A Multi-stage Monte Carlo Sampling Based Stochastic Programming Model for the Dynamic Vehicle Allocation Problem

Wei Fan and Randy Machemehl

Department of Civil Engineering, University of Texas at Austin, ECJ 6.9, Austin, TX 78712, USA

Abstract: Optimization under uncertainty has seen many applications in the industrial world. The objective of this paper is to study the stochastic dynamic vehicle allocation problem (SDVAP), which is faced by many trucking companies, container companies, rental car agencies and railroads. To maximize profits and to manage fleets of vehicles in both time and space, this paper has formulated a multistage stochastic programming based model for SDVAP. A Monte Carlo Sampling Based Algorithm has been proposed to solve SDVAP. A probabilistic statement regarding the quality of the solution from the Monte Carlo sampling method is also identified by introducing a lower bound and an upper bound of the obtained optimal solution. A five-stage experimental network was introduced for demonstration of this algorithm. The computational results indicated a solution of high quality when Monte Carlo sampling based algorithm is used for solving SDVAP, strongly suggesting that these algorithms can be used for real world applications for decision making under uncertainty.

Keywords**:** Stochastic programming; Monte Carlo Sampling Based Method (MCSBM); Simulation; Dynamic vehicle allocation; Multistage

1. Introduction

Freight transportation usually involves a significant number of empty vehicle movements. To maximize profits in a competitive industry, trucking companies, container companies, rental car agencies and railroads, must manage in both time and space fleets of vehicles. Understanding and better controlling this phenomenon is very important for the managers of freight-companies. From this need arises the vehicle allocation problem, which will be investigated in this paper. As customers, generally shippers, request the carrier to pick up cargo at specific locations and deliver these items to given destinations on a specific day, the demand for that day is determined. Note that if any demand is considered unprofitable or the service capacity cannot accommodate it, this request will be declined by the carrier. As in the real-world situation, some assumptions are made in this paper. Each demand can be served by a single vehicle and the vehicle is used for this demand on that specific day. No vehicles can be shared by different demands. Time is categorized into intervals, typically day-to-day, and on each day, each vehicle can either be assigned to pick up a demand or moved empty to another region to serve a requested or expected demand, or stay in the same region until the next day. Obviously, decisions made on one day will have a direct impact on future vehicle supplies of regions. The problem of how to manage such a vehicle fleet forms the "Stochastic Dynamic Vehicle Allocation Problem" (SDVAP).

As noted by many researchers, the major difficulty of SDVAP lies in uncertain demand and the increasing level of uncertainty further into the future. This stochastic property obviously requires the carrier to be able to estimate future demands and to make decisions that anticipate their impacts on future periods.

Several researchers have formulated SDVAP problems and have proposed different methodologies to solve this problem considering its uncertainty and finite and/or infinite horizon nature. However, static deterministic, static stochastic, or dynamic deterministic formulations are the main focus of previous research. Dejax and Crainic[8] presented a taxonomy of empty vehicle flow problems and models and reviewed the existing literature. Several research perspectives are identified and the advantages of integrated approaches for simultaneous management of empty and loaded freight vehicle movements are also discussed. Jordan and Turnquist[12] incorporated uncertainty of demands and vehicle supplies into the model for the first time and proposed a network optimization model for the allocation of empty freight cars in the railroad carrier case. The model was solved by a Frank-Wolfe algorithm and computation was shown to be very efficient. Frantzeskakis and Powell[10] proposed a heuristic algorithm and contrasted it to various deterministic approximations. A rolling horizon procedure was employed to simulate the operation of a railroad or a truckload carrier. Computational results indicated the superiority of the new algorithm over other approaches tested.

These research efforts provided several insights for SDVAP. However, few efforts are oriented to simultaneous consideration of the dynamic and stochastic properties of the SDVAP problem. The main purpose of this paper is to formulate and solve an instance of the SDVAP under a finite planning horizon. Particular attention is given to solving this multistage stochastic model using a Monte Carlo sampling-based algorithm (MCSBA). Numerical results and vehicle allocation recommendations are presented and analyzed. This paper is organized as follows. Section 2 presents the problem statement and the assumptions that are made in this paper. Section 3 presents the model formulation and candidate solution methodologies for stochastic programming problems. Section 4 discusses the details of the Monte Carlo Sampling Based Method (MCSBM), along with a probabilistic statement regarding the quality of solution through the presentation of its lower bound and upper bound. Section 5 presents the computational results of the experimental network. Finally, section 6 concludes the paper.

2. Problem Statement

In this section, a stochastic programming formulation is presented for the SDVAP with a planning horizon of N days. As Powell (1990) noted, the SDVAP can be restated as follows: at the present time period and with the available vehicle fleet allocated among different cities, a truck carrier must decide which loads to accept or refuse and how many vehicles to relocate or hold to achieve a more favorable future vehicle allocation, and eventually to maximize the total expected profits over a planning horizon of N days under the condition of independent random future demands with a known distribution.

Figure 1 gives a graphical view for SDVAP through network flow representation. (Further notations can be seen later in Section 3.) To facilitate the formulation of the SDVAP, several assumptions are made as listed below:

1) The travel times between all cities are uniformly equal to one day, either for a loaded or an empty movement;

2) At the beginning of each day, the truck carrier knows all the demands of that day. Therefore, the first day demand is deterministic. However, he doesn't know any of the demands that will be requested on the following days. And certainly, the carrier can make the decision by solving the model once per day;

3) Any demands that cannot be satisfied for a specific day could be lost;

4) All vehicles are available for the first time on the first day;

5) The demands of all pairs of cities conform to a Poisson distribution and the parameters (mean values) are known but not necessarily equal.

In order to formulate the SDVAP as a cost minimization problem, revenues from carrying a load are treated as negative costs while costs of empty movement are regarded as positive revenues. Additionally, since each vehicle carries only one load, the two terms, “flow of vehicles” and “number of vehicles” are essentially the same. In addition, to present the formulation of the SDVAP, some notations are introduced and a stochastic programming model is developed and described in the following model formulation.

3. Model Formulation

Indices/Sets:

$i, j \in \mathrm{R}$ Regional origins and/or destinations

$\mathrm{t} \in \mathrm{T}$ Time periods

Parameters/Data:

r_{ij} = net revenue for pulling a load from i to j.

c_{ij} = cost of moving empty from i to j.

L_{ij1} = number of loads known at time t = 1 to be available moving from i to j at the first time period.

S_{i1} = supply of trucks at region i at period 1.

Random Variables:

$\tilde{d}_{ijt}$ = random demands denoting the number of loads that will be requested to go from i to j during period t, t = 2,…,N

Decision Variables:

x_{ijt}^{w} = number of trucks moving loaded from region i to region j, in the beginning of period t in scenario w, t = 1,2,…, N

y_{ijt}^{w} = number of trucks moving empty from region i to region j, in the beginning of period t in scenario w, t = 1,2,…, N

S_{it}^{w} = supply of trucks at region i during period t in scenario w, t = 2,…, N.

$$= \sum_{k \in \mathrm{R}} [x_{ki(t-1)}^{w} + y_{ki(t-1)}^{w}]$$

Objective Functions:

The optimization model for each time period t =2, 3, …, N-1 is:

$$h(x_{ijt}, y_{ijt}, \tilde{d}_{ijt}) = \min_{x_{ijt}, y_{ijt}} (-\sum_i \sum_j r_{ij} x_{ijt} + \sum_i \sum_j c_{ij} y_{ijt} + Eh(x_{ij(t+1)}, y_{ij(t+1)}, \tilde{d}_{ijt}) \qquad (1)$$

$$\begin{aligned} \text{subject to:} \quad & x_{ijt} \le \tilde{d}_{ijt} && i \in R, j \in R && (1a) \\ & x_{ijt} + y_{ijt} = S_{it} && i \in R, j \in R && (1b) \\ & \sum_i (x_{ijt} + y_{ijt}) = S_{j(t+1)} && i \in R, j \in R && (1c) \end{aligned}$$

The optimization model for the last period N is:

$$h(x_{ijt}, y_{ijt}, \tilde{d}_{ijt}) = \min_{x_{ijt}, y_{ijt}} (-\sum_i \sum_j r_{ij} x_{ijt} + \sum_i \sum_j c_{ij} y_{ijt}) \qquad (2)$$

$$\begin{aligned} \text{subject to:} \quad & x_{ijt} \le \tilde{d}_{ijt} && i \in R, j \in R && (2a) \\ & x_{ijt} + y_{ijt} = S_{it} && i \in R, j \in R && (2b) \end{aligned}$$

An N-stage stochastic programming procedure was formulated as follows to maximize total profits over the N time period horizon:

$$\min_{x_{ij1}, y_{ij1}} (-\sum_i \sum_j r_{ij} x_{ij1} + \sum_i \sum_j c_{ij} y_{ij1} + Eh(x_{ij1}, y_{ij1}, \tilde{d}_{ijt}) \qquad (3)$$

$$\begin{aligned} \text{subject to:} \quad & x_{ij1} \le L_{ij1} && i \in R, j \in R && (3a) \\ & x_{ij1} + y_{ij1} = S_{i1} && i \in R, j \in R && (3b) \\ & \sum_i (x_{ij1} + y_{ij1}) = S_{j2} && i \in R, j \in R && (3c) \end{aligned}$$

As illustrated in Figure 1, it should be noted that in the above model formulation, the first set of constraints implies that all loaded movements must be no more than the requested demand for those movements. The second set of constraints guarantee that the total number of loaded and empty vehicles that move out of region i at time period t cannot exceed the vehicle supplies available at that time period. The third set of constraints in (1) and (3) presents the flow conservation properties, i.e., the number of vehicles available at region j at time period t+1 is equal to the number of loaded or empty movements to region j at time period t.

Problems involving uncertainty in the objective function and/or constraints fall in the domain of Stochastic Programming. Furthermore, it can be seen that the SDVAP is a linear programming problem with uncertain constraint coefficients in the right-hand-side. As described by Powell (1990), the great difficulty arising from these problems is the required truncation of their infinite planning horizon to a certain number (N) of finite time periods and this might cause deviations from the infinite planning horizon optimal solution. As a result of this transformation, the SDVAP can be treated as a Multiple-stage (here N-stage) Stochastic Linear Programming (SLP) problem with recourse.

Several approaches have been proposed to solve Multiple-stage Stochastic Linear Programming during the past decades. Generally speaking, these approaches for solving multi-stage SLP problems can be classified into two main groups:

1) “Exact” Optimization Methods

As the main focus on solving Multiple-stage Stochastic Linear Programming in the early stages, exact optimization methods have been extensively investigated by many researchers. Generally speaking, these methods can be further divided into three categories, which consist of the Simplex method, Interior point methods, and the Decomposition method. Decomposition is the most popular approach to solve SLP problems, which includes Dantzig-Wolfe, Benders, L-shaped and Cutting plane algorithms. These methods are very efficient at solving SLP problems including multi-stage SLP problems, especially when the problem size is manageable.

However, if the number of scenarios is very large or the involved random variables are a continuous random vector from a certain probability distribution, then performing an “Exact” evaluation of the objective function may be difficult or even impossible unless the sub-problem has some very special structure like a simple recourse. In this case, some researchers resort to approximation methods to get some ideas of how good the solution is.

2) Approximation techniques

Existing approximation techniques to solve SLP problems can be categorized in two classes. The first, “Approximation and Bounding Techniques”, use Jensen and Edmundson-Madansky (EM) bounds, and Sequential approximation methods to provide bounds for network recourse problems. The second kind of Approximation technique is known as “Monte Carlo Sampling-Based Algorithms”. Monte Carlo sampling is an approximation technique and provides qualitative upper and lower bounds for the optimal objective function value. The best that the approach can do is to compute a feasible solution that yields an objective function value within an interval that contains the optimal objective value with a certain percentage confidence. The Monte Carlo sampling method is capable of solving difficult problems.

As approximation techniques, both methods are efficient in solving SLP problems. Due to its great flexibility and easily-implemented properties, the Monte Carlo Sampling Based Algorithm is employed as the solution methodology for SDVAP in this paper. In the following sections, the characteristics underlying the Monte Carlo sampling method are described. Quantitative upper and lower bounds are also presented for the optimal solution from the Monte Carlo Based Sampling Method.

4. Monte Carlo sampling method

4.1. Introduction

As mentioned before, since the loading demand that might be requested conforms to a Poisson distribution and the scenario is too large to be seen, it is very difficult to solve the SDVAP using the “Exact” Optimization Methods. In this paper, the Monte Carlo sampling method is employed to solve SDVAP.

When handling stochastic programming problems, it is always good to know the difference between the quality of solution obtained from a deterministic method and that of a sampling-based method. For the deterministic method, $\hat{x}$ is a solution that yields an objective function value within 1% of the optimal objective function value. While for the Sampling-Based Method, $\hat{x}$ is a feasible solution that provides 95% confidence that $\hat{x}$ yields an objective function value within 1% of the optimal objective function value. As mentioned by Morton (2002), the loss of ability to make more precise statements regarding the solution quality, is replaced by the capability to solve more difficult problems. Obviously, however, the “quality statement” given by a Deterministic Method would be preferred if it is available.

Essentially speaking, as an innovative approach to solving stochastic optimization problems, the basic idea of the Monte Carlo sampling method is to generate a random sample and approximate the expected value function by the corresponding sample average function. Figure 2 illustrates a scenario tree for multi-stage stochastic programming models. In this figure, the nodes in the tree represent states at a particular period. Decisions are made at the nodes and the arcs represent realizations of the uncertain variables. Decisions to be made further down the scenario tree depend on the decisions already made through parent nodes and the uncertain properties of children nodes. Note that the generation of scenarios is based on simulation and the decision makers can specify the probability distribution function so that the statistical properties are preserved. A single simulation scenario consists of realizations of the uncertain variables in each simulation (time) period. In practice, only the first-stage solution at the top node will be used for decision making. The decisions made at stage two or after that are only made in order to find the right incentives for the first-stage decisions (Fleten, 2002).

In all, the Monte Carlo Sampling Based Method for solving Multi-stage Stochastic Programming can be restated as follows. At the beginning of the first period, the decisions are made based on the current information (and simulation of the stochastic future), and then, at the end of the first period, the consequences of this decision are seen. Given this consequence and new information for the next period, a new decision is made at the beginning of the second period. Based on the outcomes from the second period and given new information for the third period, the decision is made again. The whole process continues, and in principle, indefinitely.

Note that for each scenario tree with generated random variates, one can use "Exact" Optimization Methods (e.g., L-shaped Method) to solve it. In fact, the first-stage decision is obtained this way. To test the solution quality at the current stage, qualitative upper and lower bounds are provided for the optimal objective function value and these are discussed in the following two subsections.

4.2 Upper Bound

Consider a general multi-stage stochastic programming model as shown in the APPENDIX. Note that for simplicity of the following descriptions, theoretical foundations of stochastic programming including some well-known solutions (e.g. the expected value strategy and wait-and-see bound) are also included in the APPENDIX. According to Morton (2002), the commonly used method for setting upper bounds is to guess a feasible solution $\hat{x} \in X$ and then estimate $c\hat{x} + Eh(\hat{x}, \tilde{\xi})$ by sampling; i.e., from $\overline{U}(n) = c\hat{x} + \frac{1}{n}\sum_{i=1}^{n} h(\hat{x}, \tilde{\xi}^{i})$ where $\tilde{\xi}^{1}, \cdots, \tilde{\xi}^{n}$ are iid. from the distribution of $\tilde{\xi}$. Obviously, the associated confidence intervals can be determined. There are many methods that can be used for obtaining upper bounds. Some methods for obtaining the "guess" $\hat{x}$ are more sophisticated than others. Such methods include:

(1) $\hat{x} = x_{EV} \in \arg\min_{x \in X}[cx + h(x,\bar{\xi})]$, where x is replaced by the expected value.

(2) $\hat{x}$ may be the average of some sampled wait-and-see solutions. That is, $x^{i} \in \arg\min_{x \in X}[cx + h(x,\tilde{\xi}^{i})]$ where $\tilde{\xi}^{1}, \cdots, \tilde{\xi}^{n}$ are iid $\tilde{\xi}$ and $\hat{x} = \frac{1}{n}\sum_{i=1}^{n} x^{i}$

(3) Let $\tilde{\xi}^{1}, \cdots, \tilde{\xi}^{n}$ be iid from $\tilde{\xi}$. And let $\hat{x}$ be a solution of $\min_{x \in X}\ [cx + \frac{1}{n}\sum_{i=1}^{n} h(x,\tilde{\xi}^{i})]$.

(4) Stochastic Quasi-gradient Methods.

$$x^{k+1} = P_X[x^k + t^k[c + \frac{1}{n}\sum_{i=1}^{n} \nabla_x h(x^k, \tilde{\xi}^i)]]$$

x^k = current solution

$c + \frac{1}{n}\sum_{i=1}^{n} \nabla_x h(x^k, \tilde{\xi}^i)$ = an approximation of the gradient [a quasigradient]

t^k = steplength

P_X = projection onto 1st stage feasible region.

In this paper the third method is used to obtain the upper bound.

4.3 Lower Bound

The theorem for A Monte Carlo Method for Lower Bounds is given in the APPENDIX. Note that the method to get the lower bound is similar to that for the upper bound except that only the first-stage decision is needed.

The following sections discuss the solution procedures of how the upper bound and lower bound are generated using this approach. For simplicity and without loss of generality, an N-stage stochastic programming model is illustrated using the Monte Carlo Sampling Based Method. The solution procedures are described as follows.

4.4 Solution Methodology

The whole procedure for obtaining lower and upper bounds are as follows:

<u>Input</u>: Related required data for SDVAP-N include r_{ij} , the reward for satisfying a unit of demand on each arc, and c_{ij} , the cost for running an unloaded truck on each arc, and the confidence level α. (For the lower bound, the sample size (i.e., the number of scenario trees) is n_l and batch size (i.e., the number of scenarios inside a tree) is n_b. For the upper bound, the sample size is n_u and batch size (i.e., the number of scenarios inside a tree) is n_b. One sample tree for generating lower and upper bounds is given in Figure 3.

Output: Candidate solution $\hat{x}$, $\hat{y}$ and approximate (1-2α)-level confidence interval

$$[0,(\overline{U}-\overline{L})^{+}+\tilde{\varepsilon}_u+\tilde{\varepsilon}_L] \text{ on } r\hat{x}-c\hat{y}+Eh(\hat{x},\hat{y},\tilde{\xi})-z^{*}.$$

The whole procedure can be described as follows:

Step 1: Set stage=1.

do i=1 to n_l

Generate a scenario tree as show in Figure 3.1 and solve the generated tree.

Obtain $\hat{x}^{i}$, $\hat{y}^{i}$ as the optimal solution and L^{i} as the optimal objective values.

enddo

Form $\overline{L}(n_l)=\frac{1}{n_l}\sum_{i=1}^{n_l}L^{i}$, and $S_L^2(n_l)=\frac{1}{n_l-1}\sum_{i=1}^{n_l}\left(L^{i}-\overline{L}(n_l)\right)^2$

Calculate optimal solution for stage 1 as: $\hat{x}_{\text{stage}}=\frac{1}{n_l}\sum_{i=1}^{n_l}\hat{x}^{i}$, $\hat{y}_{\text{stage}}=\frac{1}{n_l}\sum_{i=1}^{n}\hat{y}^{i}$.

Step2: Set stage=stage+1.

do i=1 to n_l

Generate a scenario tree as show in Figure 3.stage.

do tmp_stage=1 to stage

Fix the decision variables in previous stages as $\hat{x}_{\text{stage}}$, $\hat{y}_{\text{stage}}$.

enddo

Solve the generated tree as shown in Figure 3.stage.

Obtain $\hat{x}^i$ and $\hat{y}^i$ as the optimal solution.

enddo

Calculate $\hat{x}_{\text{stage}} = \frac{1}{n_l}\sum_{i=1}^{n_l}\hat{x}^i$, $\hat{y}_{\text{stage}} = \frac{1}{n_l}\sum_{i=1}^{n_l}\hat{y}^i$.

If stage<N-1, goto Step2, else goto Step 3.

Step3: do i=1 to n_u

do tmp_stage=1 to stage-1

Fix all the decision variables in tmp_stage as $\hat{x}_{\text{tmp\_stage}}$, $\hat{y}_{\text{tmp\_stage}}$.

enddo

Generate a scenario tree as show in Figure 4.

Form $\overline{U}(n_u) = \sum_{t=1}^{T-1}\left(r\hat{x}_t - c\hat{y}_t\right) + \frac{1}{n_u}\sum_{i=1}^{n}\left(r\hat{x}_T - c\hat{y}_T\right)$ and

$$S_u^2(n_u) = \frac{1}{n_u - 1}\sum_{i=1}^{n}\left(r\hat{x}_T - c\hat{y}_T - \frac{1}{n_u}\sum_{i=1}^{n}\left(r\hat{x}_T - c\hat{y}_T\right)\right)^2.$$

Step 4: Construct $\tilde{\varepsilon}_u = \frac{t_{n_u-1,\alpha}S_u(n_u)}{\sqrt{n_u}}, \tilde{\varepsilon}_L = \frac{t_{n_L-1,\alpha}S_L(n_L)}{\sqrt{n_L}}$ and the (1-2α)-level confidence

interval $[0, \left(\overline{U} - \overline{L}\right)^+ + \tilde{\varepsilon}_u + \tilde{\varepsilon}_L]$.

It is noted that Step 1 generates the lower bound procedure and Step 2 produces the feasible solutions through the methods mentioned before. Step 3 evaluates all the feasible solutions in the previous N-1 stages in a stochastic environment (as shown in Figure 4) for N-th stage and gets the upper bound for the optimal solution.

5. Numerical Results

5.1 Example Network Descriptions

A 5-stage experimental network was designed to show the quality of the solution when the Monte Carlo sampling based method is used to solve SDVAP. The following describes the example input information required for SDVAP.

A planning horizon of 5 days and freight transportation between four cities are taken in this model instance. The net revenue per loaded truck (r_{ij}) and the cost per empty truck (c_{ij}) are expressed in matrices and both use \$ as the unit. Also as the previous notations showed, d_{ij1} is the demand at day 1 and d_{ijk}, k=2,3,4,5 denotes the mean value of the random demands (Poisson distributed) at day 2 ~ 5 respectively. In addition, S_{i1} is the initial number of trucks available in region i at day 1. (All demands are expressed as units of "trucks".) All the data are summarized as follows:

N = 5. R = 4. α=0.05.

$$r_{ij} = \begin{pmatrix} 8 & 10 & 9 & 15 \\ 10 & 11 & 8 & 17 \\ 9 & 8 & 21 & 19 \\ 15 & 17 & 19 & 15 \end{pmatrix} c_{ij} = \begin{pmatrix} 0 & 3 & 4 & 4 \\ 3 & 0 & 4 & 5 \\ 4 & 4 & 0 & 6 \\ 4 & 5 & 6 & 0 \end{pmatrix} d_{ij1} = \begin{pmatrix} 18 & 8 & 12 & 10 \\ 6 & 10 & 17 & 8 \\ 10 & 14 & 9 & 7 \\ 8 & 16 & 12 & 14 \end{pmatrix}$$

$$d_{ijk} = \begin{pmatrix} 12 & 9 & 8 & 15 \\ 6 & 9 & 14 & 7 \\ 9 & 13 & 10 & 6 \\ 8 & 18 & 11 & 16 \end{pmatrix} k\text{=2,3,4,5. } S_{i1} = \begin{pmatrix} 40 \\ 55 \\ 30 \\ 50 \end{pmatrix}$$

As described before, SDVAP is a typical multistage stochastic programming problem. Although it involves vehicle allocation within only four cities, this is a large problem. In each stage, the demands on all of the 16 arcs are stochastic. Assume the simplest case that on each arc, there are only two situations, namely high demand and low demand. Then, the total number of scenarios of this problem is $2^{16\times4} = 2^{64}$. Therefore, even in such a simple case, to solve it using regular deterministic stochastic programming methods is impossible. Furthermore, in this model, the demands are Poisson distributed, making it intractable using traditional stochastic programming methods. Therefore, the Monte Carlo Sampling Based Method is a reasonable choice. In this paper, GAMS with CPLEX Solver is used as the optimization tool and the main codes have been successfully tested.

5.2 Computational Results

It should be noted that when CPLEX is involved with generating random variates, the speed of code executions is sharply reduced. To expedite the simulation process, a C++ programming code was developed to generate all the random demands (that are needed and Poisson distributed) in all the scenario trees. These data are written into a file as inputs for the GAMS codes.

The most promising property of using the Monte Carlo Sampling Based Method to solve the SDVAP problem is that the execution time is very short. About 1 minute is required for generating all the input data and about 68 seconds for the GAMS code execution of the scenario tree. Furthermore, as will see in the next sections, the computational quality of the solution is very good. Obviously, these two characteristics strongly suggest that this Monte Carlo Sampling Based Method could be applied to solve the SDVAP problem efficiently and effectively.

The following matrix shows the recommended loaded movement (x_{ij1}), empty movement (y_{ij1}) and overall movement (M_{ij1}) of trucks for day 1 through the Monte Carlo Sampling Based Method.

$$x_{ij1} = \begin{pmatrix} 18 & 8 & 4 & 10 \\ 6 & 10 & 17 & 8 \\ 10 & 4 & 9 & 7 \\ 8 & 16 & 12 & 14 \end{pmatrix}, \qquad y_{ij1} = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 2 & 3 & 0 & 9 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix},$$

$$M_{ij1} = x_{ij1} + y_{ij1} = \begin{pmatrix} 18 & 8 & 4 & 10 \\ 8 & 13 & 17 & 17 \\ 10 & 4 & 9 & 7 \\ 8 & 16 & 12 & 14 \end{pmatrix}$$

As seen, the first-stage decision (M_{ij1}) provides the optimal movement of the number of trucks from region i to region j in a matrix form. (e.g., 18 trucks should be moved from city 1 is city 1 and the number of trucks needed to move from city 1 to city 3 is 4, etc.) As a result, the trucks available at each region on the second day are:

$$S_{i2}^{\mathrm{T}} = \begin{pmatrix} 44 & 41 & 42 & 48 \end{pmatrix}^{\mathrm{T}}$$

Comparing this recommended allocation of trucks with the demands at day 1, one can see that some demands are refused while some vehicles are assigned to move empty. This is as expected because the goal to maximize profits results in unprofitable demands (with lower net revenue) not being satisfied and spare empty trucks are allocated to a more favorable allocation position to serve future demands with potentially higher net revenue.

The lower bound procedure is solved as mentioned in the solution methodology. The estimated lower bound for the objective function is: $\overline{\mathrm{L}}$ = - \$ 10055, which means that the company can earn at most 10055 dollars.

To examine the quality of the recommended allocation of trucks at day 1, one might estimate both an upper and a lower bound for the objective function. In a multi-stage problem, not only the first stage decision but also feasible decisions at subsequent stages except the last stage are required in the upper bound procedure. The detailed steps, as illustrated before are used to derive those intermediate-stage allocations. The following matrices list the feasible resulting allocations of trucks for days 2, 3 and 4.

$$M_{ij2} = \begin{pmatrix} 14 & 9 & 8 & 13 \\ 9 & 9 & 12 & 11 \\ 10 & 12 & 12 & 8 \\ 8 & 13 & 11 & 16 \end{pmatrix},\ M_{ij3} = \begin{pmatrix} 14 & 7 & 7 & 13 \\ 10 & 10 & 13 & 10 \\ 11 & 12 & 12 & 8 \\ 8 & 13 & 10 & 17 \end{pmatrix},\ M_{ij4} = \begin{pmatrix} 14 & 8 & 8 & 13 \\ 9 & 11 & 11 & 11 \\ 10 & 12 & 14 & 6 \\ 8 & 13 & 11 & 16 \end{pmatrix}$$

Given the decisions on days 1, 2 3 and 4, the estimated upper bound for the objective is: $\overline{\mathrm{U}}$ = - \$ 9647. Based on the estimated lower bound and upper bound, a confidence interval is also constructed, as shown in Table 1. The sample mean for the lower bound means that the company can earn an expected profit of no more than 10055 dollars. The sample mean for the upper bound means that the practicing company can earn at least an expected profit with 9647 dollars. And one can say that the quality of the solution using Monte Carlo Sampling Based Method: “I am 90% percent sure that the different between the expectation of the upper bound and the analytical optimal expectation of profits falls in between the interval [0, $442]. Namely, statistically if I strictly follow the decision that was made in the first stage using this method, the worst I can earn is 442 dollars less than the best I can possibly do.” Therefore, this bound is very tight and looks very good.

It is interesting to know that for the expected value strategy, that is to say, if the problem is solved replacing random demands by their expected values, the result is: EV = - $ 10689, which means that one can earn 10689 dollars. This is expected because in this case, the problem has changed from a stochastic programming problem to a deterministic optimization problem. When the demand is deterministic instead of random, one has perfect information about the demand, as a result, one can get a better solution. Also mathematically it is noted that the magnitude of the objective function for the expected value problem is less than that of the stochastic problem, which is in accord with the principle of Jensen’s inequality. Note that this is also the commonly used approach in the truck industry, where the first-stage decisions are made usually only based on the expected demand strategy. However, if the solution of the expected value problem is evaluated in the random demand environment, (namely, if the decision made at each stage is strictly followed but the demand is random instead of deterministic,) the objective function of the stochastic problem becomes: EEV = - $ 9388, which is higher than the estimated upper bound $\overline{U}$. This is as expected because this is not an optimal solution for the stochastic programming problem. Therefore, one can only earn an expectation of 9388 dollars, which is much less than that one can get from the optimal solution of the Monte Carlo Sampling Based Method.

Another interesting thing one might want to know is that a wait-and-see solution is also obtained when solving several separate stochastic linear programming models each consisting of a separate (unrelated) sampling tree as shown in Figure 5. The statistically optimal solution is WS= - $ 10128. Note that the objective of the wait-and-see problem is statistically less than that of the stochastic problem. This is as expected because one can always earn more if perfect information about the demand at each stage is known. However, it is higher than that of the expected value problem, which is predicted by the theorem that WS ≥ EV if only right-hand-side variables are random (See APPENDIX for more information). In addition, when the solution at each stage of the wait-and-see problem is evaluated in the random demand environment, the objective function of the stochastic problem has the value of: EWS= - $ 9636, which is higher than the estimated upper bound $\overline{U}$, indicating a worse solution (worse upper bound) than that of the previous method. That is to say, the upper bound using this wait-and-see solution is lousy----one can possibly earn an expectation of 9636 dollars, which is worse than the solution from the Monte Carlo Sampling Method ($9647). Although this amount seems to be insignificant (one might argue), the aggregated value can be quite attractive when Monte Carlo Sampling Based Method is used to solve a large network under a long time horizon (say annually or in principle, infinite), especially when the scale or absolute value of the net revenue of each truck movement enlarges.

6. Conclusions

Optimization under uncertainty has seen many real world applications. The stochastic dynamic vehicle allocation problem (SDVAP) is faced by trucking companies, container companies, rental car agencies and railroads. To maximize profits in a competitive industry, scientific tools and computer-based advanced algorithms are needed to manage fleets of vehicles in both time and space.

This paper has formulated a multistage stochastic programming based model for SDVAP. A Monte Carlo Sampling Based Algorithm has been proposed to solve SDVAP. A probabilistic statement regarding the quality of the solution from the Monte Carlo sampling method is also identified by introducing a lower bound and an upper bound of the obtained optimal solution. A five-stage experimental network was introduced for demonstration of this algorithm. The computational results indicated a solution of high quality when Monte Carlo sampling based algorithm is used to solve SDVAP, suggesting that these algorithms can be used for real world applications.

Due to the limitation of computation time and solver capability, the problem solved for demonstration in this paper is a small network with a short time horizon. It is expected that as the sample size increases, the lower bound will become larger and a better probabilistic statement can be obtained. Further research is expected on the computation for large networks with large-sample-sized Monte Carlo Sampling Based Methods for solving SDVAP problems.

7. Acknowledgements

This paper is based on course project work from Stochastic Programming. Many thanks are given to other group members, including Xiao Qin and Lin Wan, for their help in the previous example network. The authors also would like to thank Dr. Morton for his intensive and incisive teaching in Stochastic Programming.

References:

1. BEALE E. M., DANTZIG G. B. and WATSON R. D. A First Order Approach to a Class of Multi-time Period Stochastic Programming Problems. *Mathematical Program Study* 27, 103-177 (1986)

2. BIRGE J. Decomposition and Partitioning Techniques for Multistage Stochastic Linear Programs. *Operations Research* 33, 989-1007 (1985)

3. BOOKBINDER J. H. and SETHI, S. P. The Dynamic Transportation Problem: A Survey. *Naval Research Logistic Quarterly* 27, 447-452 (1980)

4. CHEUNG, R. K. and W. POWELL. SHAPE: A Stochastic Hybrid Approximation Procedure for Two-Stage Stochastic Programs. *Operations Research* 48(1), 73-79 (2000)

5. CHEUNG, R. K. and CHEN, C-Y., A two-stage stochastic network model and solution methods for dynamic empty container allocation problem, *Transportation Science* 32, No. 2, 142-162 (1998)

6. DANTZIG G. Linear Programming under Uncertainty. *Management Science* 1, 197-206 (1955)

7. DANTZIG G. and WOLFE P. Decomposition Principle for Linear Programs. *Operations Research* 8, 101-111 (1960)

8. DEJAX P. J. and CRAINIC T. G. A Review of Empty Flows and Fleet Management Models in Freight Transportation. *Transportation Science* 21, 227-247 (1987)

9. FLETEN S-E, HOYLAND K. and WALLACE S. W. The Performance of Stochastic Dynamic and Fixed Mix Portfolio Models. *European Journal of Operational Research* 140, 37-49 (2002)

10. FRANTZESKAKIS L. F. and POWELL W. B. A Successive Linear Approximation Procedure for Stochastic, Dynamic Vehicle Allocation Problems. *Transportation Science* 24, 40-57 (1990)

11. HOYLAND K. and WALLACE S. W. Generating scenario trees for multistage decision problems. *Management Science* 47, No. 2, 295-307 (2001)

12. JORDAN W. C. and TURNQUIST M. A. A Stochastic, Dynamic Model for Railroad Car Distribution. *Transportation Science* 17, 123-145 (1983)

13. KOUWENBERG R. Scenario Generation and Stochastic Programming Models for Asset Liability Management. *European Journal of Operational Research* 134, 279-292 (2001)

14. MORTON DAVID P. Stochastic Optimization Class Notes. The University of Texas at Austin, (2002)

15. POWELL W. B. A Stochastic Model of the Dynamic Vehicle Allocation Problem. *Transportation Science* 20, 117-129 (1986)

16. WALLACE S. Solving Stochastic Programs with Network Recourse. *Networks* 16, 295-317 (1986)

17. WETS R. Solving Stochastic Programs with Simple Recourse. *Stochastics* 10, 219-242 (1983)

18. WOLLMER R. Two State Linear Programming Under Uncertainty with 0-1 Integer First Stage Variables. *Mathematical Programming* 19, 279-288 (1980)

19. ZENIOS S. A. etc. Dynamic Models for Fixed-income Portfolio Management under Uncertainty. *Journal of Economic Dynamics and Control* 22, 1517-1541 (1998)

20. ZIEMBA W. Computational Algorithms for Convex Stochastic Programs with Simple Recourse. *Operations Research* 18, 414-431 (1970)

LIST OF FIGURES

LIST OF TABLES

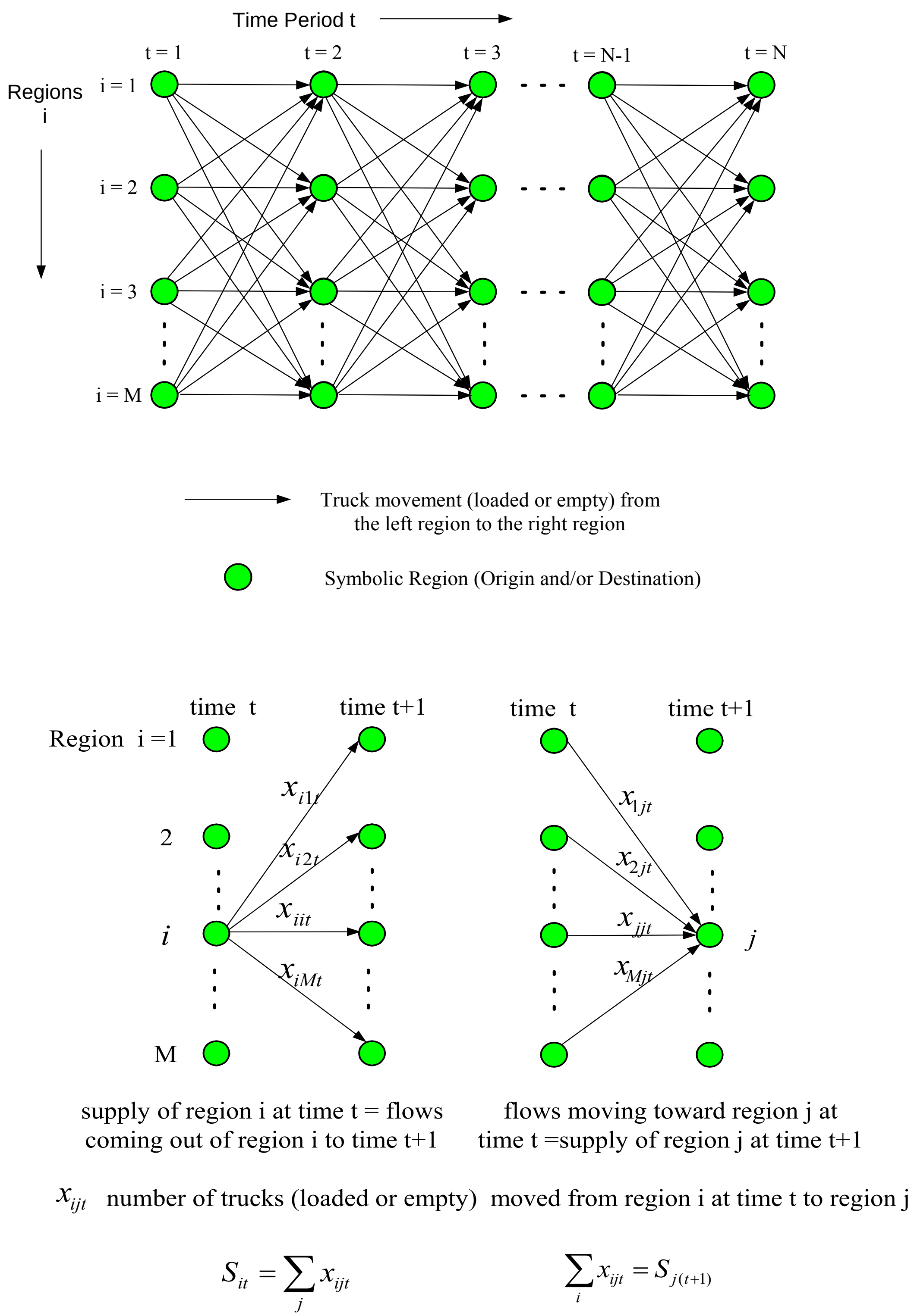

Figure 1. Network Flow Representation of SDVAP

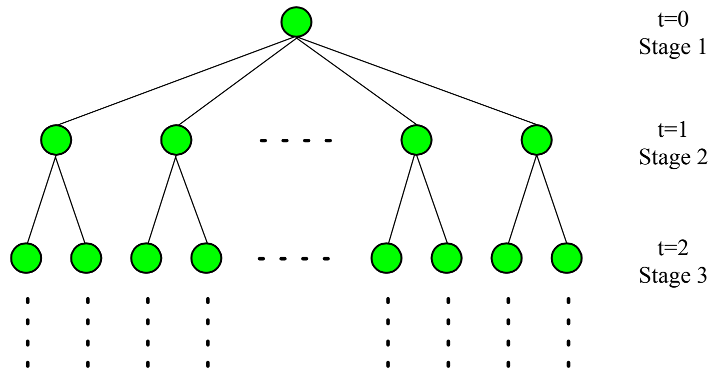

Figure 2 Scenario Tree for Multi-stage Stochastic Programming Models

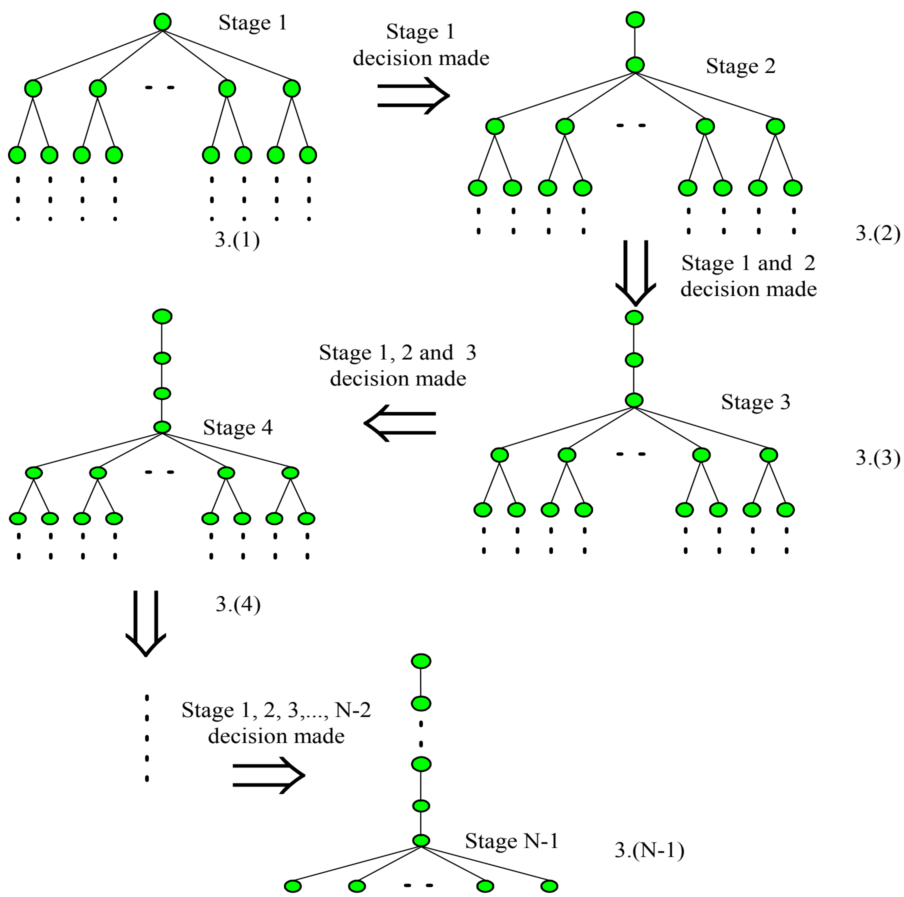

Figure 3. Lower bound and upper bound determination. Note that only the demands at day 1 are deterministic and the initial number of trucks at day k (k>1) is given from the feasible solutions that are calculated in previous stages.

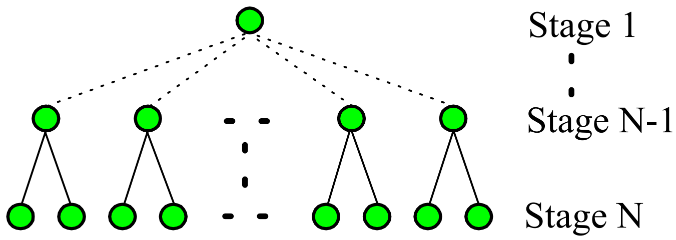

Figure 4. Evaluation of lower bound and upper bound solution under stochastic environment. Note that only the demands at day 1 are deterministic and demands at day 2 ~ N are randomly Poisson distributed.

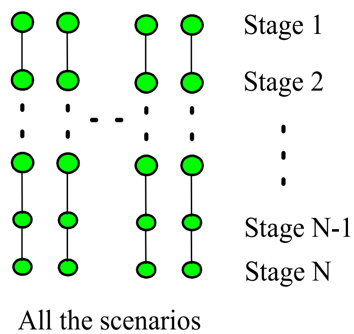

Figure 5. Wait and See Solution Strategy

Table 1. Confidence Interval

	Sample mean	Sample standard variance	Sample size	t	ε
lower bound	-$10055	69.34	30	1.697	21.49
upper bound	-$9647	244.52	1000	1.645	12.72
90% CI for ($E\widetilde{U} - Z^*$): [0, $ 442]					

APPENDIX Properties of Stochastic Programming Problems

Suppose one want to deal with a stochastic programming problem:

$$
\begin{aligned}
&\min \ cx + Eh(x,\tilde{\xi}) \\
&\text{s.t.} \quad x \in X \\
&\qquad X = \{x : \ Ax = b, \ x \geq 0\} \\
&\text{where} \\
&\qquad h(x,\tilde{\xi}) = \min \ \tilde{f}\, y \\
&\qquad\qquad \text{s.t.} \quad \tilde{D}y = \tilde{B}x + \tilde{d} \\
&\qquad\qquad\qquad y \geq 0 \\
&\tilde{\xi} = \text{vec}(\tilde{D},\tilde{d},\tilde{B},\tilde{f})
\end{aligned}
$$

Lower Bound and Upper bound for Solution from Monte Carlo Sampling Based Method:

Theorem 1: Let $\tilde{\xi}^1, \tilde{\xi}^2 \cdots, \tilde{\xi}^n$ be iid (independently identically distributed) from $\tilde{\xi}$, and let $\tilde{L} = \min_{x \in X}\left[cx + \frac{1}{n}\sum_{i=1}^{n} h(x,\ \tilde{\xi}^i)\right]$ and let $\overline{U}(n) = c\hat{x} + \frac{1}{n}\sum_{i=1}^{n} h(\hat{x},\ \tilde{\xi}^i)$ ($\hat{x} \in X$ are any feasible solutions). Then, $E\tilde{L} \leq Z^* \leq E\tilde{U} = E\overline{U}(n)$ where $Z^* = \min_{x \in X}\ cx + Eh(x,\tilde{\xi})$.

Theorem 2: Let $\overline{U}(n_u) = c\hat{x} + \frac{1}{n_u}\sum_{i=1}^{n_u} h(\hat{x},\ \tilde{\xi}^i)$ ($\hat{x} \in X$ are any feasible solutions) and $S_u^2(n_u) = \frac{1}{n_u - 1}\sum_{i=1}^{n_u}\left[c\hat{x} + h(\hat{x},\tilde{\xi}^i) - \overline{U}(n_u)\right]^2$. Let $\tilde{\xi}^{i1}, \tilde{\xi}^{i2}, \cdots, \tilde{\xi}^{im}$ are independently identically distributed variables from $\tilde{\xi}$, i=1,…,n and form $\tilde{L}^i = \min_{x \in X}\left[cx + \frac{1}{m}\sum_{j=1}^{m} h(x,\ \tilde{\xi}^{ij})\right]$. Let $\overline{L}(n_l) = \frac{1}{n_l}\sum_{i=1}^{n_l}\tilde{L}^i$ and $S_L^2(n_l) = \frac{1}{n_l - 1}\sum_{i=1}^{n_l}\left(\tilde{L}^i - \overline{L}(n_l)\right)^2$.

Let $\tilde{\varepsilon}_u = \frac{Z_\alpha S_u(n_u)}{\sqrt{n_u}}$ and $\tilde{\varepsilon}_L = \frac{Z_\alpha S_L(n_L)}{\sqrt{n_L}}$. Then an (1-2α) confidence interval for $E\tilde{U} - Z^*$ is: $[0, \left(\overline{U} - \overline{L}\right)^+ + \tilde{\varepsilon}_u + \tilde{\varepsilon}_L]$.

Theorem 3 (Jensen's Inequality): If $f(\bullet)$ is convex and $\tilde{\xi}$ is a random vector. Then, $Ef(\tilde{\xi}) \geq f(E\tilde{\xi})$. For the above-mentioned general stochastic programming, if $h(x,\bullet)$ is convex, then: $Eh(x,\tilde{\xi}) \geq h(x,E\tilde{\xi}) \qquad \forall x \in X$

Theorem 4: Let $EV = \min_{x \in X} cx + h(x, E\tilde{\xi})$ (expected value strategy), $RP = \min_{x \in X} cx + Eh(x, \tilde{\xi})$ (recourse problem) and $WS = E[\min_{x \in X} cx + h(x, \tilde{\xi})]$ (wait and see bound). Also, let $x^*_{EV} \in \operatorname{argmin}_{x \in X}[cx + h(x, E\tilde{\xi})]$, $x^*_{RP} \in \operatorname{argmin}_{x \in X}[cx + Eh(x, \tilde{\xi})]$ and $EEV = cx^*_{EV} + Eh(x^*_{EV}, \tilde{\xi})$. If $\tilde{\xi} = \mathrm{vec}(\tilde{d})$ and B, f and D are deterministic, then: $EV \leq WS \leq RP \leq EEV$.