The Sample Average Approximation Method applied to Routing Problems with Random Travel Times

Erasmus University Rotterdam
School of Economics

Bachelor Thesis Quantitative Logistics & Operations Research

Author
Gerben Bergsma
435717

Supervisor
Thomas R. Visser MSc.

Second Assesor
Dr. R. Spliet

July 8, 2018

Abstract

In this thesis most parts of the computational study of Verweij et al. (2003) on the Sample Average Approximation (SAA) method have been replicated. The SAA method uses Monte Carlo simulation to solve stochastic optimization problems. Verweij et al. (2003) investigated the Shortest Path Problem with Random Travel Times (SPRT), the Shortest Path Problem with Random Arc Failures (SPAF) and the Traveling Salesman Problem with Random Travel Times (TSPRT). We have replicated both the SPRT and TSPRT. Additionally, we extended the TSPRT to a Vehicle Routing Problem with Random Travel Times (VRPRT).

Some differences in results are expected as different versions of the same commercial MIP solver CPLEX have been used. Besides, the SAA method makes use of random generated samples, so different samples are expected. Precise instances are also not available, so reconstructing them as close as possible was the only option, but the dimensions turned out to be different.

The results for the SPRT are close to those found by Verweij et al. (2003). However, for the TSPRT the found values are much larger. When comparing the TSPRT and VRPRT results, we see that the difference is quite small. Taking customer satisfaction into account, it might be beneficial to use multiple vehicles. Further research on the VRPRT using different deadlines and different amounts of vehicles could therefore be interesting.

1 Introduction

In this thesis, most parts of the computational study of Verweij et al. (2003) on the Sample Average Approximation (SAA) method applied to stochastic routing problems are replicated. The SAA method uses Monte Carlo simulation in order to solve stochastic optimization problems. In the study of Verweij et al. (2003), multiple problems are investigated. In this thesis, only the shortest path problem with random travel times (SPRT) and the traveling salesman problem with random travel times (TSPRT) are investigated. In addition, we investigate the vehicle routing problem with random travel times (VRPRT) as an extension of the TSPRT.

Applications for the SPRT, TSPRT and VRPRT are found in everyday life. Delivery companies such as PostNL and DHL have to decide on routes up front, while having to deal with uncertainty about possible delays. These delays can be a result of traffic, road work, accidents or more generally congestion. Similarly, for shipping companies, weather conditions could lead to uncertainty about travel times. Applications for the deterministic TSP and VRP are described by Matai et al. (2010).

The stochastic routing problems in this thesis consist of two stages. In the first stage, a route has to be selected. Here, the random travel times are only known with a probability distribution. In the second stage, a decision has to be made based on the first stage route and known delays and a penalty is incurred according to a penalty or recourse function.

Although we try to replicate the results of Verweij et al. (2003) as good as possible, some differences are expected. The SAA method uses randomly generated samples, so it is expected that samples will differ from the ones used by Verweij et al. (2003). Furthermore, differences can occur due to the use of a different version of the commercial MIP solver CPLEX. We have used CPLEX version 12.6.3, while Verweij et al. (2003) have used CPLEX version 7.0. Also, the precise instances used by Verweij et al. (2003) are not available, so we reconstructed them as close as possible.

The remainder of the paper is structured as follows. First, we give a brief literature review in Section 2. Then, in Section 3 we give a description of the three problems investigated in this thesis. Next, in Section 4 we first introduce the SAA method for general two-stage stochastic routing problems. Then we introduce the branch-and-cut framework used to solve the problems and finally we highlight how we solved each problem specifically. After that, in Section 5 we describe how we generated instances to test the SAA method. In Section 6 we describe our results and in Section 7 we give a conclusion.

2 Literature Review

Early solution methods for stochastic routing problems are based on dynamic programming (Andreatta, 1987; Andreatta and Romeo, 1988) and on heuristics (Spaccamela et al., 1984). These methods are however not applicable in general. Cases with a small number of scenarios have been solved by Laporte et al. (1992, 1994b,a) by using the fact that a small number of scenarios allows for exact evaluation of the recourse function. Solving the stochastic routing problems was done by using the integer L-shaped method of Laporte and Louveaux (1993). The integer L-shaped method underestimates the value of the recourse function and iteratively adds optimality cuts to better approximate it. For two-stage stochastic problems where the first-stage feasible set is convex and the second stage involves routing decisions, Wallace (1986, 1987, 1988) has studied the exact evaluation of the recourse function.

For problems where the number of scenarios is large, exact evaluation methods are not applicable. Typically, sampling methods are used to solve such problems. We are able to classify sampling methods into two main groups: interior and exterior sampling methods. Interior sampling methods allow for the samples to be modified during the optimization process, for example by taking subsets of previous samples or generating new samples. Slyke and Wets (1969), Higle and Sen (1991) and Infanger (1992) have, for example, proposed methods that modify samples within the L-shaped algorithm.

In exterior sampling methods, a sample $\{\omega^1, \ldots, \omega^N\}$ is generated according to a known probability distribution up front. After that, a deterministic optimization problem is solved. One of the exterior sampling methods is the SAA method. The SAA method has been used by, for example, Rubinstein and Shapiro (1990) and Geyer and Thompson (1992). Plambeck et al. (1996), Shapiro (1996), Shapiro and Homem-de Mello (1998) and Mak et al. (1999) have proposed SAA methods for stochastic linear programs.

The shortest path problem with random travel times has been investigated previously by Polychronopoulos and Tsitsiklis (1996), where a dynamic programming approach is used to solve the problem. Using the nearest neighbour rule to solve the traveling salesman problem with random travel times has been studied by Leipälä (1978). An overview of literature on stochastic vehicle routing problems is given by Gendreau et al. (1996).

Other routing problems with uncertainty that have been researched include the capacitated vehicle routing problem with unknown demands (Bertsimas, 1992), the capacitated traveling salesman location problem with unknown customer positions (Bertazzi and Maggioni, 2015) and the vehicle routing problem with random travel times, soft time windows and service cost (Taş et al., 2013).

3 Problem Description

In this section we define the problems that are investigated in this thesis. Firstly, we define the shortest path problem with random travel times. Next, we formulate the traveling salesman problem with random travel times. The formulations for these problems are taken from Verweij et al. (2003). Finally, we investigate the vehicle routing problem with random travel times which extends the TSPRT by introducing multiple vehicles.

3.1 Shortest Path Problem with Random Travel Times

The shortest path problem with random travel times (SPRT) is defined on an directed graph $G = (V, A)$. Here, $V = \{v_1, v_2, \ldots, v_n\}$ denotes the set of vertices. In this set, node $v_s \in V$ denotes the sink node and $v_t \in V$ denotes the source node. The set of arcs is denoted by $A = \{(v_i, v_j) : v_i, v_j \in V, i \neq j\}$, with arc costs $c \in \mathbb{R}_+^{|A|}$. The set of scenarios is denoted by Ω. The probability distribution $\mathcal{P}$ of the vector of random travel times $\xi(\omega) \in \mathbb{R}_+^{|A|}$ corresponding to scenario $\omega \in \Omega$ is known, as is the deadline $\kappa \in \mathbb{R}$. The goal is to find a $v_s - v_t$ path that minimizes traversed arc cost plus the expected violation of the deadline.

We make use of the binary decision vector $x \in \{0,1\}^{|A|}$ to represent a path. If arc a is in the path $x_a = 1$, otherwise $x_a = 0$. Let $\delta_G^+(v_i)$ denote the set of arcs in G leaving vertex $v_i \in V$. Similarly, let $\delta_G^-(v_i)$ denote the set of arcs in G entering vertex v_i. For any $S \subseteq V$, let $\gamma_G(S)$ denote the set of arcs in G that have both endpoints in S. For any $x \in \mathbb{R}^{|A|}$ and $A' \subseteq A$, $x(A')$ denotes $\sum_{x \in A'} x_a$. The SPRT can then be formulated as the following two-stage stochastic integer program:

$$\min_{x\in\{0,1\}^{|A|}} c^T x + \mathbb{E}_{\mathcal{P}}[Q(x,\xi(\omega))] \tag{1}$$

$$\text{subject to} \quad x(\delta_G^+(v_i)) - x(\delta_G^-(v_i)) = \begin{cases} 1, & \text{for } v_i = v_s \\ -1, & \text{for } v_i = v_t \\ 0, & \text{for } v_i \in V \setminus \{v_s, v_t\} \end{cases} \tag{2}$$

where the recourse function $Q(x,\xi(\omega))$ is given by

$$Q(x,\xi(\omega)) = \max\{\xi(\omega)^T x - \kappa, 0\}. \tag{3}$$

The objective function (1) minimizes total expected cost. Constraints (2) ensure appropriate flow over the arcs. In our case, we have a positive cost-vector c. If c is non-positive and G contains negative-cost cycles, the following constraints need to be added to the model:

$$x(\gamma_G(S)) \leq |S| - 1, \quad \text{for all } S \subset V \text{ with } |S| \geq 2. \tag{4}$$

These constraints ensure negative-cost cycles are eliminated.

3.2 Traveling Salesman Problem with Random Travel Times

The traveling salesman problem with random travel times (TSPRT) is defined on an undirected graph $G = (V,E)$. The set $V = \{v_1, v_2, \ldots, v_n\}$ denotes the set of vertices. The set $E = \{(v_i, v_j) : v_i, v_j \in V, i < j\}$ denotes the set of edges with edge costs $c \in \mathbb{R}_+^{|E|}$. The set of scenarios is denoted by Ω. The probability distribution $\mathcal{P}$ of the vector of random travel times $\xi(\omega) \in \mathbb{R}_+^{|E|}$ corresponding to scenario $\omega \in \Omega$ is known, as is the deadline $\kappa \in \mathbb{R}$. The goal is to find a cycle visiting all nodes in G exactly once, while minimizing total traversed edge costs plus the expected violation of the deadline.

We make use of the binary decision vector $x \in \{0,1\}^{|E|}$ to represent the cycle, where $x_e = 1$ if edge e is in the cycle and $x_e = 0$ otherwise. The set of edges incident to vertex v_i is denoted by $\delta_G(v_i)$. The TSPRT can be formulated as the following two-stage stochastic integer program:

$$\begin{aligned} \min \quad & c^T x + \mathbb{E}[Q(x,\xi(\omega))] && && (5) \\ \text{subject to} \quad & x(\delta_G(v_i)) = 2, && \text{for } v_i \in V, && (6) \\ & x(\gamma_G(S)) \leq |S| - 1, && \text{for } S \subset V \text{ with } |S| \geq 2, && (7) \\ & x \in \{0,1\}^{|E|}, && && (8) \end{aligned}$$

where the recourse function $Q(x,\xi(\omega))$ is given by

$$Q(x,\xi(\omega)) = \max\{\xi(\omega)^T x - \kappa, 0\}. \tag{9}$$

The objective function (5) minimizes the total expected cost. Constraints (6) ensure every node is visited exactly once. Constraints (7) ensure sub-tours are eliminated.

3.3 Vehicle Routing Problem with Random Travel Times

The vehicle routing problem with random travel times (VRPRT) is an extension of the TSPRT, in which multiple vehicle are available. The VRPRT is defined on a directed graph $G = (V,A)$. The set $V = V_c \cup \{0\}$ is the set of nodes, with $V_c = \{1, 2, \dots, n\}$. Node 0 represents the depot. The set $A = \{(v_i, v_j) : v_i, v_j \in V, i \neq j\}$ is the set of arcs with arc costs $c \in \mathbb{R}_+^{|E|}$. The set of scenarios is denoted by Ω. The probability distribution $\mathcal{P}$ of the vector of random travel times $\xi(\omega) \in \mathbb{R}_+^{|E|}$ corresponding to scenario $\omega \in \Omega$ is known. Let there be R identical vehicles. For each vehicle $r \in \{1, \dots, R\}$, there is a deadline $\kappa_r \in \mathbb{R}$. The goal is to find at most R routes starting and ending at the depot in such a way that all nodes in V_c are visited exactly once, while minimizing the total traversed arc costs plus the expected violation of the deadline of each vehicle.

We make use of the binary decision variable $x_{ijr} \in \{0, 1\}$, where $x_{ijr} = 1$ if arc (v_i, v_j) is traversed by vehicle r and $x_{ijr} = 0$ otherwise. The VRPRT can be formulated as the following two-stage stochastic integer program:

$$
\begin{aligned}
\min \quad & \sum_{r=1}^{R} \sum_{(i,j)\in E} c_{ij} x_{ijr} + \mathbb{E}[Q(x, \xi(\omega))] && && (10)\\
\text{subject to} \quad & \sum_{r=1}^{R} \sum_{i=0, i\neq j}^{n} x_{ijr} = 1 && \forall j \in \{1, \dots, n\}, && (11)\\
& \sum_{j=1}^{n} x_{0jr} \leq 1 && \forall r \in \{1, \dots, R\}, && (12)\\
& \sum_{i=0, i\neq j}^{n} x_{ijr} - \sum_{i=0, i\neq j}^{n} x_{jir} = 0 && \forall j \in \{0, \dots, n\}, r \in \{1, \dots, R\}, && (13)\\
& \sum_{i\in S} \sum_{j\in S, i\neq j} x_{ijr} \leq |S| - 1 && \forall S \subseteq \{1, \dots, n\}, \forall r \in \{1, \dots, R\}, && (14)\\
& x_{ijr} \in \{0, 1\}, && && (15)
\end{aligned}
$$

where the recourse function $Q(x, \xi(\omega))$ is given by

$$
Q(x, \xi(\omega)) = \sum_{r=1}^{R} \max\{\xi_r(\omega)^T x_r - \kappa_r, 0\}. \tag{16}
$$

The objective function (10) minimizes the total expected travel cost. Constraints (11) ensure that each customer is visited exactly once by one vehicle. Flow constraints (12) and (13) guarantee that each vehicle can leave the depot only once and that the number of vehicles arriving at a node equals the number leaving. The inequality in constraints (12) makes it possible for vehicles to not be used if this is more cost efficient. The sub-tour elimination constraints (14) ensure no cycles disconnected from the depot.

This formulation is an adaptation of the three-index formulation for the capacitated vehicle routing problem described by Borcinova (2017). We have removed the capacity constraints and added stochasticity. While using a two-index formulation might give better performance, we have decided to use this three-index formulation. By using this three-index formulation, it is easy to keep track of the edges traversed by each vehicle, which simplifies the recourse function.

4 Solution Method

In this section we introduce the Sample Average Approximation (SAA) method as described by Verweij et al. (2003). First, we introduce the SAA method for generic two-stage stochastic routing problems. Then we introduce the branch-and-cut framework used to solve the stochastic routing problems. Finally, we will highlight how we used it to solve the SPRT, TSPRT and VRPRT specifically.

4.1 The Sample Average Approximation Method

A generic formulation for a two-stage stochastic routing problem is

$$z^* = \min_{x \in X} c^T x + \mathbb{E}_{\mathcal{P}}[Q(x, \xi(\omega))], \tag{17}$$

where the recourse function $Q(x, \xi(\omega))$ is given by

$$Q(x, \xi(\omega)) = \min_{y \geq 0} \{q(\omega)^T y | Dy \geq h(\omega) - T(\omega)x\}. \tag{18}$$

Here, the decision vector x denotes the first-stage decision. The first-stage feasible set is denoted by X and the set of scenarios is denoted by Ω. When making the second-stage recourse decision y the scenarios are known. Cost vector c represents the routing costs. It is assumed that the probability distribution $\mathcal{P}$ on Ω is known. $Q(x, \xi(\omega))$ represents the optimal value of the second-stage problem corresponding to the first-stage solution x and second-stage parameters $\xi(\omega) = (q(\omega), h(\omega), T(\omega))$ used for the recourse decision problem, which might depend on scenario $\omega \in \Omega$.

The SAA method solves such stochastic routing problems by generating a sample of N scenarios $\{\omega^1, \dots, \omega^N\}$ according to the probability distribution $\mathcal{P}$. Then, the expected value function $\mathbb{E}_{\mathcal{P}}[Q(x, \xi(\omega))]$ is approximated by the sample average function $\sum_{n=1}^{N} Q(x, \xi(\omega^n))/N$. This gives us the Sample Average Approximation problem corresponding to (17):

$$z_N = \min_{x \in X} c^T x + \frac{1}{N} \sum_{n=1}^{N} Q(x, \xi(\omega^n)). \tag{19}$$

The SAA method makes use of M independent sample of size N. Solving the corresponding SAA problems provides us with objective values $z_N^1, z_N^2, \dots, z_N^M$ and candidate solutions $\hat{x}_N^1, \hat{x}_N^2, \dots, \hat{x}_N^M$. We denote the average objective value of the M SAA problems by $\bar{z}_N$, so

$$\bar{z}_N = \frac{1}{M} \sum_{m=1}^{M} z_N^m. \tag{20}$$

Mak et al. (1999) and Norkin et al. (1999) prove that $\mathbb{E}[\bar{z}_N] \leq z^*$. This means that $\bar{z}_N$ is a statistical estimate for a lower bound on the optimal value of (17). An estimate for the variance of $\bar{z}_N$ is

$$\hat{\sigma}^2_{\bar{z}_N} = \frac{1}{(M-1)M} \sum_{m=1}^{M} (z_N^m - \bar{z}_N)^2. \tag{21}$$

For any feasible point $\hat{x} \in X$, an upper bound for z^* can be given by objective value $c^T \hat{x} + \mathbb{E}[Q(x, \xi(\omega))]$. Using a independent sample of size N', the objective value can be estimated by

$$\hat{z}_{N'} = c^T \hat{x} + \frac{1}{N'} \sum_{n=1}^{N'} Q(\hat{x}, \xi(\omega^n)). \tag{22}$$

Commonly, N' is chosen to be large. It is possible to do this for N', as we do not have to solve the SAA problem with this large sample. Therefore it does not take a lot of time to compute this. If we would make N larger, it would have an effect on the computation time. It is therefore common to take $N' > N$ at least. This sample of size N' is independent of the sample used to generate $\hat{x}$. Therefore, $\hat{z}_{N'}(\hat{x})$ is an unbiased estimator of $c^T x + \mathbb{E}_{\mathcal{P}}[Q(x, \xi(\omega))]$. Then, for any feasible solution $\hat{x}$, it holds that $\mathbb{E}[\hat{z}_{N'}(\hat{x})] \geq z^*$. An estimate for the variance of $\hat{z}_{N'}(\hat{x})$ is

$$\hat{\sigma}^2_{\hat{z}_{N'}(\hat{x})} = \frac{1}{(N'-1)N'} \sum_{n=1}^{N'} (c^T \hat{x} + Q(\hat{x}, \xi(\omega^n)) - \hat{z}_{N'}(\hat{x}))^2. \tag{23}$$

In the above procedures, we have obtained M different candidate solutions. We take $\hat{x}^*$ as the candidate solution with the best objective value. The quality of the solution $\hat{x}^*$ can be evaluated by computing the optimality gap estimate

$$\hat{\Delta} = \hat{z}_{N'}(\hat{x}^*) - \bar{z}_N. \tag{24}$$

Here $\hat{z}_{N'}(\hat{x}^*)$ is the objective value of $\hat{x}^*$ using an independent sample of size N'. This is done in order to obtain an unbiased estimate. According to Kleywegt et al. (2001), the variance of this optimality gap estimator can be estimated by

$$\hat{\sigma}^2_{\hat{\Delta}} = \hat{\sigma}^2_{\hat{z}_{N'}(\hat{x}^*)} + \hat{\sigma}^2_{\bar{z}_N}. \tag{25}$$

4.2 Solving the SAA problem using a Branch-and-Cut Algorithm

A branch-and-cut procedure can be used to solve mixed integer programming problems. It uses the simplex method to solve LP relaxations. When an optimal solution for the relaxation is found, and the integrality constraints have not been satisfied, a cutting plane algorithm is used tighten the feasible region while not removing any feasible integer points. The new LP problems are solved and the process is repeated. During this process, feasibility and optimality cuts can be added. These cuts help to find a good solution. In our branch-and-cut algorithm, the sub-tour elimination constraints are added as feasibility cuts for the TSPRT and VRPRT. For all problems, optimality cuts are added. The procedure of generating these optimality cuts will be discussed in Section 4.3.

To be able to use the branch-and-cut algorithm to solve the SAA problem (19), we rewrite it as

$$\min_{x \in X} c^T x + \frac{1}{N} Q_N(x), \tag{26}$$

where X is the set of feasible first-stage solutions and

$$Q_N(x) = \sum_{n=1}^{N} Q(x, \xi(\omega^n)). \tag{27}$$

Here $Q(x, \xi(\omega))$ is given by Equation (18). Verweij et al. (2003) state that it is well-known that $Q(x, \xi(\omega))$ in Equation (18), and therefore $Q_N(x)$, are piecewise linear and convex in x. This means that $Q_N(x) = \max\{a_i^T x - a_{i0} \,|\, i \in \{1, 2, \ldots, L\}\}$ for all $x \in X$, for some $\{(a_i, a_{i0}), i = 1, 2, \ldots L\}$, and positive integer L. This means that we can rewrite problem (26) as a mixed-integer linear program with L constraints as follows:

$$\begin{aligned} \min \quad & c^T x + \theta/N && \qquad (28) \\ \text{subject to} \quad & \theta \geq a_i^T x - a_{i0} \qquad \text{for } i \in \{1, 2, \ldots, L\}, && \qquad (29) \\ & x \in X. && \qquad (30) \end{aligned}$$

For a given first-stage solution x, the optimal value of θ equals $Q_N(x)$. The coefficients of the optimality cuts (29), $\{(a_i, a_{i0})\}$, are given by the extreme subgradients of $Q_N(x)$. These are themselves given by the extreme point optimal dual solutions of the second-stage problems (18). The number of constraints (29), L, is usually very large. Instead of including all of these constraints, only a subset will be generated within the branch-and-cut framework using the well-known integer L-shaped decomposition method (Slyke and Wets, 1969).

4.3 Optimality cut generation

As mentioned before, within the branch-and-cut framework optimality cuts are iteratively generated at each integer solution. In this subsection we describe how we generate these for each of the investigated problems.

4.3.1 SPRT and TSPRT

For the SPRT and the TSPRT the second-stage recourse problem for sample scenario ω^n, with $n \in \{1, 2, \dots, N\}$, is given by

$$Q_n(x) = \max(\xi(\omega^n)^T x - \kappa, 0). \tag{31}$$

It holds that

$$\sum_{n=1}^{N} Q_n(x) = \max\left\{ \sum_{n\in S} (\xi(\omega^n)^T x - \kappa) \;\middle|\; S \subseteq \{1, \dots, N\} \right\}. \tag{32}$$

Therefore, the sample average approximation problem can be formulated as

$$\begin{aligned}
\min \quad & c^T x + \theta/N && (33)\\
\text{subject to} \quad & Ax \le b, && (34)\\
& \theta \ge \sum_{n\in S} (\xi(\omega^n)^T x - \kappa) \qquad \text{for } S \subseteq \{1, 2, \dots, N\}, && (35)\\
& x \in \{0, 1\}^{|D|}, && (36)
\end{aligned}$$

where D is the set of arcs A when solving the SPRT and the set of edges E when solving the TSPRT.

The inequalities (35) are the optimality cuts, and the system $Ax \le b$ represents the constraints in the original problem. For the SPRT, these are only constraints (2) and for the TSPRT these are constraints (6) and (7). However, constraints (7) are added as feasibility cuts. Note that (35) represents a very large number of constraints. However, in our branch-and-cut algorithm we only add the maximally violated optimality cut, which corresponds to the subset $S(x) = \{n \in \{1, 2, \dots, N\} \big| \xi(\omega^n)^T x > \kappa\}$ for a feasible x and $\omega^n \in \Omega$.

4.3.2 VRPRT

For the VRPRT the second-stage recourse problem for sample scenario ω^n, with $n \in \{1, 2, \dots, N\}$ is given by

$$Q_n(x) = \sum_{r=1}^{R} \max\{\xi(\omega^n)^T x - \kappa_r, 0\}. \tag{37}$$

It holds that

$$\sum_{n=1}^{N} Q_n(x) = \max\left\{ \sum_{(n,r)\in S} \left(\xi(\omega^n)^T x_r - \kappa_r\right) \;\middle|\; S \subseteq \Psi \right\}, \tag{38}$$

where $\Psi = \{(n,r)\,|\,n \in \{1,2,\ldots,N\}, r \in \{1,\ldots,R\}\}$. Therefore the sample average approximation problem can be formulated as

$$\min \quad c^T x + \theta/N \tag{39}$$
$$\text{subject to} \quad Ax \le b, \tag{40}$$
$$\theta \ge \sum_{(n,r)\in S} \left(\xi(\omega^n)^T x_r - \kappa_r\right) \qquad \text{for } S \subseteq \Psi, \tag{41}$$
$$x \in \{0,1\}^{|D|}, \tag{42}$$

Here the inequalities (41) represent the optimality cuts, and the system $Ax \le b$ represents the constraint in the original problem. For the VRPRT these are constraints (11) – (14). Again, constraints (14) are iteratively added as feasibility cuts. Note that again (41) represents a very large number of constraints. In the branch-and-cut algorithm we only add the maximally violated optimality cut, which corresponds to the subset $S(x) = \{(n,r) \in \Psi |\left(\xi(\omega^n)^T x_r - \kappa_r\right) > 0\}$ for a feasible x and $\omega^n \in \Omega$.

5 Computational Experiments

This section describes the methods used to generate the instances used to solve the SPRT, TSPRT and VRPRT problems. We implemented the SAA method and Branch-and-Cut algorithm using the commercial MIP solver CPLEX version 12.6.3 with Java version 8. We made use of the LazyConstraintCallback function within CPLEX. This way we were able to add feasibility and optimality cuts at integer solutions.

5.1 Instances

Since the precise instances of Verweij et al. (2003) are not available, we have used their description to generate the instances. We now describe this process of generating a problem instance using the well-known TSPlib instances (Reinelt, 1995). Each TSPlib instance has a set coordinates for each of the cities in it and states in which way the inter-city distance can be determined.

5.1.1 Shortest Path with Random Travel Times

The procedure used by Verweij et al. (2003) to generate a SPRT instance from a TSPlib instance can be summarized as follows.

To generate the directed graph $G = (V, A)$, we associate city i from the TSPlib instance with vertex $v_i \in V$. We iterate over the vertices in G in ascending order, from v_1 to v_n. In iteration i, we connect the δ closest vertices that have not yet been connected with vertex v_i. Connecting vertices v_i and v_j is done by adding arcs (v_i, v_j) and (v_j, v_i) to A. For the cost c_{ij} of arc (v_i, v_j) we take the corresponding distance d_{ij}. Choosing the source node v_s and sink node v_t is done by first finding all pairs (v_i, v_j) with v_i, $v_j \in V$ that maximize the minimum number of arcs traversed over the optimal $v_i - v_j$ path. We then choose one pair uniformly at random to be the source and sink nodes. We denote the vector of random travel times by $\xi(\omega)$ according to scenario $\omega \in \Omega$. For arc $a \in A$ we use the following probability distribution:

$$\mathcal{P}\{\xi_a(\omega) = Fc_a\} = \begin{cases} p, & \text{if } c_a \le \bar{c} \\ 0, & \text{if } c_a \ge \bar{c} \end{cases} \qquad \text{and } \mathcal{P}\{\xi_a(\omega) = c_a\} = \begin{cases} 1-p, & \text{if } c_a \le \bar{c} \\ 1, & \text{if } c_a \ge \bar{c} \end{cases}$$

Here $\bar{c}$ denotes the median arc length and $F > 1$ and $0 < p < 1$ are given parameters. We determine deadline κ by computing the shortest path in G with respect to the arc costs.

5.1.2 Traveling Salesman Problem with Random Travel Times

The procedure used by Verweij et al. (2003) to generate a TSPRT instance from a TSPlib instance is similar to the one of the SPRT. The difference is in how we connect two vertices as we have an undirected graph $G = (V, E)$ in the TSPRT. For the TSPRT, connecting the vertices v and w is done by adding edge (v, w) to E where we take the edge cost c_{vw} to be the corresponding distance d_{vw}. We denote the vector of random travel times by $\xi(\omega)$ according to scenario $\omega \in \Omega$. For edge $e \in E$ we use the following probability distribution:

$$\mathcal{P}\{\xi_e(\omega) = Fc_e\} = \begin{cases} p, & \text{if } c_e \leq \bar{c} \\ 0, & \text{if } c_e \geq \bar{c} \end{cases} \qquad \text{and } \mathcal{P}\{\xi_e(\omega) = c_e\} = \begin{cases} 1-p, & \text{if } c_e \leq \bar{c} \\ 1, & \text{if } c_e \geq \bar{c} \end{cases}$$

Here $\bar{c}$ denotes the median edge cost, and $F > 1$ and $0 < p < 1$ are given parameters. The deadline κ is taken to be the cost of the shortest Hamiltonian cycle in G.

The dimensions of the graphs generated following this method can be seen in Table 1. In this table, the dimensions of the graphs generated by Verweij et al. (2003) can be seen as well. When comparing the results, we find quite some differences. This could lead to differences in the final results.

Table 1: Dimensions of the graphs generated from the TSPlib instances

		Our Result		Verweij et al.				Our Result		Verweij et al.	
Instance	$\lvert V\rvert$	$\lvert A\rvert$	$\lvert E\rvert$	$\lvert A\rvert$	$\lvert E\rvert$	Instance	$\lvert V\rvert$	$\lvert A\rvert$	$\lvert E\rvert$	$\lvert A\rvert$	$\lvert E\rvert$
burma14	14	182	91	172	86	rat99	99	1980	990	1872	936
ulysses16	16	228	114	212	106	kroA100	100	2000	1000	1892	946
ulysses22	22	404	202	332	166	kroB100	100	2000	1000	1892	946
eil51	51	1020	510	912	456	kroC100	100	2000	1000	1892	946
berlin52	52	1040	520	932	466	kroD100	100	2000	1000	1892	946
st70	70	1400	700	1292	646	kroE100	100	2000	1000	1892	946
eil76	76	1520	760	1412	706	rd100	100	2000	1000	1892	946
pr76	76	1520	760	1412	706	eil101	101	2020	1010	1912	956
gr96	96	1920	960	1812	906						

This table contains the dimensions of the graphs generated from TSPlib instances for both our results and the results of Verweij et al. (2003). As can be seen, the dimensions differ quite a lot.

5.1.3 Vehicle Routing Problem with Random Travel Times

The procedure used to generate a VRPRT instance from a TSPlib instance is very similar to procedure for the SPRT instances. The directed graph $G = (V, A)$ is generated exactly the same. For the VRPRT we have to determine which node will function as a depot instead of finding a pair of sink and source nodes. Determining the depot is done by computing the distance from a node $v_i \in V$ to all other nodes $v_j \in V$, $j \neq i$. We chose the node v_i with the smallest total distance to be our depot. This way we ensure the depot is located relatively centered in the instance.

The deadline κ_r corresponding to vehicle $r \in \{1, \ldots, R\}$ is chosen to be the deadline for the TSPRT devided by the amount of vehicles r, so $\kappa_r = \kappa/R$. In our case, we only look at the problem where 2 vehicles are available, so $\kappa_r = \kappa/2$.

5.1.4 Parameter choice

The parameters are chosen to be the same as the parameters used by Verweij et al. (2003).

For the probability distribution, the following parameters are used: $F = 10$ for the SPRT and $F = 20$ for the TSPRT and VRPRT, and $p = 0.1$ for all problems. For the generation of the graphs, δ is set to 10. We generate $M = 10$ samples when using the SAA method. A sample size of $N' = 10^5$ is used when generating a sample of scenarios to check the quality of our solution.

6 Results

In Table 2 the results of the computational experiments can be found. For a lot of the TSPRT and VRPRT instances, the results are not available. This is because of unexpected out-of-memory errors and time restrictions.

When comparing these values with the values found by Verweij et al. (2003), we see some differences. These differences are especially visible for instances where the nodes are given by geographical coordinates in the TSPlib. When comparing the value for the TSPRT for the `burma14` instance found by Verweij et al. (2003) and the optimal value for a Hamiltonian cycle in this instance given in the TSPLib documentation, we find that the value found by Verweij et al. (2003) is around half of this optimal value. As the TSPRT should give at best the optimal cycle, we think something went wrong with the implementation of this distance by Verweij et al. (2003).

When comparing the instances that are generated according to euclidean distance, we see that the results for the SPRT are close. For the TSPRT, however, the differences are larger. For every instance, the values found by Verweij et al. (2003) for the TSPRT are much lower. The gap estimate $\hat{\Delta}$ and its variance differ as well. In our results, almost all gap estimates are negative, which is not what is expected. Besides that, they are much further away from 0 than found by Verweij et al. (2003). The variance of the gap estimate is also much larger than found by Verweij et al. (2003).

When comparing the values of the VRPRT and the TSPRT, we see that the costs have increased a little. Taking the gap estimate of the TSPRT into account, the differences are not that big. This small increase in routing costs could however be compensated by the increased customer service. As time did not allow us to investigate other deadlines or different amount of vehicles, we do not know what their effect would be on the final results. Further research has to be done in order to be able to make good conclusions on this.

Table 2: Summary of solution values for $N = 1000$

	SPRT			TSPRT			VRPRT	
Instance	$\hat{z}_{N'}(\hat{x}^*)$	$\hat{\Delta}$	$\sigma_{\hat{\Delta}}$	$\hat{z}_{N'}(\hat{x}^*)$	$\hat{\Delta}$	$\sigma_{\hat{\Delta}}$	$\bar{z}_N$	$\sigma_{\bar{z}_N}$
burma14†	482.0	-148	36.20	7214.9	37.15	88.25	8112.2	65.13
ulysses16†	1501.3	-367.8	155.00	12450.2	-145.5	641.02	13490.7	265.51
ulysses22†	1013.5	-328.8	155.00	10270.5	-2225.50	92.32	13440.8	103.43
eil51	69.0	1.1	0.23	1194.7	12.82	10.63	-	-
berlin52	1229.0	219.4	28.19	12885.8	-1684.02	91.30	-	-
st70	113.0	0.0	0.00	-	-	-	-	-
eil76	71.0	-11.2	1.87	-	-	-	-	-
pr76	15538.0	0.0	0.00	-	-	-	-	-
gr96†	8515.0	-867.9	198.58	-	-	-	-	-
rat99	211.0	0.0	0.00	3408.7	11.72	27.16	-	-
kroA100	4132.2	-50.9	28.25	-	-	-	-	-
kroB100	4112.0	0.0	0.00	-	-	-	-	-
kroC100	3784.0	-106.4	70.93	-	-	-	-	-
kroD100	3970.0	-95.5	53.84	-	-	-	-	-
rd100	1121.2	-157.3	26.87	-	-	-	-	-
eil101	61.4	-6.8	1.29	-	-	-	-	-

This table contains the solution values for all three problems. The instance `kroE100` has been taken out, as no solution was found without going out of memory for all problems. For all instances with more than 25 nodes, a time limit of 2 hours has been set on the solution time for the TSPRT and VRPRT problems. Still, when solving most of the instances out-of-memory errors were received. When decreasing the time limit even further, no solution was found.

† Instances with nodes given by geographical coordinates

7 Conclusion

We have applied the Sample Average Approximation method to several stochastic routing problems. Two of these problems, the shortest path problem with random travel times and the traveling salesman problem with random travel times, have been previously studied by Verweij et al. (2003). We have tried to replicate the results in their study. Additionally, as an extension on the TSPRT where multiple vehicles are available, we have investigated the vehicle routing problem with random travel times.

When comparing our results with those of Verweij et al. (2003), we have found that they are quite similar. Extending the TSPRT to a VRPRT does not give much more routing costs, and therefore it might be useful when in need of quick delivery and in general to increase customer satisfaction. Computation time is very long for the VRPRT however. Therefore the method has not been tested on larger instances. As a consequence it has also not been tested on instances with euclidean distance.

For further research, a two-index model for the VRPRT may be interesting as this should decrease computational time. Furthermore, a formulation with edges instead of arcs for the VPRRT may be beneficial for the computational time. Other interesting research areas include the effect of changing the deadline on the final result and the effect of having more than 2 vehicles on the final result. Time restrictions and difficulties with the implementation prevented us from investigating these areas.

References

Andreatta, G. (1987). Shortest path models in stochastic networks. *Stochastics in combinatorial optimization*, pages 178–186.

Andreatta, G. and Romeo, L. (1988). Stochastic shortest paths with recourse. *Networks*, 18(3):193–204.

Bertazzi, L. and Maggioni, F. (2015). Solution approaches for the stochastic capacitated traveling salesmen location problem with recourse. *Journal of Optimization Theory and Applications*, 166(1):321–342.

Bertsimas, D. J. (1992). A vehicle routing problem with stochastic demand. *Operations Research*, 40(3):574–585.

Borcinova, Z. (2017). Two models of the capacitated vehicle routing problem. *Croatian Operational Research Review*, 8(2):463–469.

Gendreau, M., Laporte, G., and Séguin, R. (1996). Stochastic vehicle routing. *European Journal of Operational Research*, 88(1):3–12.

Geyer, C. J. and Thompson, E. A. (1992). Constrained monte carlo maximum likelihood for dependent data. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 657–699.

Higle, J. L. and Sen, S. (1991). Stochastic decomposition: An algorithm for two-stage linear programs with recourse. *Mathematics of operations research*, 16(3):650–669.

Infanger, G. (1992). Planning under uncertainty solving large-scale stochastic linear programs. Technical report, Stanford Univ., CA (United States). Systems Optimization Lab.

Kleywegt, A. J., Shapiro, A., and Homem-de-Mello, T. (2001). The Sample Average Approximation Method for Stochastic Discrete Optimization. *SIAM Journal on Optimization*, 12:479–502.

Laporte, G., Louveaux, F., and Mercure, H. (1992). The Vehicle Routing Problem with Stochastic Travel Times. *Tansportation Science*, 26(3):161–170.

Laporte, G. and Louveaux, F. V. (1993). The integer l-shaped method for stochastic integer programs with complete recourse. *Operations research letters*, 13(3):133–142.

Laporte, G., Louveaux, F. V., and Hamme, L. V. (1994a). Exact solution of a stochastic location problem by an integer l-shaped algorithm. *Transportation Science*, 28(2):95–103.

Laporte, G., Louveaux, F. V., and Mercure, H. (1994b). A priori optimization of the probabilistic traveling salesman problem. *Operations research*, 42(3):543–549.

Leipälä, T. (1978). On the solutions of stochastic traveling salesman problems. *European Journal of Operational Research*, 2(4):291–297.

Mak, W.-K., Morton, D. P., and Wood, R. K. (1999). Monte Carlo bounding techniques for determining solution quality in stochastic programs. *Operations Research Letters*, 24:47–56.

Matai, R., Singh, S., and Mittal, M. L. (2010). *Traveling Salesman Problem: an Overview of Applications, Formulations, and Solution Approaches*. InTech.

Norkin, V. I., Pflug, G. C., and Ruszczynski, A. (1999). A branch and bound method for stochastic global optimization. *Mathematical Programming*, 83:425–450.

Plambeck, E. L., Fu, B.-R., Robinson, S. M., and Suri, R. (1996). Sample-path optimization of convex stochastic performance functions. *Mathematical Programming*, 75(2):137–176.

Polychronopoulos, G. H. and Tsitsiklis, J. N. (1996). Stochastic shortest path problems with recourse. *Networks: An International Journal*, 27(2):133–143.

Reinelt, G. (1995). http://softlib.rice.edu/pub/tsplib/tsp/.

Rubinstein, R. Y. and Shapiro, A. (1990). Optimization of static simulation models by the score function method. *Mathematics and Computers in Simulation*, 32(4):373–392.

Shapiro, A. (1996). Simulation-based optimization—convergence analysis and statistical inference. *Stochastic Models*, 12(3):425–454.

Shapiro, A. and Homem-de Mello, T. (1998). A simulation-based approach to two-stage stochastic programming with recourse. *Mathematical Programming*, 81(3):301–325.

Slyke, R. V. and Wets, R.-B. (1969). L-shaped Linear Programs with Applications to Optimal Control and Stochastic Programming. *SIAM Journal on Applied Mathematics*, 17:638–663.

Spaccamela, A. M., Rinnooy Kan, A., and Stougie, L. (1984). Hierarchical vehicle routing problems. *Networks*, 14(4):571–586.

Taş, D., Dellaert, N., Van Woensel, T., and 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.

Verweij, B., Ahmed, S., Kleywegt, A. J., Nemhauser, G., and Shapiro, A. (2003). The Sample Average Approxiation Method Applied to Stochastic Routing Problems: A Computational Study. *Computational Optimization and Applications*, 24:289–333.

Wallace, S. (1988). A two-stage stochastic facility-location problem with time-dependent supply. *Numerical Techniques for Stochastic Optimization,(Yu. Ermoliev and R. JB. Wets (eds.)) pp*, pages 489–513.

Wallace, S. W. (1986). Solving stochastic programs with network recourse. *Networks*, 16(3):295–317.

Wallace, S. W. (1987). Investing in arcs in a network to maximize the expected max flow. *Networks*, 17(1):87–103.