Introduction to Stochastic Programming

Fatih Cavdur

November 7, 2015

Stochastic Programming

- Stochastic Programming (SP) is an approach to for modeling problems that involve uncertainty.
- Although deterministic problems are formulated with known parameters, real-life problems often include parameters that are unknown at the time a decision should be made.
- When the parameters are uncertain, but assume some given set of possible values, we can seek a solution that is feasible for all possible parameter choices and optimizes a given objective function.

Stochastic Programming
Example

A company has to decide an order quantity x of a certain product to satisfy demand d. The cost of ordering is $c > 0$ per unit. If the demand is greater than the order quantity, a back order penalty of $b \geq 0$ per unit is incurred. If the demand is less than the order quantity, the company has a holding cost of $h \geq 0$ per unit. The total cost to minimize is then written as

$$\begin{aligned} G(x, d) &= cx + b[d - x]^+ + h[x - d]^+ \\ &= \max\{(c - b)x + bd, (c + h)x - hd\} \end{aligned}$$

Stochastic Programming
Example

We want to minimize the total cost.

$$\min_{x \geq 0} G(x, d)$$

For a numerical instance, if $c = 1$, $b = 1.5$ and $h = 0.1$,

$$G(x, d) = \begin{cases} -0.5x + 1.5d, & x < d \\ \ \ 1.1x - 0.1d, & x \geq d \end{cases}$$

Stochastic Programming

Example

If we must make the decision before a realization of the demand becomes known, we can view the demand D as a random variable whose probability distribution is known. We can then write the corresponding optimization problem

$$\min_{x \geq 0} \mathbb{E}[G(x, d)]$$

which minimizes the total cost on *average*. If the process repeats itself, by the Law of Large Numbers, for a given x, the average of the total cost will converge to the expectation.

Stochastic Programming

Example

A simple example of a recourse action is as follows:

- At the first stage, before the realization of the demand D, a decision is made about the ordering quantity, x.
- At the second stage, after the demand D is known, it may happen that $d > x$.
- In that case, the demand can be met by taking the recourse action of ordering the required quantity $d - x$ at a penalty cost of $b > c$.

Stochastic Programming

Example

Example problem can be solved in closed form. It is possible to show that

$$\mathbb{E}[G(x, D)] = b\mathbb{E}[D] + (c - b)x + (b + h)\int_0^x F(z)dz$$

By setting,

$$\frac{d\mathbb{E}[G(x, D)]}{dx} = 0 \Rightarrow (b + h)F(x) + c - b = 0 \Rightarrow x = F^{-1}(\kappa)$$

where κ is the quantile defined as

$$\kappa = \frac{b - c}{b + h}$$

Stochastic Programming

Example

- Suppose that the RV D has a discrete distribution assuming values $d_1, \ldots, d_K$ (scenarios) with respective probabilities $p_1, \ldots, p_K$.
- The corresponding CDF is viewed as the empirical CDF giving an approximation or estimation of the true CDF, and the associated κ-quantile is viewed as the sample estimate of the κ-quantile associated with the true distribution.
- If we compare the quantile solution to the $\bar{x}$ with a solution corresponding to one scenario $d = \bar{d}$, where $\bar{d}$ is, say, the expected value of RV D.
- The solution of such deterministic problem is $\bar{d}$ which can very different from the κ-quantile of $\bar{x}$.

Stochastic Programming

Example

A closed-form solution as in the example is rarely available. As a result, in case of finitely many scenarios, it is possible to model the stochastic program as a deterministic problem, by writing the expected value as the weighted sum

$$\mathbb{E}[G(x, D)] = \sum_{k=1}^{K} p_k G(x, d_k)$$

Stochastic Programming

Example

The deterministic formulation corresponds to one scenario d with probability 1 which can be written

$$\min_{x,t} t$$

subject to

$$\begin{aligned} t &\geq (c - b)x + bd \\ t &\geq (c + h)x + hd \\ x &\geq 0 \end{aligned}$$

Stochastic Programming
Example

Similarly, the expected value problem with scenarios $d_1, \ldots, d_K$ can be written as the following linear program.

$$\min_{x; t_1, \ldots, t_K} \sum_{k=1}^{K} p_k t_k$$

subject to

$$\begin{aligned} t_k &\geq (c-b)x + bd_k, \; k = 1, \ldots K \\ t_k &\geq (c+h)x + hd_k, \; k = 1, \ldots K \\ x &\geq 0 \end{aligned}$$

Stochastic Programming
Example

- Using approximations by scenarios is an attractive approach for attacking SPPs.
- We can investigate the convergence of the solution of such a scenario approximation.
- We also note the "almost-separable" structure of the problem.
- That is, for fixed x, the problem is separable into the sum of optimal values of problems with $d = d_k$.
- Such separable structure is typical for *two-stage linear* SPPs.

Stochastic Programming

Example

To compare the exact solution of the problem with the scenario solution with $c = 1.0$, $b = 1.5$ and $h = 0.1$, suppose that D is uniformly distributed between 1 and 100. We can then write

$$\begin{aligned}\mathbb{E}[G(x, D)] &= b\mathbb{E}(D) + (c - b)x + (b + h)\int_0^x F(z)dz \\ &= 75 - 0.5x + 0.008x^2\end{aligned}$$

Stochastic Programming

Example

If the demand is approximated by a discrete distribution with equally likely scenarios $d_1 = 20$ and $d_2 = 80$,

$$\mathbb{E}[G(x, D)] = \frac{1}{2}\sum_{k=1}^{2} G(x, d_k)$$

What if we have 3 scenarios with $d_1 = 20$, $d_2 = 50$ and $d_3 = 80$?

Two-Stage Stochastic Programming

We can formulate the classical two-stage stochastic programming problems as

$$\min_{x \in X} \{g(x) := cx + \mathbb{E}[Q(x, \xi)]\}$$

where $Q(x, \xi)$ is the optimal value of the second-stage problem

$$\min_{y} qy : Tx + Wy \leq h$$

where x is the first-stage decision vector, X is a polyhedral set, y is the second-stage decision vector and $\xi = (q, T, W, h)$ is the second-stage problem data.

Two-Stage Stochastic Programming

- At the first stage, we make a "here-and-now" decision before the realization of the uncertain data ξ is known.
- At the second stage, after a realization of ξ becomes available, we optimize our behavior by solving an appropriate problem.
- Note that, at the first stage, we minimize the cost cx of the first-stage decision plus the expected cost of the (optimal) second-stage decision.
- We can then view the second-stage problem simply an optimization problem which describes our supposedly optimal behavior when the uncertain data is revealed.
- Or, we can consider its solution as a *recourse action* where the term Wy compensates for a possible inconsistency of the system $Tx \leq h$ and qy is the cost of this recourse action.

Two-Stage Stochastic Programming

- The formulation involves the assumption that the second-stage data ξ can be modeled as a *random* vector with a *known* probability distribution.
- This would be justified in situations where the problem is solved repeatedly under random conditions which do not significantly change over the considered period of time.
- In such situations, we can estimate the required distribution and the optimization *on average* could be justified the Law of Large Numbers.

Two-Stage Stochastic Programming

To solve the problem numerically, the standard approach is to assume that random vector ξ has a finite number of possible realizations, called *scenarios*, say $\xi_1, \ldots, \xi_K$, with respective (positive) probabilities $p_1, \ldots, p_K$. We can then write the expectation as

$$\mathbb{E}[Q(x,\xi)] = \sum_{k=1}^{K} p_k Q(x,\xi_k)$$

Two-Stage Stochastic Programming

We can then write the two-stage problem as

$$\min_{x; y_1, \ldots, y_K} cx + \sum_{k=1}^{K} q_k y_k$$

subject to

$$x \in X, \;\; T_k x + W_k y_k \leq h_k, \;\; k = 1, \ldots, K$$

Two-Stage Stochastic Programming

Note that, in this formulation, we make one copy y_k, of the second stage decision vector, for every scenario $\xi_k = (q_k, T_k, W_k, h_k)$. By solving the problem, we obtain an optimal solution $\bar{x}$ of the first-stage problem and optimal solutions $\bar{y}_k$ of the second-stage for each scenario ξ_k, $k = 1, \ldots, K$. Given $\bar{x}$, each y_k gives an optimal second-stage decision corresponding to a realization $\xi = \xi_k$ of the respective scenario.

Two-Stage Stochastic Programming

Example

Recall the inventory model with K scenarios written as follows, where x and t_k, $k = 1, \ldots, K$ are the first and second-stage decisions, respectively.

$$\min_{x;t_1,\ldots,t_K} 0x + \sum_{k=1}^{K} p_k t_k$$

subject to

$$\begin{aligned}
(c-b)x - t_k &\leq -bd_k, \;\; k = 1, \ldots K \\
(c+h)x - t_k &\leq +hd_k, \;\; k = 1, \ldots K \\
x &\geq 0
\end{aligned}$$

Two-Stage Stochastic Programming

When ξ has an infinite (or very large) number of possible realizations, then, we can represent this distribution by scenarios. We should then answer,

- How to construct scenarios?
- How to solve the LP?
- How to measure the quality of the solutions with respect to the 'true' optimum?

Scenario Construction

- In practice, it might be possible to construct scenarios by eliciting experts' opinion on the future.
- We would like the number of constructed scenarios to be relatively modest so that the obtained LP requires reasonable computation effort.
- It is often claimed that a solution that is optimal using a few scenarios is better than the one that assumes a single scenario only.
- In some cases, such a claim could be verified by a simulation.

Scenario Construction

- In theory, we would want some measure of guarantee that an obtained solution solves the original problem with reasonable accuracy.
- Also note that typically in applications only the *first* stage optimal solution $\bar{x}$ has a practical value since almost certainly a 'true' realization of the random (uncertain) data will be different from the set of constructed scenarios.

Scenario Construction

- The modeler has two somewhat contradictory goals.
- First of all, the number of scenarios should be computationally manageable.
- Secondly, the constructed approximation should provide a reasonably accurate solution to the problem.
- A possible approach to reconcile these contradictory goals is randomization, and thus, scenarios could be generated by Monte-Carlo sampling techniques.

Scenario Construction

- For some feasible $x \in X$ and a scenario ξ_k, the problem might be unbounded, i.e., $Q(x, \xi_k) = -\infty$.
- It means that for such a feasible x, we can improve with a positive probability the second-stage cost indefinitely.
- We should make sure that such a situation does not happen.
- Another problem is that for some feasible $x \in X$ and scenario ξ_k, the second-stage problem is infeasible.
- In that case, we can set $Q(x, \xi_k) = +\infty$.
- It is said that the problem has *relatively complete recourse* if such infeasibility does not happen.

Scenario Construction

It is always possible to make the second-stage problem feasible (complete recourse) by changing it as

$$\min_{y,t} qy + \gamma t : Tx + Wy - te \leq h, \ t \geq 0$$

where γ is constant and e is a vector-of-ones of appropriate dimension.

Monte Carlo Techniques

- We can decrease the number of scenarios by using Monte Carlo simulation.
- Suppose that the total number of scenarios is very large or infinite.
- Suppose also that we can generate $\xi^1, \dots, \xi^N$ of N replications of the random vector ξ which are IID.

Monte Carlo Techniques

We can approximate the expectation function $q(x) = \mathbb{E}[Q(x, \xi)]$ by the average

$$\hat{q}_N(x) = N^{-1} \sum_{j=1}^{N} Q(x, \xi^j)$$

and consequently the 'true' (expectation) problem by

$$\min_{x \in X} \left\{ \hat{g}_N(x) = cx + \frac{1}{N} \sum_{j=1}^{N} Q(x, \xi) \right\}$$

which is known as the *sample average approximation* (SAA) method.

Monte Carlo Techniques

Example

We can illustrate the SAA method on the instance of the inventory example. Note that

$$G(x, D) = \begin{cases} -0.5x + 1.5D, & x < D \\ 1.1x - 0.1D, & x \geq D \end{cases}$$

where $D \sim \mathrm{U}\,(0, 100)$. We can show SAA approximations for 3 random samples, two with $N = 5$ and one with $N = 10$ as

$$\xi^j = 15, 60, 72, 78, 82 \text{ and } \xi^j = 24, 24, 32, 41, 62$$

and

$$\xi^j = 8, 10, 21, 47, 62, 63, 75, 78, 79, 83$$

Evaluating Candidate Solutions

Given a feasible point $\hat{x} \in X$ obtained by solving a SAA problem, a practical problem is how to evaluate the quality of this point. Since the point $\hat{x}$ is feasible, we clearly have that $g(\hat{x}) \geq v^*$, where

$$v^* = \min_{x \in X} g(x)$$

is the optimal solution for the 'true' problem. The quality of the solution can be measured by the optimality gap

$$\text{gap }(\hat{x}) = g(\hat{x}) - v^*$$

Multi-Stage Stochastic Programming

The stochastic programming models up to this point are static in the sense that we made a (supposedly optimal) decision at one point in time, while accounting for possible recourse actions after all uncertainty has been resolved.
There are many situations where decisions should be made sequentially at certain periods of time based on information available at each time period.
Such multi-stage stochastic programming problems can be viewed as an extension of two-stage stochastic programming to a multi-stage setting.

Multi-Stage Stochastic Programming

Example

In the inventory model, suppose that the company has a planning horizon of T periods of time. We view demand D_t as a random process indexed by the time $t = 1, \ldots, T$. At $t = 1$, there is a (known) inventory level y_1, and at each period, the company first observes the current inventory level y_t and then places an order to replenish the inventory level to x_t.

$$\min_{x_t \geq y_t} \sum_{t=1}^{T} \mathbb{E}\left\{c_t(x_t - y_t) + b_t[D_t - x_t]_+ + h_t[x_t - D_t]_+\right\}$$

subject to

$$y_{t+1} = x_t - D_t, \ \ t = 1, \ldots, T-1$$

Multi-Stage Stochastic Programming

Example

At the last stage, $t = T$,

$$\min_{x_T} c_T(x_T - y_T) + \mathbb{E}\left\{b_T[D_T - x_T]_+ + h_T[x_T - D_T]_+ | D_{T-1} = d_{T-1}\right\}$$

subject to $x_T \geq y_T$ with its optimal denoted as $V_T(y_T, d_{[T-1]})$. Similarly, at the first stage, $t = 1$,

$$\min_{x_1 \geq y_1} c_1(x_1 - y_1) + \mathbb{E}\left\{b_1[D_1 - x_1]_+ + h_1[x_1 - D_1]_+ + V_2(x_1 - D_1, D_1)\right\}$$

Thanks... Questions?