Vehicle Routing with Space- and Time-Correlated Stochastic Travel Times: Evaluating the Objective Function

Zhaoxia Guo[1], Stein W. Wallace[*2], and Michal Kaut[3]

[1]Business School, Sichuan University, Chengdu 610065, China, zx.guo@alumni.polyu.edu.hk
[2]Business School, Sichuan University, Chengdu 610065, China, and NHH Norwegian School of Economics, Bergen, Norway, stein.wallace@nhh.no
[3]SINTEF Technology and Society, Trondheim, Norway, michal.kaut@sintef.no

May 9, 2019

We study how to model and handle correlated travel times in two-stage stochastic vehicle routing problems. We allow these travel times to be correlated in time and space, that is, the travel time on one link in one period can be correlated to travel times on the same link in the next and previous periods, as well as travel times on neighboring links (links sharing a node) in both the same and the following periods. Hence, we are handling a very high-dimensional dependent random vector. We shall discuss how such vehicle routing problems should be modeled in time and space, how the random vector can be represented, and how scenarios (discretizations) can meaningfully be generated to be used in a stochastic program. We assume that the stochastic vehicle routing problem is being solved by a search heuristic, and focus on the objective function evaluation for any given solution. Numerical procedures are given and tested. As an example, our largest case has 142 nodes, 418 road links and 60 time periods, leading to 25,080 dependent random variables. To achieve an objective-function evaluation stability of 1%, we need only fifteen scenarios for problem instances with 64 customer nodes and 9 vehicles.

Keywords: stochastic vehicle routing; correlated travel times; correlated random variables; scenario generation; objective function evaluation

*Corresponding author

1. Introduction

Vehicle routing, in all its variants, is one of the most studied problems in logistics (Fisher and Jaikumar 1981, Laporte 2009, Potvin 2009, Pillac et al. 2013, Campbell and Wilson 2014, Lin et al. 2014, Prins et al. 2014, Cattaruzza et al. 2016). But for historical as well as numerical reasons, the vast majority of papers are on deterministic problems. Stochastic versions have started to occur, see for example the reviews Pillac et al. (2013), Gendreau et al. (2016), Oyola et al. (2018), Ritzinger et al. (2016). The most studied stochastic phenomenon is demand (Juan et al. 2013, Zhu et al. 2014, Goodson 2015), followed by travel time (speed) (Laporte et al. 1992, Tas et al. 2013, Han et al. 2014), service time (Lei et al. 2012, Errico et al. 2016) and finally, random occurrence of customers (Bent and Van Hentenryck 2004).

Most vehicle routing problems (VRPs) are solved using search heuristics (Potvin 2009, Vidal et al. 2013, Martinelli and Contardo 2015, Wang et al. 2015), and a natural part of doing so is to evaluate the objective function for a given solution. If that was easy, most existing heuristics for vehicle routing could fairly easily be adopted to the case of correlated stochastic travel times (or speeds). After all, most search heuristics have two parts, the search part and the objective function evaluation part. The focus of this paper is only on the evaluation part. In order to demonstrate our approach, we shall need both a test case within the family of VRPs and a search heuristic, but neither of these represent contributions.

In the following we shall talk about speeds rather than travel times. The two are of course equivalent, but we do this for two reasons. The first is that it seems more natural to talk about speed changing from one period to the next when in fact the trip is not completed within a given period. The second is that for scenario generation, it is best to work on bounded random variables, and travel times are unbounded if speeds are allowed to be zero (which makes sense over short time periods).

Our contributions are closely related to the many challenges set out in Gendreau et al. (2016) on how to represent a high-dimensional dependent random vector of speeds, how to generate scenarios for the underlying stochastic program, and how to perform function evaluations in the case where speeds are stochastic and correlated in time and space. By "correlated in time", we mean that the speed on a link in one period is correlated with the speed on the same link in nearby time periods. By "correlated in space", we mean that speeds on close by links are correlated, so that if there is a traffic jam on one link, most likely—but not for sure—there is a traffic jam also on neighboring links. This must be distinguished from VRP literature (stochastic as well as deterministic) covering "time-dependent travel times" (or speeds) but where the time dependence only means that speeds (or expected speeds in a few cases) are different in different time periods (Tas et al. 2014, Soysal et al. 2015, Nejad et al. 2016). The time-dependent VRPs in the literature have been investigated both on the map (Ehmke et al. 2016, Huang et al. 2017, Ehmke et al. 2018) and on the network of customer nodes (Ichoua et al. 2003, Tas et al. 2014, Cimen and Soysal 2017). Interested readers can refer to the review paper by Ticha et al. (2018) for more information. There will be an implicit time correlation in the speed scenarios in such a context (Fu and Rilett 1998, Ehmke et al. 2018). The correlation will take its sign from the direction in which the mean speed moves. Similarly, Huang et al. (2017) consider implicitly spatial and temporal correlations by classifying the links into three categories (i.e., expressways, arterials, and residential roads) and assuming the travel speeds of links in each category are time-dependent. However, correlations in real-world speeds represent more than that. They must represent such as the fact that free flow of traffic usually lasts for a while, and slow traffic takes time to disappear. Therefore the research cited above cannot cope with the challenges proposed by Gendreau et al. (2016). Our approach covers correlations whatever their cause.

Few previous VRP studies have considered space- and time-correlated travel times although real-world speeds are correlated in both time and space. Lecluyse et al. (2013) consider this type of correlated travel times in VRPs by using the concept of congestion circles to represent traffic congestion in road networks. Following the same representation, Schilde et al. (2014) further consider similar spatio-temporal correlations between travel speeds on different streets in a dynamic dial-a-ride problem. However, it is hard to determine the congestion circles accurately over time.

Hence, our starting point is very general. We are concerned with the evaluation of any solution (set of vehicle routes) to any VRP, by either declaring it to be infeasible or calculating the objective function value, when speeds are stochastic and correlated explicitly in time and space. For details, see towards the end of Section 2. The major questions will be how we can represent the stochasticity, including how scenarios can be generated (as discrete random variables will be needed), and how many scenarios are needed. The number of random variables will easily be in the tens of thousands, and care has to be taken in order to handle such big dimensions.

In other words, the setting is one where we have a stochastic VRP and a heuristic that can be separated into a solution-generating and a solution-evaluating part. Our contribution is then to perform the latter in a way that works for a large collection of different VRPs where there is an underlying map. We do this by combining an existing general scenario generation method (which fits the structure of VRPs well) with a stability test for evaluating objective functions. For VRPs with time- and space-correlated stochastic speeds, scenario generation is a major undertaking, which creates discrete random variables from the background data (or from some modeling activity). In fact, it is most likely the main bottleneck for solving such problems.

The paper is organized as follows: In Section 2 we discuss the modeling of VRPs with correlated speeds. Section 3 presents our overall approach. Section 4 is dedicated to scenario generation. Stability tests and an investigation into how many scenarios are needed can be found in Section 5. The test case is outlined in Section 6, while numerical results can be found in Section 7. We sum up in Section 8.

2. Modeling the Routes

The VRP is defined on a road network, represented by a directed graph $\mathcal{G} = (\mathcal{V}, \mathcal{L})$, where $\mathcal{V} = \{0, 1, \ldots, V\}$ is a set of vertices (nodes) and $\mathcal{L} = \{1, \ldots, L\}$ is a set of directed (road) links connecting the nodes in $\mathcal{V}$. Node 0 in $\mathcal{V}$ represents the depot at which are based K vehicles with capacity Q. Let $\mathcal{N} = \{1, \ldots, N\}$ ($\mathcal{N} \subset \mathcal{V}$) denote the set of customers, with specified demands, that need to be served. The VRP is the problem of finding the optimal set of vehicle routes starting and ending at the depot, such that each customer in $\mathcal{N}$ is served exactly once and the total demand of a route does not exceed Q, while optimizing a specified objective.

We shall split the planning horizon into P periods of equal length. In deterministic models with $P = 1$, we can simply compute the shortest travel times between the customer pairs and then proceed using the (much smaller) network of customers, instead of the road network $\mathcal{G}$. But if travel times (speeds) are random and depend on time, this is not quite as easy when $P > 1$. Huang et al. (2017) discuss this issue in great detail. They demonstrate that when speeds are time dependent, path selection between a pair of customers will depend on the time of day. They call this *path flexibility*. If speeds are also stochastic, the path will depend on both time of day and realizations of the random speeds.

Let us add a few comments about dependence as that is the focus of this paper. First, note that even if all road link speeds were independent, the pairwise customer speeds would not, as long as some of the best paths between customers share a link. Hence, assuming independence for customer-to-customer speeds would imply that no pairs ever shared a road link on the map, which is impossible to achieve for larger cases. So if we were to operate on the network of customer nodes, we would have to use stochastically dependent speeds. However, this dependence would be hard to describe as it would depend on path selection. And this relationship would be even more complicated if the road link travel speeds were already dependent. We are not investigating this potential approach for handling dependencies in this paper.

To be able to represent path flexibility as outlined by Huang et al. (2017), while also handling path dependencies as just outlined, we find it necessary to operate on the map and not on the network of customer nodes. Therefore, a scenario contains the speed on each road link in each period, so it is of dimension $L \times P$.

Note that there are papers that use the assumption of independent customer-pair travel times (Tas et al. 2013, Feng et al. 2017). This can of course be seen as an approximation of real-world travel times. But overlooking dependence, caused by link speeds being themselves dependent and paths sharing links, will result in too low variance on route travel times, as disregarded correlations will mostly be positive. Models using independent customer-pair speeds will therefore underestimate travel time variation.

For route selection within the VRP, as described in Section 3, our approach will allow for any of the following principles.

- The route, given from the solution-generating part of the algorithm, is a complete sequence of nodes (or links) from $\mathcal{V}$, not just a sequence of customer nodes. In the case that the route is a sequence of nodes, it can be allowed to look for quicker detours (rather than the direct link).
- A route is a sequence of customer nodes, and the path between two consecutive customer nodes is determined by the path with the shortest *expected* travel time, given the point in time we leave the first node in the pair. An interpretation is

hence that when the vehicle is actually operating, it will need expected speeds (given the time of day), but not actual speeds. This is partly in line with the concept of path flexibility as outlined earlier (Huang et al. 2017), in which the path between two consecutive customer nodes is selected from a set of candidate paths consisting of distance-minimizing paths and time-minimizing paths for a given departure time and average travel times in each time period.

- A route is a sequence of customer nodes, and the path between two consecutive customer nodes is determined by the path with the shortest travel time, given the speeds at the point in time when we leave the first node in the pair. In our view this is a very reasonable choice if real-time data for speed is available when a vehicle is actually operating.

- Technically, we can also handle the case when paths are picked based on the full set of speeds in the scenario. But this makes little sense to us as it implies using principally unavailable information about the future.

It is worth noting that many companies use available apps or other software to determine paths when actually driving. But beware that this is very difficult (if not impossible) to include in a model for a VRP, where finding routes is the issue. So even though these services may indeed lead to very good paths on a route, they are not easy to use to find the routes (i.e., which customers to assign to which vehicle, and in which order they should be visited). Using present speeds to determine paths approximates this approach.

We are operating in a two-stage setting. So our approach does not support a true multi-stage setting where customers assigned to a given vehicle are determined on-route (which makes sense for pickups and some deliveries) or where the customers are determined beforehand, but the sequence is not. The approach can be used to approximate the latter case, but no true dynamic optimality will be found.

Given a complete route of nodes (or links) in the graph, we need to calculate the travel time (and certain costs) for a given vehicle and a given scenario. Our approach can handle the following alternative situations, depending on what is the setup in the VRP.

- No time windows.

- If there is an earliest arrival time at a customer, we can handle a requirement to wait (hard constraint) and / or pay a penalty for being early (soft constraint).

- If there is a latest arrival time at a customer, we can handle a penalty for being late (but still serve the customer, a soft constraint), simply skipping the customer and pay a penalty (another type of soft constraint), or declare the route to be infeasible (hard constraint).

In the test case in Section 6, we assume that a route is a sequence of customer nodes, and that path selection is based on present speeds when a vehicle leaves a customer. In all cases, we shall be careful to handle the stochastically dependent speeds consistently when evaluating the objective function for a given route on a given scenario, as that is the core of this paper. For example, even if the path between a pair of customers is determined based on present speeds, the actual travel time will be valued based on the realizations in the scenarios.

3. Evaluating the objective function

As mentioned in Section 1, search heuristics are usually used to find good solutions to VRPs. These heuristics start with one or more initial candidate solutions and then try to improve the candidate solutions iteratively via local moves. During iterations, the performance of each candidate solution to the stochastic VRP is evaluated by the function evaluation method outlined in this section.

Let us sum up the parts that are needed to solve stochastic VRPs:

(1) a search heuristic,

(2) an objective function evaluation approach, and

(3) scenario set consisting of $|S|$ scenarios. Each scenario $s \in S$ has one speed for each link in each period, so a complete scenario represents one possible realization of speeds across time and space (e.g., all travel speeds on all road links in all periods).

We take the search heuristic from the literature. Assume we have a feasible solution to the VRP (a set of routes) using K vehicles. Each route is a sequence of customer nodes plus a starting time from the depot. The search heuristic operates just as in the deterministic case, as it moves from one solution to the next. Given a solution, the road network graph $\mathcal{G}$, and the scenarios, we now need to compute the expected objective value corresponding to the solution. Take a VRP of minimizing the total cost f as an example. Whenever a solution needs to be valued, the objective function evaluation approach proceeds as follows:

```
For each scenario s ∈ S do
   For each route r in the solution do
      Set t(s,r) equal to time leaving depot
      Set f(s,r) to zero
      For each pair (i,j) of customers (including depot) do (* in sequence on route *)
         Take speeds from scenario s at time t(s,r)
         Find the best path from customer i to customer j using these speeds, based
            on the objective to be optimized
         Update t(s,r) based on speeds in scenario s, handling time and possible
            waits at the customer to know when leaving customer j
         Update f(s,r) to reflect the costs incurred from customer i to customer j
      End
   End
End
```

We now have travel times and costs on each scenario for each route, and can calculate values of travel times and costs as required by the VRP being analyzed.

We see that both the heuristics for finding routes and the best path calculations can be done with existing deterministic methodology. Thus, what is missing are the scenarios. We shall take a scenario generation method from the literature that takes marginal speed distributions, a limited set of correlations and the requested number of scenarios as input and produces scenarios as output. This procedure is purely technical, and will be presented in Section 4. To see if it actually delivers in our context, we check how many scenarios are needed to create results with acceptable accuracy and numerical feasibility by a stability test. Also this test follows known procedures from the literature, and will be presented in Section 5. So the contribution of this paper is to show how all these pieces can be put together in the right way so as to produce a numerically feasible approach for the space- and time-correlated stochastic VRP.

If two consecutive customers i and j on vehicle k's route are not directly connected by a link in $\mathcal{G}$, we find, using Dijkstra's algorithm, or one of its variants, the best path (e.g., with the shortest travel time) between customers i and j using speeds on the links in scenario s for the period in which the vehicle leaves node i. The logic is that when the vehicle leaves node i, in reality, only speeds for that period are known, and they will be used to decide on where to travel. Hence, this path might not be best ex-post, i.e., after reaching node j, if that happens in a later period. Of course, other ways to find the best path between customers i and j can be used, without any changes in the function evaluation approach. We refer to Casey et al. (2014) for a more detailed overview over time-dependent shortest path algorithms.

With the setup above we start at the depot. We use present speeds (or possibly expected speeds as discussed in Section 2) in the Dijkstra algorithm to find the best path to the first customer. The setup then follows the method of Ichoua et al. (2003); we follow a link either to the next node (as we are still in the same period) or we follow the link until the period ends, using the speed of that period, and then continue on the same link with the speed of the following period. When we arrive at the customer, we check potential time windows, and add possible earliness or tardiness penalty costs, if any, wait or declare an infeasibility, depending on the VRP at hand. Then we move to the next customer using exactly the same setup, until we are back to the depot. We then register total travel time plus potential penalties along the route.

Our approach, as outlined above, can handle many variants:

- A route is a sequence of nodes from the map rather than sequence of customers. In that case the best path calculations can be skipped.
- The best path calculation can be skipped if two customer are neighbors on the map in case one wants to model that drivers never make detours for direct links.
- The best paths are based on average speeds rather than present speeds to represent that real-time speeds will not be available during operations.
- Hard and soft time windows.

Numerical tests on variants of the VRP can be found at the end of Section 7.1 and in the Appendix. This will show that the approach is fairly flexible, and that the core question is how to find the right set of scenarios.

4. Multi-Dimensional Distributions and Scenario Generation

Stochastic programs need discrete random variables. As pointed out earlier, some papers build on the assumption that customer-to-customer speeds are independent, despite the fact that this cannot be the case in real-world road networks with sufficient traffic flow. Apart from this, we see that Tas et al. (2013, 2014), Feng et al. (2017) use distribution functions (so not scenarios) that can be added along paths, and these approaches are distinctly using stochastic independence. In Huang et al. (2017) and Han et al. (2014), the scenarios of travel speeds are assumed given, whereas the scenarios are sampled from a truncated normal distribution in Lee et al. (2012). In Cimen and Soysal (2017), we find assumptions of independent normal distributions that are subjected to a discretization scheme to fit into an approximate dynamic programming framework.

Most of the general literature on scenario generation is about how to generate scenarios from a distribution of some sort, see for example Dupačová et al. (2000), Lee et al. (2012) and Chapter 4 of King and Wallace (2012) for reviews. But most of these approaches will not work numerically in the kind of dimensions we face in many real-world VRPs. For example, our largest case with 418 road links and 60 time periods results in 25,080 random variables and more than 310 million distinct correlations (more details in Section 6), and this is not even a very large setting from an applied perspective. So even if we had data, and could calculate marginal distributions and a correlation matrix (in their own rights not very difficult), it would be quite a challenge to use them for finding scenarios using any of the existing methods. It seems to us that the only way would be to sample from the empirical distribution if we had one. If so, the issue would be how many scenarios would be needed in order to achieve a good set of scenarios representing the distribution well in these high dimensions. We expect the number to be very large, in line with the expectations expressed in Gendreau et al. (2016).

It is common in the optimization literature, when the issue is testing of algorithms and we do not have data, to simply invent a reasonable data set. In our case, "reasonable" mostly refers to the resulting correlations that, even if not real, should make sense – they should describe a potentially real situation. Either one sets up scenarios directly, or one guesses on a distribution and then uses that distribution to find scenarios, be that by sampling or some other methodology, for example using one of the many methods outlined in Chapter 4 of King and Wallace (2012). However, notice that there are some major issues here. We believe that simply guessing a limited set of scenarios in these dimensions makes little sense. The distribution we would then, in fact, be using, would easily be pure noise. So the alternative would be to guess a distribution. We would then suffer from the same dimensionality problem as we just mentioned for the case when we had data. For our largest case we would need more than 310 million correlations. But the fact is, guessing on a 25,080 by 25,080 matrix and ending up with a positive semi-definite matrix (so that it would potentially be a correlation matrix) is close to impossible, except for the uncorrelated case, which is not what we discuss in this paper.

4.1. Updating a Guess of a Correlation Matrix

If we have a matrix that we "like", i.e., a matrix that to the best of our understanding shows how the dependencies are, but which does not lead to positive semi-definiteness, we can, in principle, proceed by finding some matrix that is close to the one we have, but which is positive semi-definite, and hence could be used as a correlation matrix. This approach is outlined in Lurie and Goldberg (1998), where they minimize the root-mean-square error of the positive semi-definite matrix relative to the starting matrix. Be aware, though, that non-zero correlations may show up at the most peculiar places representing strange dependencies. These strange correlations may affect the optimal routes directly, for example by utilizing a fake negative correlation to hedge against variation in route travel times. It is normally possible, ex post, to check if strange correlations ended up effecting the optimal solution. We shall discuss this in more detail in Section 5 where we discuss stability. Updating a guessed matrix in the dimensions we are facing will in any case be quite a numerical challenge, as a Cholesky decomposition and a Gauss-Newton method are involved, so also this argues against trying to guess a correlation matrix. The updated matrix will normally be singular, and that is also a challenge for some scenario generation methods[1].

Let us point out that it is up to the user to accept if a positive semi-definite matrix is reasonable or not as a correlation matrix. This is not a mathematical question, but a question of modeling or problem understanding. The matrix will most certainly have unexpected non-zero elements, representing unexpected correlations (even negative ones), but may still be acceptable for our problem.

4.2. Practical Consideration

Let us see where we stand. We could guess a set of scenarios and use them. Most likely that would result in a distribution with totally arbitrary properties. We could estimate (if we had data) or guess a correlation matrix, but as we have pointed out, guessing is very difficult – close to impossible. In any case, we would be facing serious numerical challenges due to the size of the matrix. So, in our view, we need an alternative approach that is numerically feasible, and would work whether or not we had data. It will amount to a heuristic, but we shall see that we can control the quality of the approach.

We shall use the scenario-generation method from Kaut (2014) that allows for speci-

[1] Lurie and Goldberg (1998) write "We applied the procedure to replace this indefinite matrix with a "nearby" positive semi-definite matrix. All of the eigenvalues changed as a result of this procedure. In particular, the single negative eigenvalue was replaced with the value of zero, resulting in an adjusted correlation matrix that was positive semi-definite (but not positive-definite). This result occurred because the procedure attempts to minimize the distance between the adjusted and unadjusted correlation matrices. Any larger adjustment, yielding a strictly positive eigenvalue, would have increased the value of the distance measure."

fying marginal distributions plus a subset of correlations[2]. This method fixes values for all variables upfront and assigns those values to scenarios, trying to match the specified "correlations". Randomness appears only as a tie-breaker in the heuristic used for the assignment. As a result, the method has a high probability of producing exactly the same scenarios on two consecutive runs. The method allows any subset of correlations, and we shall make a specific choice, consistent with the structure of solutions to VRPs — routes. Numerical testing will show that it is a good choice. Moreover, the scenario generation method has the property that the means in the scenario set are always equal to the means in the underlying distribution. So even though other aspects of the distribution might be off (it is an approximation, after all), the mean of travel times on each link and hence the mean on each route are correct.

Let Y denote the number of neighboring link pairs (pairs of links meeting in a node) in a road network. We shall specify the correlations for each such link pair in each period. This leads to $Y \times P$ correlations. Furthermore, in period p $(1 \leq p < P)$, the travel speed on link l is correlated with the speeds on the same link and its neighboring links in the next period, which leads to $(Y + L) \times (P - 1)$ correlations within P periods. Other correlations are not specified as they are not used in the scenario generation method. That is, given P and L, the number of correlations among links in the network, used in our scenario generation method, is $Y \times P + (Y + L) \times (P - 1)$, which is much less than the total number of correlations. Since Y is linear in L, the total number of correlations is linear in the number of random variables, $L \times P$. In our largest case with $Y = 2015$, L=418 and P =60, we need to define $2015 \times 60 + (2015 + 418) \times 59 = 264,447$ correlations, much less than the total number of over 310 million.

The question is then whether or not using such a limited set of correlations causes major problems in the overall setup of Section 3. If we have data, we would calculate these correlations from the data set, then knowing that there really is at least one true correlation matrix containing the correlations we used. If we do not have data, we would use problem understanding or intelligent guesses to find them. This could lead to a set of "correlations" that in fact are not part of any actual correlation matrix, and we would not know if this had happened. For this reason, any scenario generation method that needs consistent data could be in trouble, since the scenario generation could be based on a non-existing distribution. Our setup does not suffer from this problem directly, as the scenario generation method we use looks for scenarios that in some sense are close to what was specified. But that means that we are facing two sources of noise if we have guessed on the data; The fact that only a limited set of correlations was used, and that possibly, the scenarios were generated based on inconsistent specifications. With real data, only the first source of noise is present. The stability tests in the next section will help us see whether or not this noise caused serious errors. We shall see that despite these potential problems, the approach does indeed produce high quality results.

[2]Actually, the method works by matching bivariate copulas for all specified variable pairs, but since we use normal copulas that have correlations as their only parameter, we will use the term 'correlations' in the rest of the paper.

5. Stability

Gendreau et al. (2016) point out that "*the number of scenarios required to realistically describe all of the possible realizations of the parameters will likely be quite large, thus creating tractability issues*". But how large is that? Unless the scenarios put into the stochastic program *is* the full description – and that is hardly the case in vehicle routing – there is a need to be assured that the scenario set represents the underlying data or distributions well. The classical problem here is a trade-off between a solvable problem but where the results are just noise due to a bad representation, and a numerically unsolvable problem (due to size) but where the representation is good. This is a central question for any attempt to solve stochastic VRPs, and the answer will depend on what method is used to generate the scenario set.

Since we assume that the VRP is solved with a search heuristic, evaluating the objective function for a given feasible solution will be linear in the number of scenarios. So with $|S|$ scenarios, the objective function evaluation will take $|S|$ times longer for the stochastic case than for the deterministic case (which corresponds to one scenario). It is therefore crucial to keep the number of scenarios low, while at the same time knowing the quality of the evaluation for the given scenario set. Our point in this paper is to show that with a careful choice of scenario generation method, $|S|$ can be kept rather low.

There are several ways to assess the quality of a set of scenarios, either in terms of the set itself or the solution it provides. An overview can be found in Chapter 4 of King and Wallace (2012). What we shall use is a heuristic test, inspired by Kaut and Wallace (2007). The goal is to provide confidence that we have enough scenarios to obtain quality solution evaluations.

In order to determine the necessary number of scenarios, we proceed as follows. We take a number of instances of our problem. The instances will vary in terms of the number of customers and vehicles, but will be for a given graph $\mathcal{G}$. We then ask, given our scenario generation method and our test case (details in Section 6), how many scenarios do we need so that we know that the objective function evaluations are good enough if that scenario generation method is used?

Many standard scenario-generation methods, such as sampling, will produce quite different results if they are rerun with the same input data. We can therefore generate a number of scenario sets of the same size and measure stability on them. This research uses the scenario-generation method from Kaut (2014). As mentioned in Section 4.2, this method works differently and has a high probability of producing exactly the same scenario set on two consecutive runs, so the 'standard' stability tests would overestimate its quality. Instead, we use a workaround suggested in Kaut and Wallace (2007) and replace N scenario sets of size $|S|$ with $2m+1$ sets of sizes $|S|-m$, $|S|-m+1, \dots, |S|, \dots, |S|+m$, for some chosen m.

The stability test is then as follows: for each instance, generate a set of feasible solutions; the first is the optimal (or a good) solution to the deterministic version of the problem (as that is available by assumption), the others some permutations of this solution (more details in Section 7.1). For each of these solutions, calculate the objective function value according to the method described in Section 3 using each of the $2m+1$ scenario sets. Then find the largest (F^+) and smallest (F^-) objective function values and calculate the relative difference $(F^+ - F^-)/F^+$. We define the *stability level* of $|S|$ for that instance and that value of m to be the largest such difference over all the tested solutions. Then pick the smallest $|S|$ having the required stability level, say 1% or 2%. We study how the stability varies across instances to come to overall views on the choice of $|S|$. We make sure that we only apply this test when F^+ and F^- have the same sign. In our cases, they are always positive. We also test the dependence on m.

Doing stability testing this way has an added positive feature relative to the more common in- and out-of-sample stability used in Kaut and Wallace (2007). When a stochastic VRP is solved with a heuristic, there will be two sources of noise: the search heuristic and the scenario generation method. When fixing a set of solutions, as we do, all the noise in the measures come from the scenarios, helping in deciding the necessary number of scenarios. After all, the choice of search heuristic is not the point of this paper; the point is to evaluate feasible solutions. This problem of scenario generation and search heuristic is also part of the setup in Hui et al. (2014).

As already outlined, we are using a scenario generation method that is in itself a heuristic, sometimes not even based on a genuine multi-dimensional random vector. In particular, the method will most likely lead to some spurious correlations in the scenario set. In Section 4.1, we discussed how a matrix that had been guessed, based on our best understanding of a problem, could be updated to become a proper correlation matrix, if the guess did not lead to a positive semi-definite matrix. We pointed out that, apart from major difficulties of dimensions, this is in own right straightforward. But we warned against the method as it would result in spurious correlations, also negative – here and there. But our scenario generation method above also results in spurious correlations. So is there any difference? The answer is yes; we make sure that the most important correlations – those along routes – are correct, while the spurious ones (though still potentially troublesome) are away from routes. When updating a matrix as in Section 4.1, the spurious correlations can (and will) show up anywhere, including along routes, and hence have more severe effects on function evaluations. But even so, how can we know that our approach did not lead to some real problems in terms of function evaluations, even though the spurious correlations are off the routes? Stability testing helps us here. If our approach to scenario generation led to strange correlations (somewhere), and this again affected the objective function values, we shall be unable to obtain stability because it is extremely unlikely that the spurious correlations would be the same in all the generated scenario sets, given how we created sets of different sizes. Hence, when we in this paper report stability – which we will – that does not only show that we have enough scenarios, but also that the scenarios (though they have some peculiar correlations here and there) did not disturb the objective function evaluation (too much).

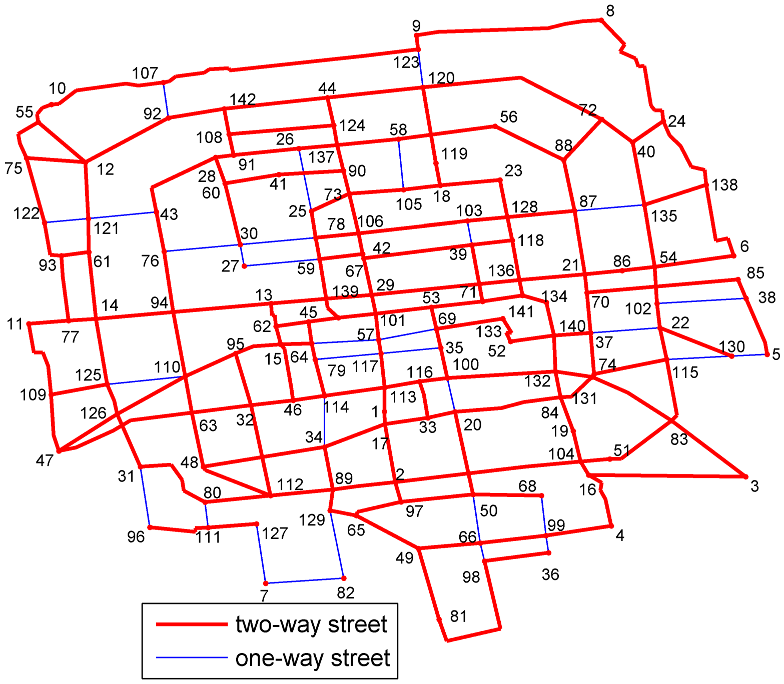

Figure 1: Simplified Beijing road network with 142 nodes and 418 road links.

6. Test Case

As a general observation, stochastic speeds are only of major importance when there is an asymmetry between the effects of being early and being late. In particular, just minimizing expected travel times with no special constraints such as time windows (hard or soft) will hardly make it worthwhile to study stochastics. Hence, time windows (where the effects of being early or late are very different) or penalties for particularly long working days, are cases where stochastics is likely to matter.

As an example, we consider a two-stage stochastic VRP where there are penalties for late arrivals of vehicles back to the depot, but no gains for early arrivals. An example would be over-time pay for drivers. The instances are defined on a simplified map of Beijing with $V = 141$ (142 nodes in total), $L = 418$ (road links) and $Y = 2015$ (neighboring-link pairs); see Figure 1. On this map, we define *instances* specified by triplets (N, K, P). As defined in Section 2, the N customer nodes are nodes 1, 2, ..., N in $\mathcal{V}$. For each N, the numbers for K, vehicle capacity Q and customer demands are based mainly on the values of corresponding instances in set A of NEO capacitated VRP instances[3], to ensure that they make sense. We shall not have time windows in the examples. Note that all the instances have $L \times P = 418 \times P$ random variables, which leads to $(L \times P - 1) \times (L \times P)/2$ correlations. For the largest case with $L = 418$ and $P = 60$, the corresponding number of correlations is 314,490,660. The objective is to minimize expected overtime pay.

[3]See http://neo.lcc.uma.es/vrp/vrp-instances/capacitated-vrp-instances/

The details of the relevant experimental data for these instances and other experiments in the following sections are available in the online supplement of this paper. In this paper, travel speeds are described by bounded beta distributions, denoted by (α_{lp}, β_{lp}, v_{lp}^{min}, v_{lp}^{max}). Here v_{lp}^{min} and v_{lp}^{max} denote the minimal and the maximal travel speeds on link l in period p. α_{lp} and β_{lp} are two parameters which control the shape of the distribution. Setting $\alpha_{lp} = \beta_{lp} > 1$ leads to a symmetric unimodal distribution. In period p, for a given edge l, we set: (1) $\alpha_{lp} = \beta_{lp} = \gamma_{lp}$, where γ_{lp} is a constant associating with link l and period p; (2) v_{lp}^{min} equal to the average speed in the current period minus 15; and (3) v_{lp}^{max} equal to the average speed in the current period plus 15. To ensure $\gamma_{lp} > 1$, we set $\gamma_{lp} = (\zeta^{max} \times v_{lp} \times 1.01)$ / $(\zeta_l \times v_{lp}^{min})$, where ζ^{max} denotes the maximal link length in the road network, ζ_l the length of link l, v_{lp} the average speed of link l in period p, and v_{lp}^{min} the minimal average speed of link l in all periods. These parameters can certainly be set at other reasonable values. The correlations we need are set as follows. The correlations for travel speeds on each neighboring link pair (pair of links meeting in a node) in the same period and in two neighboring periods are set to 0.7 and 0.42 respectively. The correlations for travel speeds on the same link in two neighboring periods are set to 0.6. Given the approach we use to create scenarios, no other correlations are needed.

To solve the stochastic VRP, the first stage (decided by the search heuristic) is to pick K routes, in terms of sequences of customer nodes and starting times from the depot, and the second stage decision (done by us) is, for each scenario, to travel these routes, picking paths between pairs of customer nodes to minimize expected costs. We mentioned in Section 3 that our approach could handle variants of the overall setup. In this test we shall use one of these. We assume that if two customers are neighbors on the map, i.e., there is a direct link between them, we shall follow this link, and not look for potential shorter detours. This is a modeling assumption on how drivers behave when visiting two customers that are directly connected by a road link. Looking for detours, the general setup, would of course also work.

6.1. Chosen Search Heuristic

In order to demonstrate our approach we need to apply a search heuristic to our problem. However, our approach does in no way depend on this choice, and we do not claim that this is the best approach for our test case; that is not the point of this paper.

For the stochastic VRP, the objective value of each candidate solution is evaluated by the method described in Section 3. We use the active-guided evolution strategy (AGES) proposed by Mester and Bräysy (2007) to find good solutions to our test cases. The AGES involves two phases. The first phase creates a starting solution by generating a set of solutions first with a hybrid cheapest insertion heuristic and then choosing the best solution found as the starting solution. The second phase is to improve the starting solution with a two-stage procedure by using the well-known guided local search metaheuristic (Voudouris and Tsang 1998) and the 1+1 evolution strategy metaheuristic (Beyer and Schwefel 2002). It has been demonstrated that the AGES had a better performance than several well-known benchmarking methods (Vidal et al. 2014).

Table 1: Out-of-sample performance comparison for SP and DP solutions for some triplets (N, K, P).

$(18,3,3)$			$(32,5,3)$			$(48,7,3)$			$(48,7,30)$			$(48,7,60)$		
DP	SP	diff.	DP	SP	diff.	DP	SP	diff.	DP	SP	diff.	DP	SP	diff.
0.87	0.85	-2.3%	2.26	1.98	-12.4%	6.07	5.46	-10.0%	8.48	7.66	-9.67%	8.51	7.91	-7.1%

Table 2: Number of correlations specified for generating scenarios for different P.

$P = 3$	$P = 5$	$P = 15$	$P = 30$	$P = 60$
10,911	19,807	64,287	131,007	264,447

6.2. Is it important to solve the stochastic VRP?

We start by testing whether uncertainty matters for this case – otherwise, there would be no point in testing the stochastic VRP approach. To compare the performance of the stochastic VRP (solved by the heuristic in Section 6.1) with its deterministic counterpart, we use some simulated instances with customers randomly placed on the map. We compare the out-of-sample performance of the solutions to the stochastic problem (SP) with the solutions to the corresponding deterministic problem (DP).

Table 1 compares results for five such problem instances. For each instance, we show the mean of nine out-of-sample objective values (corresponding to $m = 4$ in Section 5) of the best solution to the DP and the SP as well as the percentage differences between the averages. The out-of-sample test used is described as the "weaker out-of-sample test"[4] in Kaut and Wallace (2007, p. 262). For each instance, using $|S|$ from Table 4 – which shows the number of scenarios needed for a stability level of less than 2% – we have generated nine scenario sets of sizes $|S|$-4, $|S|$-3, ..., $|S|$, ..., $|S|$+4, solved the VRP problem on each of them and then evaluated the solutions on the remaining 8 scenario sets and computed the average. For the deterministic version, we have used all 9 scenario sets for the evaluation. As expected, the SP generates lower out-of-sample ("true") objective values than the DP does. Taking problem (48,7,3) as an example, the average objective value (5.46 hours) generated by the SP is around 10% less than the corresponding value (6.07 hours) generated by the DP. This shows that uncertainty has a reasonably large effect on the solutions in the test case. The number of specified correlations, as explained in Section 4.2, is given in Table 2. The numbers might seem high, but they are extremely low compared to the total number of correlations involved.

[4]We use the weaker out-of-sample test because a full-scale test requires either a full distribution (with an exact evaluation of the out-of-sample values as in Zhao and Wallace (2014)) or more often, a distribution to sample from in order to obtain an out-of-sample estimate. But since we do not have a full distribution, we cannot do any of these tests — sampling requires something to sample from.

7. Experimental Results

7.1. Stability Tests for Test Case

The next step is to find the minimal number of scenarios $|S|$ that can achieve stability for each of our test instances. For this, we follow the method described in Section 5, with $m = 4$. We first illustrate the approach on instance $(18, 3, 3)$ — that is, a case with 18 customers, three vehicles and three time periods — and then present results for all the tested instances.

For the test, we use 10 solutions, where the first one is the **best solution found** to the deterministic version of the problem instance, and the other nine are generated by randomly swapping two customer nodes or randomly putting a customer node from one route into another in that solution. For each solution, we test stability for several values of $|S|$. For each of them, we generate nine scenario sets sized from $|S| - 4$ to $|S| + 4$, evaluate the objective function on each of them and calculate the average objective value (AOV) and the relative differences (RD) between the largest and the smallest objective values. The results are presented in Table 3, where the last line is the stability level for a given $|S|$, defined in Section 5 as the maximum RD over all the tested solutions. We see that the scenario generation method gives a reasonable stability. When increasing the number of scenarios, the relative differences usually get smaller, and the AOVs usually get closer to the AOV found with $|S| = 1,000$. In this case, if the required stability level happens to be 2%, we can set $|S| = 100$ since it is the minimum $|S|$ that achieves this level.

Applying the same logic, we have determined the minimal $|S|$ to achieve stability levels under 1% for a number of instances. The results, including the actual stability level for each instance, are presented in Table 4. The main observation is that the number of scenarios needed is very low – **just 15** – for the larger instances, though somewhat larger for the smaller instances. Note that we do not present results with less than 15 scenarios, although several large problem instances need only 10 scenarios to achieve a stability level of 1%.

Note that so far we have tested stability levels for the evaluation of given feasible solutions, which is a weaker concept than in- and out-of-sample stability as discussed in Kaut and Wallace (2007), as that involves also the interaction with the optimization (in this case heuristic) procedure, as explained in Section 5. We do this as our goal is to show how objective functions can be evaluated without reference to the chosen heuristic. However, for completeness, we also tested relative differences for the best found solutions using the heuristic in Section 6.1. We did this for all instances using the scenario counts in Table 4. In all cases, stability was confirmed for the stochastic solution. The results are presented in Table 5. We see that the relative differences are within or very close to those presented in Table 4.

All tests have used nine scenario sets, corresponding to $m = 4$. To make sure $m = 4$ was not a bad choice, we investigated the effects of increasing m for each problem instance and the value of $|S|$ shown in Table 4 . The complete results are shown in Table 6. We note that increasing m from 4 to 6, 8 and 10 increases the relative differences on average

Table 3: Results of stability tests for test instance VRP (18,3,3), that is, an instance with 18 customers, 3 vehicles and 3 time periods.

Number of scenarios		10	30	50	100	500	1000
Solution 1	AOV	0.8640	0.8615	0.8619	0.8616	0.8648	0.8646
	RD	4.99%	2.85%	3.79%	1.65%	0.65%	0.33%
Solution 2	AOV	1.3138	1.3080	1.3080	1.3107	1.3106	1.3105
	RD	4.56%	1.92%	0.91%	0.87%	0.36%	0.14%
Solution 3	AOV	1.7906	1.7811	1.7882	1.7863	1.7864	1.7858
	RD	3.63%	1.33%	1.49%	0.53%	0.32%	0.19%
Solution 4	AOV	4.0130	4.0007	4.0023	4.0068	4.0078	4.0073
	RD	1.58%	1.17%	0.96%	0.57%	0.27%	0.23%
Solution 5	AOV	4.7591	4.7572	4.7522	4.7538	4.7552	4.7563
	RD	2.73%	2.59%	1.51%	0.73%	0.58%	0.30%
Solution 6	AOV	4.7591	4.7572	4.7520	4.7535	4.7548	4.7559
	RD	2.73%	2.59%	1.51%	0.73%	0.59%	0.30%
Solution 7	AOV	3.2539	3.2447	3.2362	3.2350	3.2375	3.2402
	RD	1.62%	1.79%	1.22%	1.19%	0.37%	0.33%
Solution 8	AOV	3.0840	3.0612	3.0672	3.0632	3.0625	3.0624
	RD	1.53%	0.87%	1.35%	1.17%	0.28%	0.27%
Solution 9	AOV	2.8224	2.7840	2.7793	2.7780	2.7845	2.7863
	RD	7.66%	1.85%	2.57%	1.32%	0.55%	0.44%
Solution 10	AOV	4.2755	4.2517	4.2485	4.2382	4.2469	4.2483
	RD	2.80%	1.85%	1.44%	1.04%	0.35%	0.33%
stability level		7.66%	2.85%	3.79%	1.65%	0.65%	0.44%

Table 4: Stability levels (SL) in the evaluations.

	$P=3$		$P=5$		$P=15$		$P=30$		$P=60$	
(N,K)	$\lvert S\rvert$	SL	$\lvert S\rvert$	SL	$\lvert S\rvert$	SL	$\lvert S\rvert$	SL	$\lvert S\rvert$	SL
$(18,3)$	100	1.7%	60	0.7%	65	1.0%	45	0.9%	35	1.0%
$(32,5)$	55	0.9%	20	0.9%	25	0.8%	22	0.9%	32	0.8%
$(48,7)$	25	0.8%	15	1.0%	15	0.6%	15	0.9%	15	0.9%
$(64,9)$	15	1.0%	15	0.6%	15	0.6%	15	0.8%	15	1.0%

Table 5: Stability of best solutions of cases from Table 4, in terms of relative differences (RD) of objective values.

	$P=3$		$P=5$		$P=15$		$P=30$		$P=60$	
(N,K)	$\lvert S\rvert$	RD	$\lvert S\rvert$	RD	$\lvert S\rvert$	RD	$\lvert S\rvert$	RD	$\lvert S\rvert$	RD
$(18,3)$	100	1.7%	60	0.9%	65	0.7%	45	1.1%	35	1.3%
$(32,5)$	55	0.9%	20	0.9%	25	0.8%	22	0.7%	32	0.5%
$(48,7)$	25	0.4%	15	1.1%	15	0.3%	15	0.7%	15	1.0%
$(64,9)$	15	0.4%	15	0.4%	15	0.3%	15	0.4%	15	0.3%

by 19%, 29%, and 37%, respectively. Given the definition of stability level, it is bound to increase when m increases. But we notice that even for m=10, the stability levels are between 0.6% and 2.0% for the 20 instances. This is especially surprising for cases with $|S| = 15$, since $m = 10$ means that we are comparing values from sets with $5, 6, \ldots, 25$ scenarios. To conclude, the table suggests that the numbers of scenarios indicated in Table 4 are safe to use, and that we have the necessary stability.

To further validate the results shown in Table 4, we took the solutions from the tests in Table 4 and performed the weaker out-of-sample test outlined in Kaut and Wallace (2007, p. 262). We cannot do a complete out-of-sample test as we have nothing to sample from to perform such a test. For each instance, we take each of the nine solutions (as we have nine different scenario sets and each scenario set corresponds to one solution) and evaluate them in the other eight scenario sets. Each solution then leads to a relative difference (over eight values), and we thus get nine relative differences, among which the largest is defined as the out-of-sample stability level. The results given in Table 7 further confirm the stabilities of the numbers of scenarios indicated in Table 4, and they are reliable to use.

How do we understand and interpret these results above?

1. A solution consists of routes. A route is actually a sequence of neighboring road links in time and space, and they have correlations we have controlled well in the scenario generation method. The calculation of travel time on a route and corresponding overtime pay is therefore rather good, particularly taking into account that the expected travel time is always correct (details in Section 4.2). Even if the scenario set contains spurious correlations, they do not affect the route travel time

Table 6: Stability levels of cases from Table 4, for varying m.

(N, K, P)	$\|S\|$	$m = 4$	$m = 6$	$m = 8$	$m = 10$
$(18, 3, 3)$	100	1.7%	1.8%	1.8%	2.0%
$(18, 3, 5)$	60	0.7%	0.8%	0.8%	0.8%
$(18, 3, 15)$	65	1.0%	1.8%	1.9%	1.9%
$(18, 3, 30)$	45	0.9%	1.5%	1.5%	1.5%
$(18, 3, 60)$	35	1.0%	1.0%	1.5%	1.5%
$(32, 5, 3)$	55	0.9%	1.1%	1.2%	1.2%
$(32, 5, 5)$	20	0.9%	0.9%	1.2%	1.2%
$(32, 5, 15)$	25	0.8%	0.8%	0.8%	0.9%
$(32, 5, 30)$	22	0.9%	1.2%	1.2%	1.2%
$(32, 5, 60)$	32	0.8%	1.1%	1.1%	1.1%
$(48, 7, 3)$	25	0.8%	0.8%	1.1%	1.3%
$(48, 7, 5)$	15	1.0%	1.1%	1.1%	1.1%
$(48, 7, 15)$	15	0.6%	0.6%	0.9%	0.9%
$(48, 7, 30)$	15	0.9%	1.1%	1.1%	1.2%
$(48, 7, 60)$	15	0.9%	1.4%	1.4%	1.4%
$(64, 9, 3)$	15	1.0%	1.0%	1.1%	1.3%
$(64, 9, 5)$	15	0.6%	0.6%	0.7%	0.9%
$(64, 9, 15)$	15	0.6%	0.6%	0.6%	0.6%
$(64, 9, 30)$	15	0.8%	0.9%	0.9%	1.2%
$(64, 9, 60)$	15	1.0%	1.0%	1.1%	1.1%
average stab.		0.89%	1.06%	1.15%	1.22%

distribution estimations too much. But of course, two links (in time and space) that are incorrectly correlated in the scenario set may be on the same route, and hence, the distribution of travel times on the route will be affected (though the mean is always correct).

2. We also see that we need more scenarios when K (and hence N) and P are low. The reason is simple enough: All routes, but to a different degree, will be affected by spurious correlations. With a low K and / or P, the effect of a single correlation becomes more important. Since stability is defined as a maximal difference over many scenario sets, this max will therefore tend to increase as the number of routes and periods decrease.

3. What were the effects of the spurious correlations on the number of scenarios needed for stability? First note two facts: Firstly, as the number of scenarios increases, the spurious correlations will change. They may converge to something, or they may not. But in any case, there is no reason why they should converge to the right values. Hence, the noise from correlations will remain in the results. Secondly, as we do not know what the uncontrolled correlations are (or could be),

Table 7: The out-of-sample stability levels for solutions from Table 4.

	$P = 3$		$P = 5$		$P = 15$		$P = 30$		$P = 60$	
(N, K)	$\|S\|$	SL	$\|S\|$	SL	$\|S\|$	SL	$\|S\|$	SL	$\|S\|$	SL
$(18, 3)$	100	1.7%	60	1.2%	65	0.8%	45	1.2%	35	1.3%
$(32, 5)$	55	1.3%	20	1.2%	25	1.0%	22	1.1%	32	0.8%
$(48, 7)$	25	0.6%	15	1.1%	15	0.6%	15	0.9%	15	1.1%
$(64, 9)$	15	0.5%	15	0.6%	15	0.4%	15	0.6%	15	0.8%

we cannot compare the correlations from the scenarios (which we can find) with anything meaningful. So they will either converge to something or keep jumping around, but we will never get rid of the noise. What we observe is that there are errors in the stability tests that come from the correlations, and these errors are (except for some noise) independent of the number of scenarios, but larger the smaller are K and P.

Hence, when we perform the stability tests, and we measure errors, these errors stem from correlations and scenarios. With the errors from correlations more or less fixed for a given pair (K, P), it becomes harder to achieve a certain total error by increasing the number of scenarios if the errors from the correlations are large. Hence, when correlation errors are large, we need many more scenarios to achieve a certain total errors. And this explains the relationship between problem size and necessary scenarios.

4. Despite the, at times, enormous dimension of our problem, each route actually covers rather few random variables, with a vehicle traveling on twenty to thirty links while visiting six to eight customers. The number of random variables involved increases with P, but the increase is purely caused by adding many speeds representing the same link in neighboring periods – and those are the ones we control the most. Assume, just for the sake of argument, that we knew exactly which random variables were involved in a route, and we generated scenarios for just those variables. It is maybe not so surprising that rather few scenarios are needed in such a moderate dimension. The clue of the scenario generation method we use is that it produces good scenarios for any possible route. In a sense, that is why it takes so long to find just 15 scenarios (see next section), as the scenario set must work for all routes at the same time. So, it is not always true that the number of scenarios will *likely be quite large*, as expressed by Gendreau et al. (2016). The reason is that in a VRP we end up with routes (of course) which have a very nice structure relative to our scenario generation heuristic. So the number of random variables can be huge, but typically not the number of scenarios. In detail, a route can be seen as a sequence of link pairs, and we control all the correlations representing link pairs. It means that even though we control rather few correlations (relative to the total number) we can focus on the ones that we know will be most important. This may not be true for other scenario generation

methods, and certainly not for other optimization problems that lack this useful structure. It is because, in problems where it is not clear which correlations will be important, it is also not clear which limited subset of correlations to handle in the scenario generation method. And the result will be that crucial correlations are not controlled.

Other VRPs. To investigate if our approach works for other VRPs, we further perform stability tests for eight typical VRPs with different constraints and objectives. This includes vehicle capacity constraints, soft route duration constraints, hard route duration constraints, earliest arrival time constraints, and latest arrival time constraints. The objectives involve such as maximizing the expected number of served customers, minimizing the expected total costs, and minimizing the expected total travel costs. To keep the main text short, we put the details of these VRPs and the corresponding stability tests in the Appendix. The results show that fifteen scenarios are sufficient to achieve a stability level of 1% in almost all problem instances.

7.2. Computation Time

The CPU time for solution evaluations consists of two parts. The first part is the CPU time of the scenario generation, measured in seconds, which we denote by T_1. Note that this is run only once before the start of the search heuristic. The scenarios can be reused for any instances as long as the speed patterns do not change.

The second part, denoted T_2, is the average CPU time (in seconds) needed to evaluate the objective function for one scenario (for a given feasible solution). We compute this by generating 5,000 feasible solutions for each of the 20 problem instances, and report the average CPU times. These results are shown in Table 8. The tests were carried out on a laptop with Intel Core i7-5500U CPU @2.4GHz and 8 GB RAM using MATLAB version R2009a.

Let us first look at T_1 in Table 8. The number of random variables is $418 \times P$. So for $P = 3$ we have 1254 random variables, up to 25,080 random variables for $P = 60$. For a certain $|S|$, T_1 increases with P, and the increase is much more than linear as the handling of dependencies becomes increasingly complicated. This is also partly caused by memory and disk I/O operations on our laptop. The largest T_1 in the table is 21,287 seconds, i.e., about 6 hours. A more descriptive case in terms of algorithmic behavior (not affected by memory operations) can be found for $P = 30$. This could, for example, correspond to a 10-hour day splitting into thirty 20-minute intervals. In this case it took about 37 minutes to create 15 scenarios. It might seem excessive to use 37 minutes on fifteen scenarios. But remember that these fifteen scenarios involve over 188,000 different numbers, and it is the carefulness of setting these up that makes it possible to have so few scenarios. There are certainly quicker ways to set up fifteen scenarios, but the question will then be if such stability can be achieved.

Once a scenario set is defined and found stable, it is T_2 that determines the speed of solving the stochastic VRP. It can be found from Table 8 that T_2 is approximately linear in N and K. Given a route and an evaluation algorithm, T_2 is determined mainly

Table 8: CPU times in seconds needed for stochastic instances.

	$P=3$			$P=5$			$P=15$			$P=30$			$P=60$		
(N,K)	$\lvert S\rvert$	T_1	T_2	$\lvert S\rvert$	T_1	T_2	$\lvert S\rvert$	T_1	T_2	$\lvert S\rvert$	T_1	T_2	$\lvert S\rvert$	T_1	T_2
$(18,3)$	100	160	0.008	60	113	0.008	65	659	0.008	45	2557	0.008	35	21287	0.008
$(32,5)$	55	51	0.014	20	25	0.015	25	342	0.015	22	2273	0.015	32	18097	0.015
$(48,7)$	25	15	0.021	15	20	0.021	15	294	0.021	15	2240	0.022	15	17198	0.022
$(64,9)$	15	9	0.028	15	20	0.028	15	294	0.029	15	2240	0.029	15	17198	0.029

by the number of calls to the Dijkstra algorithm (which is very close to the number of customer nodes) and the number of links included in the route. The sets of 5000 feasible solutions are the same at different P in our experiments. Thus, the average number of links and the average number of Dijkstra calls are not dependent on P. That is why T_2 at different P are almost the same for given N and K. Note that we have not optimized the computations reflected in Table 8, as that is not the point of this paper; our goal is to show a useful approach. More can be achieved by for example using a faster language such as C/C++, parallel evaluation of routes/scenarios, and caching of already evaluated routes.

But in any case, since evaluating the objective function in the deterministic case corresponds to the evaluation for one scenario, the number of scenarios is what determines how much slower one function evaluation is in the stochastic case compared to the deterministic one. And this is where, in our view, 15 is a very acceptable number. This is why it is so critical to have a good method for generating scenarios so that stability (at a chosen accuracy) can be achieved with as few scenarios as possible. Our scenario generation method is particularly well suited to VRPs. Of course, the setup of scenarios is rather expensive, but this can be done offline and is valid as long as the map and the speed patterns do not change.

8. Conclusions

This paper provides a starting point for handling correlated travel times (speeds) in VRPs. Correlations cover both time and space. We point out that such VRPs need to be defined on the underlying map (network) and not on just the customer nodes; correlations in time and space cannot be properly expressed in that setting. The main reason is that as soon as two node pairs share a link the speeds become dependent. This is even more involved when we face multiple time periods.

Next, we point out that VRPs with stochastic speeds easily bring the number of dependent random variables into the thousands. Our largest case had 25,080 random variables. In such dimensions it is close to impossible to define and represent the full correlation matrix. In our largest case, there are, for example, over 310 million distinct correlations. As a result of this problem of size, sampling, as a way to generate scenarios, is also out as methodology in most cases, as sampling requires something to sample from.

We chose to circumvent the above problem by not defining a complete distribution. Rather, we focus on the dependencies that are most critical for a VRP, namely between speeds on the same and neighboring links in neighboring periods as well as neighboring links in the same period; a route will always pick a sequence of links (random variables) visiting such neighbors. This approach will create spurious correlations for some unspecified road link pairs. Whether or not this is a major problem is tested as part of the procedure for determining the necessary number of scenarios. We show that stability can indeed be achieved, partly caused by the fact that our scenario generation method always creates correct expected speeds, and hence expected travel times. Typical numbers we get is that for a case where the expected behavior of the stochastic solution is 10% better than that of the corresponding deterministic solution, we can obtain solutions with a 1% error using 15 scenarios in dimensions as high as 25,080. This very low number of scenarios is a result of our scenario generation method and cannot be carried over to other approaches without testing.

We provide a simple way to calculate the objective function value for any feasible solution (set of routes). This can be used in most cases, somewhat adjusted based on the VRP being studied. If $P = 1$, i.e., there is no time dimension, any existing deterministic approach can be used, scenario by scenario. The number of necessary scenarios shows how much more running time is needed for a stochastic case, compared to its deterministic counterpart. For a case with 142 nodes, 418 road links, we found a ratio of about 15. The substantial time needed to set up the scenarios comes in addition, but these can be reused as long as the map and the speed distributions do not change.

It is our hope that others can use our general approach to investigate different VRPs with correlated travel times (speeds). The code used to generate scenarios is available from its author for free. Future work can also focus on developing fast and effective heuristics for stochastic VRPs based on our solution evaluation approach.

Acknowledgments

The authors would like to thank the financial supports from the National Natural Science Foundation of China (Grant No. 71872118), the MOE (Ministry of Education in China) Project of Humanities and Social Sciences (Grant No. 18YJC630045) and Sichuan University (Grant No.s 2018hhs-37, SKSYL201819). We are also in debt to Tom Van Woensel, the associate editor and three anonymous referees for extremely valuable input.

Appendix

A. Stability tests for different VRP cases

In this appendix, we present stability tests for several stochastic VRPs with typical constraints and objective functions so that the effectiveness of our approach can be validated further.

A.1. More VRP cases

There exist various constraints and objectives in VRPs. The following are used in our tests.

(1) **Vehicle capacity constraint**: The total delivery on a route must not exceed the vehicle capacity Q. Let q_i denote the delivery at customer i, r_k the route of vehicle k, and D_k the total amount of freight delivered on route r_k. We have

$$D_k = \sum_{i \in r_k} q_i \leq Q \tag{1}$$

(2) **Soft route duration constraint**: If the total travel time of a route is greater than the route duration limit, the vehicle can continue working but an overtime pay is incurred. Let T_k^s denote the total travel time of route r_k in scenario s ($1 \leq s \leq S$), T^{max} the route duration limit, and O_k^s the overtime of route r_k in scenario s. We have,

$$O_k^s = \max(T_k^s - T^{max}, 0) \tag{2}$$

(3) **Hard route duration constraint**: For route r_k, once the remaining time, using expected speeds, is insufficient to serve the next customer in scenario s, the vehicle stops working and returns to the depot. The remaining time will end up being T^{max} minus the travel time using the speeds of the scenario. But this is not known at the time of leaving customer i since future speeds are not actually known. Assume that i and j are the last two customers visited on route r_k. Let t_{ijo}^{ms} (t_{ijo}^{m0}) denote the scenario-based (expected) travel time from customer i to customer j to the depot starting at time m in scenario s. The total travel time (T_k^s) of route r_k could exceed the route duration limit if the realized travel time t_{ijo}^{ms} to serve the last customer is larger than its expected travel time t_{ijo}^{m0}. We have

$$T_k^s - t_{ijo}^{ms} + t_{ijo}^{m0} \leq T^{max} \tag{3}$$

(4) **Soft earliest arrival time constraint**: If the vehicle's arrival time is less than the specified earliest arrival time at a customer, an earliness penalty cost is incurred. Let a_i^s denote the actual arrival time at customer i in scenario s, and e_i the required earliest arrival time at customer i. The earliness E_i^s of customer i in scenario s is expressed below,

$$E_i^s = \max(e_i - a_i^s, 0) \tag{4}$$

(5) **Soft latest arrival time constraint**: If the vehicle's arrival time is greater than the specified latest arrival time at a customer, a tardiness penalty cost is incurred. Let l_i denote the required latest arrival time at customer i. The tardiness L_i^s of customer i in scenario s is expressed below,

$$L_i^s = \max(a_i^s - l_i, 0) \tag{5}$$

To examine the effects of different objectives and constraints on stability and the number of scenarios, we use VPRs with one or more constraints from above and different objectives. The detail of these cases are described as follows.

Case 1: The objective is to maximize the expected number of served customers with a vehicle capacity constraint and a hard route duration constraint. Some customers cannot be served due to the vehicle capacity constraint and some due to the hard route duration constraint. Let N_k^s denote the number of customers visited on route r_k in scenario s. The objective can be formulated as follows.

$$\max F1 = E(\sum_k (N_k^s)) \tag{6}$$

Case 2: Following Case 1, this case considers the same vehicle capacity constraint but a soft route duration constraint. The objective is to maximize the expected total profit (revenue). Let ψ denote the profit per unit freight, w_o the overtime pay per hour, w_f the cost per unit fuel (unit: liter), and f_k^s the total fuel consumption of route r_k in scenario s. The fuel consumption is a nonlinear function of travel distance and travel time[5]. The objective can be formulated as follows.

$$\max F2 = E(\sum_k (\psi \cdot D_k - w_o \cdot O_k^s - w_f \cdot f_k^s)) \tag{7}$$

Case 3: Following Case 2, this case considers the same constraints. The objective is also to maximize the expected total profit (revenue). But we do not consider the fuel cost in this objective, which is formulated as follows

$$\max F3 = E(\sum_k (\psi \cdot D_k - w_o \cdot O_k^s)) \tag{8}$$

Case 4: All customers need to be served and each customer must be served once by exactly one vehicle. The objective is to minimize the expected total cost with the vehicle capacity constraint and the soft route duration constraint. The expected total cost is the summation of expected transportation costs, including the fixed cost of vehicles used, the overtime cost, and the fuel-related travel cost. Let ϕ denote the fixed operational cost of one vehicle. Let z_k be 1 if vehicle k is used; otherwise it is 0. The objective can be formulated as follows.

$$\min F4 = E(\sum_k ((\phi + w_o \cdot O_k^s + w_f \cdot f_k^s) \cdot z_k)) \tag{9}$$

Case 5: Following Case 4, the objective is to minimize the expected total cost with the vehicle capacity constraint and the soft route duration constraint. The expected total cost is the summation of expected transportation costs, including the fixed cost of vehicles used, the overtime cost, and the distance-related travel cost. Let d_k^s denote the distance travelled in scenario s, and w_d the travel cost per unit distance. The objective can be formulated as follows.

$$\min F5 = E(\sum_k ((\phi + w_o \cdot O_k^s + w_d \cdot d_k^s) \cdot z_k)) \tag{10}$$

[5]Refer to: Barth, M., Younglove, T., Scora, G. (2005). Development of a heavy-duty diesel modal emissions and fuel consumption model. California PATH research report.

Case 6: All customers must be served and a fixed number of vehicles are available. The objective is to minimize the expected total travel cost with the vehicle capacity constraint and the soft latest arrival time constraint. The total travel cost is defined as the summation of the travel time-related cost and the tardiness penalty cost. Let γ denote the travel cost per unit time and let w_l denote the tardiness penalty per unit time. The objective can be formulated as follows.

$$\min F6 = E(\sum_k (\gamma \cdot T_k^s + \sum_{i \in r_k} (w_l \cdot L_i^s)) \tag{11}$$

Case 7: This case is similar to Case 6. The only difference is that this case considers the earliness penalty cost in the objective function instead of the tardiness penalty cost. Let w_e denote the earliness penalty per unit time. The objective can be formulated as follows.

$$\min F7 = E(\sum_k (\gamma \cdot T_k^s + \sum_{i \in r_k} (w_e \cdot E_i^s)) \tag{12}$$

Case 8: Following Case 6 and Case 7, the objective function in this case considers both the earliness penalty cost and the tardiness penalty cost, which is formulated as follows.

$$\min F8 = E(\sum_k (\gamma \cdot T_k^s + \sum_{i \in r_k} (w_e \cdot E_i^s + w_l \cdot L_i^s)) \tag{13}$$

In Cases 1-5, either the number of customers visited or the number of vehicles used needs to be determined in the decision process. Customers visited and the corresponding visiting sequence in a route could affect customers visited in other routes in Cases 1-3 and affect the number of used vehicles in Cases 4-5. However, both the number of customers visited and the number of vehicles used are pre-specified in Cases 6-8.

A.2. Numerical results

To observe how these different types of constraints and objectives affect the stability, we conduct the stability tests in terms of 20 problem instances, for each test case, based on different value combinations of N, K, and P. For each case, P could be 3, 5, 15, 30 or 60. The (N, K) combinations in different cases are shown in Table A1. In Cases 1-5, either N or K could be different from the corresponding one in the case of Section 6. Cases 1-3 consider more customers because these cases need to select out the most appropriate customers from all customers. In Cases 4-5, Ns are the same as in Section 6, but K is not pre-specified. In Cases 6-8, (N, K) could be (18, 3), (32, 5), (48, 7), or (64, 9).

The weights in the experiments are set as follows: $\psi = 1$, $\phi = 100$, $\gamma = 1$, $w_d = 1$, $w_e = 10$, $w_f = 1$, $w_l = 20$, and $w_o = 10$. The settings of other experimental data are consistent with those in the main text.

Using the same stability test method as described in Section 5, we examine the resulting relative differences at $|S|$=15 for 20 problem instances of each test case. Table A2 shows the resulting relative differences. From the results in Table A2, we find that:

Table A1: (N,K) combinations in different cases

Cases	(N^1,K^1)	(N^2,K^2)	(N^3,K^3)	(N^4,K^4)
1-3	(27,3)	(48,5)	(64,7)	(80,9)
4-5	(18,K)	(32,K)	(48,K)	(64,K)
6-8	(18,3)	(32,5)	(48,7)	(64,9)

Table A2: The relative differences at $|S|$=15

N,K,P	Case 1	Case 2	Case 3	Case 4	Case 5	Case 6	Case 7	Case 8
$(N^1, K^1, 3)$	2.73%	1.46%	0.51%	1.46%	0.42%	2.20%	2.92%	0.97%
$(N^1, K^1, 5)$	2.49%	0.72%	0.36%	0.72%	0.15%	1.06%	1.69%	1.04%
$(N^1, K^1, 15)$	1.92%	0.27%	0.25%	0.27%	0.11%	0.85%	1.38%	0.54%
$(N^1, K^1, 30)$	2.95%	1.36%	0.49%	1.36%	0.28%	1.53%	3.09%	0.85%
$(N^1, K^1, 60)$	2.91%	2.04%	0.73%	2.04%	0.21%	1.14%	2.01%	2.95%
$(N^2, K^2, 3)$	1.52%	1.02%	0.43%	1.02%	0.10%	1.44%	1.06%	0.99%
$(N^2, K^2, 5)$	1.11%	0.64%	0.34%	0.64%	0.08%	0.58%	1.81%	0.97%
$(N^2, K^2, 15)$	0.25%	0.60%	0.28%	0.60%	0.15%	0.80%	0.81%	0.44%
$(N^2, K^2, 30)$	0.56%	0.84%	0.55%	0.84%	0.05%	0.59%	1.29%	0.81%
$(N^2, K^2, 60)$	0.80%	1.37%	0.58%	1.37%	0.19%	1.17%	1.97%	1.06%
$(N^3, K^3, 3)$	0.94%	0.88%	0.36%	0.88%	0.22%	0.81%	1.60%	1.10%
$(N^3, K^3, 5)$	1.09%	0.24%	0.13%	0.24%	0.13%	0.66%	1.72%	0.47%
$(N^3, K^3, 15)$	0.92%	0.58%	0.31%	0.58%	0.11%	0.41%	0.78%	0.36%
$(N^3, K^3, 30)$	0.93%	0.73%	0.37%	0.73%	0.24%	0.72%	1.26%	0.48%
$(N^3, K^3, 60)$	1.07%	0.68%	0.44%	0.68%	0.27%	0.89%	2.03%	0.68%
$(N^4, K^4, 3)$	0.53%	0.89%	0.36%	0.89%	0.16%	0.46%	0.54%	0.40%
$(N^4, K^4, 5)$	0.52%	0.51%	0.20%	0.51%	0.06%	0.38%	0.24%	0.26%
$(N^4, K^4, 15)$	0.81%	0.30%	0.16%	0.30%	0.11%	0.50%	0.60%	0.56%
$(N^4, K^4, 30)$	0.64%	0.68%	0.20%	0.68%	0.12%	0.29%	0.67%	0.30%
$(N^4, K^4, 60)$	0.34%	0.55%	0.28%	0.55%	0.20%	0.94%	0.65%	0.75%

1. Using 15 scenarios, we can achieve a stability level of 2% or 1% for most problem instances.
2. For the last five large problem instances of all test cases, fifteen scenarios are sufficient to achieve a stability level of 1%.
3. Compared to Case 3, Case 2 leads to a larger RD, which is caused by the fuel cost term in its objective since it is the only difference between the two cases. It implies that the weight setting for different terms in a polynomial objective functions has effects on the stability results since the objective function $F3$ is actually a special case of the objection function $F2$. Just to illustrate, take Case 4 as an example. If we change w_o to 500 from 10 (an extremely large change) in $F4$, the resulting RD range for the 20 problem instances will be changed to [1.11%, 16.64%] from the current range [0.27%, 2.04%].

References

Bent RW, Van Hentenryck P (2004) Scenario-based planning for partially dynamic vehicle routing with stochastic customers. *Operations Research* 52(6):977–987.

Beyer HG, Schwefel HP (2002) Evolution strategies: A comprehensive introduction. *Natural Computing* 1(1):3–52.

Campbell AM, Wilson JH (2014) Forty years of periodic vehicle routing. *Networks* 63(1):2–15.

Casey B, Bhaskar A, Guo H, Chung E (2014) Critical review of time-dependent shortest path algorithms: A multimodal trip planner perspective. *Transport Reviews* 34(4):522–539.

Cattaruzza D, Absi N, Feillet D (2016) The multi-trip vehicle routing problem with time windows and release dates. *Transportation Science* 50(2):676–693.

Cimen M, Soysal M (2017) Time-dependent green vehicle routing problem with stochastic vehicle speed: An approximate dynamic programming algorithm. *Transportation Research Part D: Transport and Environment* 54:82–98.

Dupačová J, Consigli G, Wallace SW (2000) Scenarios for multistage stochastic programs. *Annals of Operations Research* 100(1–4):25–53, ISSN 0254-5330, URL http://dx.doi.org/10.1023/A:1019206915174.

Ehmke JF, Campbell AM, Thomas BW (2016) Vehicle routing to minimize time-dependent emissions in urban areas. *European Journal of Operational Research* 251(2):478–494, ISSN 0377-2217, URL http://dx.doi.org/10.1016/j.ejor.2015.11.034, 9.

Ehmke JF, Campbell AM, W TB (2018) Optimizing for total costs in vehicle routing in urban areas. *Transportation Research Part E: Logistics and Transportation Review* 116:242–265, URL http://dx.doi.org/10.1016/j.tre.2018.06.008.

Errico F, Desaulniers G, Gendreau M, Rei W, Rousseau LM (2016) A priori optimization with recourse for the vehicle routing problem with hard time windows and stochastic service times. *European Journal of Operational Research* 249(1):55–66.

Feng Y, Zhang RQ, Jia G (2017) Vehicle routing problems with fuel consumption and stochastic travel speeds. *Mathematical Problems in Engineering* 2017:1–16, URL http://dx.doi.org/10.1155/2017/6329203.

Fisher ML, Jaikumar R (1981) A generalized assignment heuristic for vehicle routing. *Networks* 11(2):109–124.

Fu L, Rilett L (1998) Expected shortest paths in dynamic and stochastic traffic networks. *Transportation Research Part B: Methodological* 32:499–516, URL http://dx.doi.org/10.1016/S0191-2615(98)00016-2.

Gendreau G, Jabali O, Rei W (2016) Future research directions in stochastic vehicle routing. *Transportation Science* 50(4):1163–1173.

Goodson JC (2015) A priori policy evaluation and cyclic-order-based simulated annealing for the multi-compartment vehicle routing problem with stochastic demands. *European Journal of Operational Research* 241(2):361–369.

Han J, Lee C, Park S (2014) A robust scenario approach for the vehicle routing problem with uncertain travel times. *Transportation Science* 48(3):373–390.

Huang Y, Zhao L, Woensel TV, Gross JP (2017) Time-dependent vehicle routing problem with path flexibility. *Transportation Research Part B: Methodological* 95:169–195.

Hui YV, Gao J, Leung L, Wallace S (2014) Airfreight forwarder's shipment planning under uncertainty: A two-stage stochastic programming approach. *Transportation Research Part E: Logistics and Transportation Review* 66:83–102.

Ichoua S, Gendreau M, Potvin JY (2003) Vehicle dispatching with time-dependent travel times. *European Journal of Operations Research* 144:379–396.

Juan AA, Faulin J, Jorba J, Caceres J, Marques JM (2013) Using parallel & distributed computing for real-time solving of vehicle routing problems with stochastic demands. *Annals of Operations Research* 207(1):43–65.

Kaut M (2014) A copula-based heuristic for scenario generation. *Computational Management Science* 11(4):503–516, URL http://dx.doi.org/10.1007/s10287-013-0184-4.

Kaut M, Wallace SW (2007) Evaluation of scenario-generation methods for stochastic programming. *Pacific Journal of Optimization* 3(2):257–271.

King AJ, Wallace SW (2012) *Modeling with Stochastic Programming*. Springer Series in Operations Research and Financial Engineering (Springer), URL http://dx.doi.org/10.1007/978-0-387-87817-1.

Laporte G (2009) Fifty years of vehicle routing. *Transportation Science* 43(4):408–416.

Laporte G, Louveaux F, Mercure H (1992) The vehicle routing problem with stochastic travel times. *Transportation Science* 26(3):161–170.

Lecluyse C, Sörensen K, Peremans H (2013) A network-consistent time-dependent travel time layer for routing optimization problems. *European Journal of Operational Research* 226:395–413.

Lee C, Lee K, Park S (2012) Robust vehicle routing problem with deadlines and travel time/demand uncertainty. *Journal of the Operational Research Society* 63(9):1294–1306.

Lei HT, Laporte G, Guo B (2012) A generalized variable neighborhood search heuristic for the capacitated vehicle routing problem with stochastic service times. *Top* 20(1):99–118.

Lin CH, Choy KL, Ho GTS, Chung SH, Lam HY (2014) Survey of green vehicle routing problem: Past and future trends. *Expert Systems with Applications* 41(4):1118–1138.

Lurie PM, Goldberg MS (1998) An approximate method for sampling correlated random variables from partially-specified distributions. *Management Science* 44(2):203 – 218.

Martinelli R, Contardo C (2015) Exact and heuristic algorithms for capacitated vehicle routing problems with quadratic costs structure. *INFORMS Journal on Computing* 27(4):658–676.

Mester D, Bräysy O (2007) Active-guided evolution strategies for large-scale capacitated vehicle routing problems. *Computers & Operations Research* 34(10):2964–2975.

Nejad MM, Mashayekhy L, Chinnam RB, Phillips A (2016) Hierarchical time-dependent shortest path algorithms for vehicle routing under ITS. *IIE Transactions* 48(2):158–169.

Oyola J, Arntzen H, Woodruf DL (2018) The stochastic vehicle routing problem, a literature review, part I: Models. *EURO Journal on Transportation and Logistics* (7):193–221.

Pillac V, Gendreau M, Gueret C, Medaglia AL (2013) A review of dynamic vehicle routing problems. *European Journal of Operational Research* 225(1):1–11.

Potvin JY (2009) State-of-the art review: Evolutionary algorithms for vehicle routing. *INFORMS Journal on Computing* 21(4):518–548.

Prins C, Lacomme P, Prodhon C (2014) Order-first split-second methods for vehicle routing problems: A review. *Transportation Research Part C: Emerging Technologies* 40:179–200.

Ritzinger U, Puchinger J, Hartl RF (2016) A survey on dynamic and stochastic vehicle routing problems. *International Journal of Production Research* 54(1):215–231.

Schilde M, Doerner K, Hartl R (2014) Integrating stochastic time-dependent travel speed in solution methods for the dynamic dial-a-ride problem. *European Journal of Operational Research* 238:18–30.

Soysal M, Bloemhof-Ruwaard JM, Bektas T (2015) The time-dependent two-echelon capacitated vehicle routing problem with environmental considerations. *International Journal of Production Economics* 164:366–378.

Tas D, Dellaert N, van Woensel T, de Kok T (2013) Vehicle routing problem with stochastic travel times including soft time windows and service costs. *Computers & Operations Research* 40(1):214–224.

Tas D, Dellaert N, van Woensel T, de Kok T (2014) The time-dependent vehicle routing problem with soft time windows and stochastic travel times. *Transportation Research Part C: Emerging Technologies* 48:66–83.

Ticha HB, Absi N, Feillet D, Quilliot A (2018) Vehicle routing problems with road-network information: State of the art. *Networks* 72(3):393–406, ISSN 1097-0037, URL http://dx.doi.org/10.1002/net.21808, 9.

Vidal T, Crainic TG, Gendreau M, Prins C (2013) Heuristics for multi-attribute vehicle routing problems: A survey and synthesis. *European Journal of Operational Research* 231(1):1–21.

Vidal T, Crainic TG, Gendreau M, Prins C (2014) A unified solution framework for multi-attribute vehicle routing problems. *European Journal of Operational Research* 234(3):658–673.

Voudouris C, Tsang E (1998) Guided local search. *European Journal of Operations Research* 113:80–119.

Wang XY, Golden B, Wasil E (2015) The min-max multi-depot vehicle routing problem: Heuristics and computational results. *Journal of the Operational Research Society* 66(9):1430–1441.

Zhao Y, Wallace SW (2014) Integrated facility layout design and flow assignment problem under uncertainty. *INFORMS Journal on Computing* 26(4):798–808.

Zhu L, Rousseau LM, Rei W, Li B (2014) Paired cooperative reoptimization strategy for the vehicle routing problem with stochastic demands. *Computers & Operations Research* 50:1–13.