and CPLEX 10.1 was used as the linear programming solver.

Our computational experiments were performed on two sets of problems. The first set was taken from the well-known Solomon problems (Solomon, 1987). In this case, we used R and RC problems, which have 25 randomly distributed (R) or randomly clustered (RC) customers, respectively. Because the given vehicle capacity of Solomon problems is too large to be meaningful in terms of capacity constraints, we reset the demands of customers to be twofold of the original values for the R problems (1.5 fold those for the RC problems). For all Solomon problems, we used the close time of the time window as the deadline. The second set was taken from Augerat *et al* (1995). Since Augerat problems were originally designed for the CVRP (without deadlines), we assigned deadline times of 150 and 200 for every odd node and every even node, respectively.

Monte-Carlo simulation tests were designed to evaluate the robustness of the obtained solutions. The simulation model of random travel times is based on the following real-life observations. Firstly, a travel time between two distinct customers might be significantly delayed by unpredictable accidents. In other words, a probability distribution of the travel time might have a long tail. Secondly, a travel time might be shorter by no more than a certain limit, while it is possible to be delayed without limit. This assumption implicates the asymmetrical nature of the probability distribution of the travel time. The model of the probability distribution for the travel time is illustrated in Figure 1. This model is based on a normal distribution with nominal value $\hat{t}$ as the peak value (ie, mode value) and a low cut at $\hat{t}(1-\Delta)$, where Δ is a given simulation parameter. The parameter Δ is introduced to reflect the degree of the unreliability of the nominal travel time.

The travel times of two distinct arcs may be correlated. For example, as some customers may be located in highly congested areas, travel to (or from) these customers will require more (travel) time than to other customers. Therefore, we assume that the random value y_i follows the probability distribution defined in Figure 1 with $\hat{t} = 0$ and $\Delta \leftarrow \Delta / 2\sqrt{2}$. The travel time on arc (i,j) is defined as $t_{ij} + t_{ij}(y_i + y_j)$. For the demand uncertainty, we assume that the random demand of customer i follows the normal distribution $\mathcal{N}(\hat{r}_i, (\hat{r}_i \Sigma)^2)$, where Σ is a given simulation parameter.

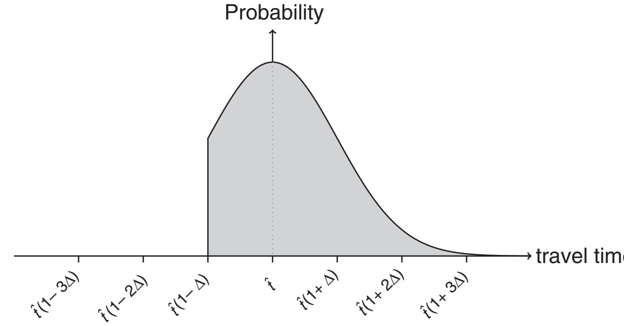

Figure 1 Probability distribution of travel time with $\hat{t}$ as the nominal value.

Comparison between the deterministic and robust solutions

For the given simulation parameters Δ and Σ, we generated 1000 scenarios which have random travel time matrices and demand vectors. The robustness of a solution is measured by testing how many of these scenarios remain feasible among the 1000 generated. The feasibility of a scenario can be checked easily by examining the deadlines and vehicle capacities of the solution's routes using the current scenario's travel time matrix and demand vector.

The computational results of the Solomon problems are summarized in Tables 1 and 2. The headings Γ and Λ refer to the degrees of robustness for the travel time and demand, which determine the uncertainty sets U_t and U_r, respectively. The asterisk (*) indicates that the time limit (one hour) was reached. The solutions with $\Gamma = 0$, $\Lambda = 0$ are a non-robust (deterministic) version of the problem and used for comparison with robust aware solutions having $\Gamma = 2$, $\Lambda = 2$. The total time spent in the branch-and price in seconds are shown under the headings `time`. Headings `opt` and `inc` denote the optimal integer solution (or the best incumbent solution) and the percentage increments in the optimal values due to the introduction of robustness of the solution, respectively. The bold numbers indicate that the optimal solutions were obtained within the time limit. Robustness of the (optimal or best) solutions measured from the Monte-Carlo simulation tests are reported under the heading `Simulation`. For each simulation case, 0.2 was used as the value of Δ and/or Σ. Note that the larger Δ (or Σ), the more uncertainty in the travel time data (or demands). For example, $\Delta = 0.2$ and $\Sigma = 0$ indicate that there is no demand uncertainty in the simulation scenarios. It is possible to investigate which kind of uncertainty makes the solution more risky in terms of data variations by comparing two simulation conditions, namely, $\Delta = 0.2$, $\Sigma = 0$ and $\Delta = 0$, $\Sigma = 0.2$.

The results clearly show the following. Firstly, the deterministic solutions are very frail, with the average robustness of the deterministic solutions being 10.26 and 26.96% for R class and RC class problems, respectively. Secondly, the robustness of solutions is significantly improved by the robust approach—up to 48.28 and 81.83% on average. Thirdly, RC class problems are much harder to solve. Since the customers are clustered in RC problems, there may be many feasible routes of similar

Table 1 Results for the Solomon R class instances. For all instances, we set $d_{ij} = \hat{t}_{ij} \times 0.2, \forall (i,j) \in A$ and $o_i = \hat{r}_i \times 0.2, \forall i \in N$. Problems: (Solomon, 1987)

prob	Γ, Λ	time	opt	inc (%)	Simulation (%): Δ, Σ		
					0.2, 0	0, 0.2	0.2, 0.2
R101	0, 0	1.32	**453.1**		56.4	46.8	26.2
	2, 2	3.62	**464.4**	2.49	94.5	84.3	79.5
R102	0, 0	705.53	**434.2**		17.1	9.5	1.7
	2, 2	201.09	**444.7**	2.42	95.3	76.6	73.1
R103	0, 0	2540.15	**407.5**		22.7	17.0	3.6
	2, 2	1440.65	**417.3**	2.40	99.8	52.7	52.5
R104	0, 0	2091.56	**404.4**		100.0	18.0	18.0
	2, 2	3517.91	**407.9**	0.87	99.9	55.2	55.1
R105	0, 0	47.79	**436.9**		10.3	13.5	1.7
	2, 2	41.14	**446.6**	2.22	98.6	60.2	59.0
R106	0, 0	360.01	**405.1**		18.4	13.7	2.6
	2, 2	465.82	**426.3**	5.23	97.3	36.2	34.6
R107	0, 0	1857.57	**394.3**		26.4	13.7	3.4
	2, 2	3600*	408.3	3.55	96.4	25.7	24.4
R108	0, 0	1797.56	**394.3**		26.4	13.7	3.4
	2, 2	836.41	**399.3**	1.27	96.4	25.7	24.4
R109	0, 0	167.78	**409.1**		79.8	28.9	24.0
	2, 2	1095.92	**429.4**	4.96	89.7	73.5	66.0
R110	0, 0	779.17	**401.7**		50.2	13.7	7.1
	2, 2	471.67	**409.0**	1.82	99.1	35.2	34.9
R111	0, 0	1024.42	**401.3**		71.7	17.0	11.0
	2, 2	3600*	417.3	3.99	100.0	52.7	52.7
R112	0, 0	3600*	399.9		99.8	20.4	20.4
	2, 2	2573.76	**399.3**	−0.15	92.5	25.7	23.1
Average	0, 0	1247.75	411.82		48.27	18.83	10.26
	2, 2	1487.34	422.48	2.59	96.63	50.31	48.28

Table 2 Results for the Solomon RC class instances. For all instances, we set $d_{ij} = \hat{t}_{ij} \times 0.2, \forall (i,j) \in A$ and $o_i = \hat{r}_i \times 0.2, \forall i \in N$. Problems: (Solomon, 1987).

prob	Γ, Λ	time	opt	inc (%)	Simulation (%): Δ, Σ		
					0.2, 0	0, 0.2	0.2, 0.2
RC101	0, 0	3600*	504.3		82.6	36.5	28.8
	2, 2	3600*	531.5	5.39	98.0	99.7	97.7
RC102	0, 0	3600*	493.7		44.5	20.5	7.9
	2, 2	3600*	522	5.73	97.7	88.3	86.2
RC103	0, 0	3600*	494.3		100.0	22.9	22.9
	2, 2	3600*	513.9	3.97	98.0	67.5	66.4
RC104	0, 0	3600*	489.8		100.0	47.2	47.2
	2, 2	3600*	513.1	4.76	94.4	70.7	67.2
RC105	0, 0	3600*	487.7		93.2	45.7	42.9
	2, 2	3600*	526.4	7.94	99.7	97.7	97.4
RC106	0, 0	3600*	506.9		63.8	40.2	25.4
	2, 2	3600*	525.8	3.73	98.9	87.1	86.0
RC107	0, 0	3600*	495.8		41.6	48.4	20.5
	2, 2	3600*	508.1	2.48	97.3	69.9	68.1
RC108	0, 0	3600*	491.9		64.9	31.9	20.0
	2, 2	3600*	526.5	7.03	100.0	85.6	85.6
Average	0, 0	3600*	495.55		73.83	36.66	26.95
	2, 2	3600*	520.91	5.13	98.00	83.31	81.83

distances, making the problem hard to solve. It should be noted that the robustness of our solutions are greatly improved for RC problems also, even when the optimal solutions could not be obtained. Fourthly, for the R112 problem in Table 1, the robust version has a better objective value than the deterministic version, primarily because the deterministic version has failed to obtain the optimal solution within the time limit. Finally, the increments of the objective values of the robust problems are larger in the RC problems. In the clustered networks, the route of any vehicle may be restricted within a cluster of customers to minimize the travel distance. Therefore, identifying a risk-averse route may require that the route visit another cluster, which results in a long route that visits multiple clusters.

Table 3 reports the computational results for the Augerat problems. The results are similar to those for the Solomon problems in that we see significant increments in the robustness of the solutions without much loss in solution quality. For example, for the simulation test in which the presence of travel time and demand uncertainties ($\Delta = \Sigma = 0.2$) is assumed, the robustness of the solutions improved by 63.8% on average.

A deterministic solution may be robust already if the resource restrictions are relaxed. As briefly stated earlier, Solomon problems have very relaxed capacity constraints. Figure 2 illustrates the changes in the robustness of the solution for problem R101 with increasing demands. It can be seen that the increased demands do not affect the robustness of the solution against the travel time uncertainty (Figure 2a). At the large demand multiplier ($\geqslant 2$), the deterministic solution becomes slightly more robust because the routes are shortened by the vehicle capacity restrictions. In the case when there is only the uncertainty of demand, shows that the capacity constraint imposes little restriction on the vehicle routes at the small demand level (Figure 2b). At the default demand level (demand multiplier = 1), it can be said that the risk associated with vehicle routes originates mainly from the travel time uncertainty—not from the demand uncertainty. This kind of inference may be useful to practitioners who want to solve the VRP for a practical purpose. For example, a logistics company can accept more unexpected customer demands in a delivery plan since the capacity of a vehicle has a sufficient immunity against the demand uncertainty.

It is fairly difficult to compare the performance of the algorithm with that of the direct reformulation (RCVRPD) because (RCVRPD) has too many constraints and variables. Based on our computational experience, even building the mathematical model for CPLEX requires too much time and memory space. Therefore, based on problem R101, we designed a small test problem involving the first 10 customers (107 arcs, and 25 vehicles). With uncertainty parameters $\Gamma = 1$, $\Lambda = 2$, $d = 0.2\hat{t}$, and $o = 0.2\hat{r}$, the branch-and-price algorithm solved the problem optimally in 0.3 s with a 7.8% linear programming (LP) relaxation gap. For comparison purposes, we solved (RCVRPD) using CPLEX and found that for this small problem the number of constraints (13) in (RCVRPD) was $107 \times 25 \times \binom{107}{1} = 286\,225$. After 10 h, CPLEX has failed to prove the optimality, and the remaining gap was 18.3% (the LP relaxation gap was 35%), mainly because of the large problem size. When $\Gamma = 2$, we received an out-of-memory error from CPLEX. In this case, the number of constraints (13) in (RCVRPD) was $107 \times 25 \times \binom{107}{2} = 15\,169\,925$, which was clearly too large to be solved directly by CPLEX.

Table 3 Results for the Augerat problems. For all instances, we set $d_{ij} = \hat{t}_{ij} \times 0.2, \forall (i,j) \in A$ and $o_i = \hat{r}_i \times 0.2, \forall i \in N$. Problems: (Augerat *et al*, 1995).

prob	Γ, Λ	time	opt	inc	Simulation (%): Δ, Σ		
					0.2, 0	0, 0.2	0.2, 0.2
A-n32-k5	0, 0	3600*	1039		38.1	60.0	23.2
	2, 2	3600*	1019	−1.92	96.6	92.2	89.3
A-n33-k5	0, 0	3600*	766		44.1	70.6	30.7
	2, 2	3600*	820	7.05	99.4	84.4	83.9
A-n33-k6	0, 0	3600*	757		40.1	12.8	5.4
	2, 2	3600*	854	12.81	99.7	72.9	72.8
A-n34-k5	0, 0	3600*	844		19.2	25.3	4.7
	2, 2	3600*	966	14.45	98.4	86.5	84.9
A-n36-k5	0, 0	3600*	953		8.9	57.5	5.1
	2, 2	3600*	989	3.78	91.6	88.9	81.5
B-n31-k5	0, 0	3600*	698		63.7	31.3	19.9
	2, 2	3600*	810	16.05	98.9	90.1	89.1
B-n34-k5	0, 0	3600*	860		29.0	19.3	5.6
	2, 2	3600*	991	15.23	99.6	74.4	74.0
B-n35-k5	0, 0	3600*	1250		15.0	63.2	9.4
	2, 2	3600*	1379	10.32	96.6	91.8	88.8
B-n38-k6	0, 0	3600*	1008		20.3	15.4	2.9
	2, 2	3600*	1104	9.52	94.8	71.0	67.3
B-n39-k5	0, 0	3600*	643		37.9	67.3	25.2
	2, 2	3600*	837	30.17	97.8	93.5	91.5
P-n19-k2	0, 0	3600*	234		53.5	45.1	25.0
	2, 2	3600*	252	7.69	99.8	99.6	99.4
P-n20-k2	0, 0	3600*	261		100.0	99.9	99.9
	2, 2	3600*	251	−3.83	100.0	99.7	99.7
P-n22-k8	0, 0	1.836	590		100.0	11.4	11.4
	2, 2	22.217	666	12.88	100.0	66.0	66.0
P-n23-k8	0, 0	22.083	529		100.0	0.8	0.8
	2, 2	22.879	618	16.82	100.0	62.2	62.2
P-n40-k5	0, 0	3600*	528		38.8	49.2	19.7
	2, 2	3600*	548	3.79	98.2	98.6	96.8
Average	0, 0	3121.59	730.67		47.2	41.9	19.3
	2, 2	3123.01	806.93	10.32	98.1	84.8	83.1

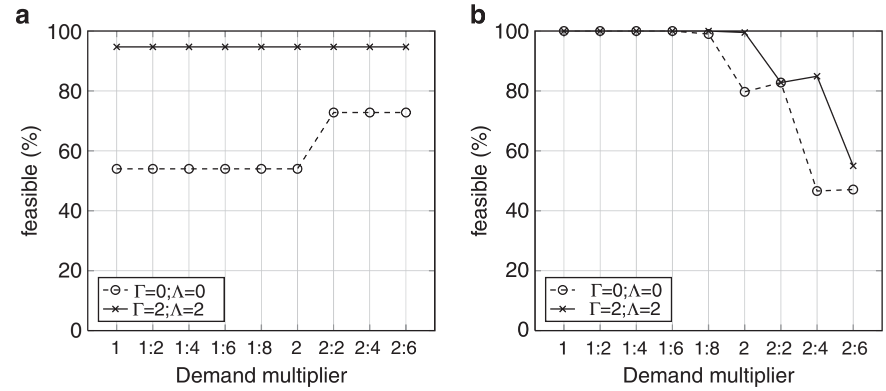

Figure 2 Percentages of feasible scenarios on different demand sizes for the problem R101. $d_{ij} = \hat{t}_{ij} \times 0.2, \forall(i,j) \in A$ and $o_i = \hat{r}_i \times 0.2, \forall i \in N$. Problems: (Solomon, 1987). (a) Simulation of travel time uncertainties only ($\Delta = 0.2$, $\Sigma = 0$). (b) Simulation of demand uncertainties only ($\Delta = 0.2$, $\Sigma = 0.2$).

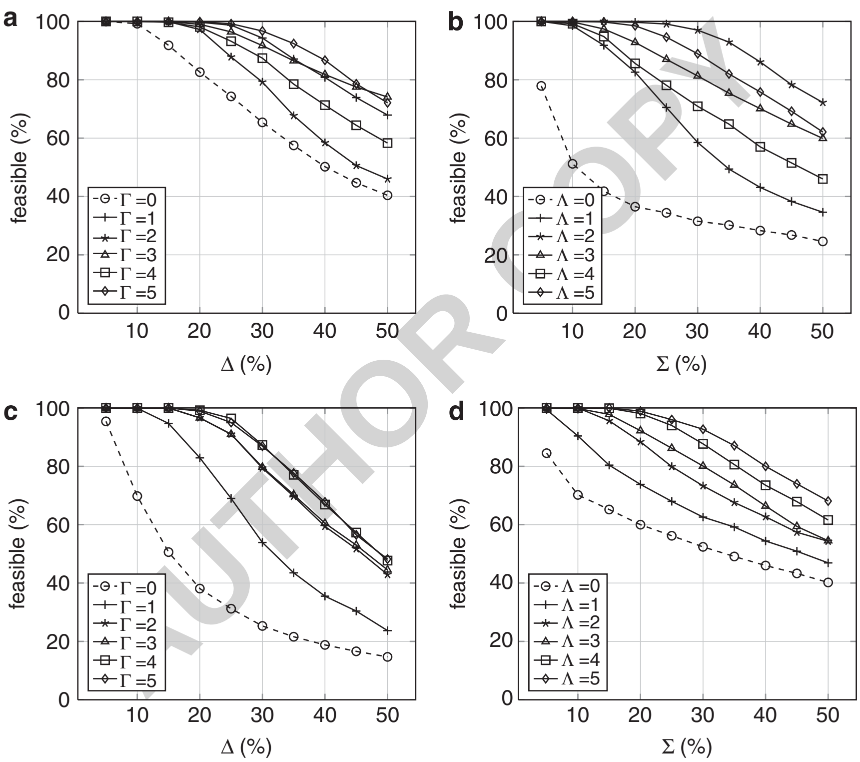

Figure 3 Percentages of feasible scenarios according to different uncertainty levels, $d_{ij} = \hat{t}_{ij} \times 0.2, \forall(i,j) \in A$ and $o_i = \hat{r}_i \times 0.2, \forall i \in N$. Problems: (Solomon, 1987) and (Augerat *et al*, 1995). (a) Problem RC101. Simulation of travel time uncertainties only ($\Lambda = 0$, $\Sigma = 0$). (b) Problem RC101. Simulation of demand uncertainties only ($\Gamma = 0$, $\Delta = 0$). (c) Problem A-n35-k5. Simulation of travel time uncertainties only ($\Lambda = 0$, $\Sigma = 0$). (d) Problem A-n35-k5. Simulation of travel time uncertainties only ($\Gamma = 0$, $\Delta = 0$).

Analysis of robustness for different uncertainty parameters

The results on the robustness of the solutions for different uncertainty parameters in terms of the percentages of feasible scenarios are plotted in Figure 3. The simulation was conducted in two ways. In the first simulation there is no uncertainty in terms of demand data ($\Sigma = 0$). Changes in the percentages of feasible scenarios with increasing travel time uncertainty (increasing Δ) are plotted in Figure 3a and 3c. In the second scenario, there is no uncertainty in terms of the travel times ($\Delta = 0$). Changes in the number of feasible scenarios with increasing demand uncertainty (increasing Σ) are plotted in Figure 3b and 3d. It can be clearly seen that the deterministic solutions become rapidly unreliable as the data become more uncertain. The robustness-aware solutions are, however, far more robust against the uncertainties. It was interesting to note that the deterministic solutions deteriorate more rapidly for the cases presented in Figure 3b and 3c. Closer examination of these solutions reveals that the deterministic solution for problem RC101 has one time-risky route and two capacity-risky routes, while the deterministic solution for A-n35-k5 has two time-risky routes and one capacity-risky route. To be feasible, the multiple risky-routes should meet joint-probability conditions, which enforces a rapid decrease in feasibility—the solution is feasible if all of the risky routes are safe against the uncertainties.

Conclusions

In this article, we have considered the case of the robust vehicle routing problem with deadlines (RVRPD). We have assumed that there is uncertainty in the travel time and demand data, which implies that the feasibility of ordinary deterministic solutions cannot be guaranteed. The goal of our approach was to obtain a more robust solution with only a small penalty in the objective value. Based on the definition of the uncertainty set of Bertsimas and Sim (2004), we have shown that the travel time and demand uncertainty of the problem can be encapsulated in the column generation subproblem, which is defined as the problem of finding robustly feasible routes with negative reduced costs. We have also been successful in modelling the subproblem as the robust shortest path problem with resource constraints (RSPPRC) and have proposed a dynamic programming solution algorithm to solve RSPPRC. The results of the computational experiments show that the robustness of the solution can be greatly improved with a moderate penalty in the optimal value.

To adopt the robust approach, one should determine the uncertainty sets, as well as the parameters to control the robustness of the solution. The travel time deviation d and maximum demand o can be estimated from historical data and/or a business contract, among other sources. It is somewhat unclear how to decide upon the parameters Γ (or Λ). Nevertheless, some guidelines can be given based on our computational experience:

- Use a higher value of Γ (or Λ) if each route is long in terms of the number of the customers to be visited. If a route has a long travel distance but only a small number of the customers to visit, the route may be protected with a small value of Γ (or Λ) as Γ (or Λ) restricts the number of delayed segments (or unexpected demands) in the route.
- The value of Γ (or Λ) can be adjusted based on the actual operation results. If the solution of routes from the current Γ (or Λ) yields many tight visits in the actual operation, it is natural to increase the value. If the current solution wastes too much time (or capacity), one may consider the possibility of reducing the value of Γ (or Λ).

Since the robust feasible routes are generated for each vehicle independently, it is even possible to apply different uncertainty sets to different vehicle types. For example, let us assume that there are two types of vehicles, one with a large capacity and the other with a small capacity, that is the heterogenous vehicles case. The large-capacity vehicle is clearly expected to visit many more customers than the small-capacity vehicle. We can assign, therefore, a higher Γ and/or Λ value to the former type of vehicle to protect the long (and important) route.

Unfortunately, the proposed algorithm cannot be directly applied to RVRPTW, where the customers have the earliest starting times for service as well as deadlines. In RVRPTW, the vehicle may be subject to a certain period of waiting at a customer's location since the service should be provided only after the earliest starting time. In this case, the delayed travel time prior to the waiting period can be buffered (absorbed) by the waiting time. Generalization of the RSPPRC to allow the waiting at the customer's location represents a valuable future research topic.

Acknowledgements—The authors would like to thank several anonymous referees for their numerous and very helpful comments and suggestions. This research is supported by Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education, Science and Technology (2009-0088465).

References

Augerat P, Belenguer J, Benavent E, Corberan A, Naddef D and Rinaldi G (1995). *Computational results with a branch and cut code for the capacitated vehicle routing problem*. Technical Report RR949-M, Artemis-Imag, Grenoble, France.

Barnhart C, Johnson EL, Nemhauser GL, Savelsbergh MWP and Vance PH (1998). Branch-and-price: Column generation for solving huge integer programs. *Operations Research* **46**: 316–329.

Barnhart C, Hane CA and Vance PH (2000). Using branch-and-price-and-cut to solve origin-destination integer multicommodity flow problems. *Operations Research* **48**: 318–326.

Berger J and Barkaoui M (2003). A new hybrid genetic algorithm for the capacitated vehicle routing problem. *Journal of the Operational Research Society* **54**: 1254–1262.

Bertsimas D and Sim M (2003). Robust discrete optimization and network flows. *Mathematical Programming* **98**: 49–71.

Bertsimas D and Sim M (2004). The price of robustness. *Operations Research* **52**: 35–53.

Bertsimas D and Simchi-Levi D (1996). A new generation of vehicle routing research: Robust algorithms, addressing uncertainty. *Operations Research* **42**: 286–304.

Bertsimas D, Pachamanova D and Sim M (2004). Robust linear optimization under general norms. *Operations Research Letters* **32**: 510–516.

Campbell AM and Thomas BW (2008). Probabilistic traveling salesman problem with deadlines. *Transportation Science* **42**: 1–21.

Campbell AM and Thomas BW (2009). Runtime reduction techniques for the probabilistic traveling salesman problem with deadlines. *Computers & Operations Research* **36**: 1231–1248.

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

Chang TS, Wan YW and Ooi WT (2009). A stochastic dynamic traveling salesman problem with hard time windows. *European Journal of Operational Research* **198**: 748–759.

Cheung RK and Hang DD (2003). A time-window sliding procedure for driver-task assignment with random service times. *IIE Transactions* **35**: 433–444.

Christiansen C and Lysgaard J (2007). A branch-and-price algorithm for the capacitated vehicle routing problem with stochastic demands. *Operations Research Letters* **35**: 773–781.

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

Desrochers M. (1986). *La fabrication d'horaires de travail pour les conducteurs d'autobus par une methode de generation de colonnes*. PhD Thesis, Centre de recherche sur les Transports, Publication #470, Universite de Montreal, Canada.

Dethloff J (2002). Relation between vehicle routing problems: An insertion heuristic for the vehicle routing problem with simultaneous delivery and pick-up applied to the vehicle routing problem with backhauls. *Journal of the Operational Research Society* **53**: 115–118.

Dorigo M, Maniezzo V and Colorni A (1996). Ant system: Optimization by a colony of cooperating agents. *IEEE Transactions on Systems, Man and Cybernetics* **26**: 29–41.

Fisher ML, Jornsten KO and Madsen OBG (1997). Vehicle routing with time windows: Two optimization algorithms. *Operations Research* **45**: 488–492.

Fukasawa R, Longo H, Lysgaard J, De Aragao MP, Reis M, Uchoa E and Werneck RF (2006). Robust branch-and-cut-and-price for the capacitated vehicle routing problem. *Mathematical Programming* **106**: 491–511.

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

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

Gendreau M, Laporte G and Seguin R (1996a). Stochastic vehicle routing. *European Journal of Operational Research* **88**: 3–12.

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

Hvattum LM, Lokketangen A and Laporte G (2006). Solving a dynamic and stochastic vehicle routing problem with a sample scenario hedging heuristic. *Transportation Science* **40**: 421–438.

Ioachim I, Gelinas S, Soumis F and Desrosiers J (1998). A dynamic programming algorithm for the shortest path problem with time windows and linear node costs. *Networks* **31**: 193–204, Wiley Online Library).

Irnich S and Desaulniers G (2004). *Shortest Path Problems with Resource Constraints*. Springer: New York.

Jula H, Dessouky M and Ioannou PA (2006). Truck route planning in nonstationary stochastic networks with time windows at customer locations. *IEEE Transactions on Intelligent Transportation Systems* **7**: 51–62.

Kohl N and Madsen OBG (1997). An optimization algorithm for the vehicle routing problem with time windows based on Lagrangian relaxation. *Operations Research* **45**: 395–406.

Kolen AWJ, Kan AHGR and Trienekens HWJM (1987). Vehicle routing with time windows. *Operations Research* **35**: 266–273.

Lin S and Kernighan B (1973). An effective heuristic algorithm for the traveling-salesman problem. *Operations Research* **21**: 498–516.

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

Russell RA and Urban TL (2008). Vehicle routing with soft time windows and Erlang travel times. *Journal of the Operational Research Society* **59**: 1220–1228.

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

Sungur I, Ordonez F and Dessouky M (2008). A robust optimization approach for the capacitated vehicle routing problem with demand uncertainty. *IIE Transactions* **40**: 509–523.

Tan KC, Cheong CY and Goh CK (2007). Solving multiobjective vehicle routing problem with stochastic demand via evolutionary computation. *European Journal of Operational Research* **177**: 813–839.

Toth P and Vigo D (2002). The vehicle routing problem. *SIAM* **9**: 5–10.

Yu B, Yang Z and Xie J (2010). A parallel improved ant colony optimization for multi-depot vehicle routing problem. *Journal of the Operational Research Society* **62**: 183–188.

Received March 2011;
accepted September 2011 after one revision