Stochastic optimization

Fabian Bastin
fabian.bastin@cirrelt.ca
University of Montreal – CIRRELT

Summer School 2015

Introduction

Consider the general deterministic program

$$\begin{aligned} &\min g_0(x) \\ &\text{s.t. } g_i(x) \leq 0, i = 1, \ldots, m \\ &x \in X \subset \mathbb{R}^n. \end{aligned}$$

All the parameters are assumed to be perfectly known.
Realistic?

- measurement errors;
- uncertainties on the future;
- data unavailable;
- ...

Mathematical programming and stochastic programming

- Mathematical programming (optimization): typically: decision problem (where the meaning of the term "decision" is broad).
- Stochastic programming concerns decision under uncertainty, the uncertainty being represented by means of random parameters.

$$
\begin{aligned}
&\text{"}\min_{x\in X}\text{"}\, g_0(x,\xi) \\
&\text{s.t. } g_i(x,\xi) \leq 0,\ i = 1,\ldots,m,
\end{aligned}
$$

where ξ is a random vector. Meaning of "min"?

Assumption: we can represent the uncertainty by means of the (joint) probability distribution.

The farmer problem

Source: Birge et Louveaux, Section 1.1.

Scenarios approach

- One assumes that the random vector is finite. Each of the realization is a scenario.
- Even if the random vector is continuous, a discrete approximation is often useful.

A European farmer has 500 acres of land and cultivates wheat, corn and sugar beets.

At least 200T of wheat and 240T of corn are needed to feed his livestock. Any additional production can be sold, but in case of underproduction, he has to buy the complement, with a purchase cost 40% greater than the sale cost. The farmer can sold the sugar beets at $36T for the first 6000 tons, and $10T after, due to European quotas.

The farmer problem II

Culture	Wheat	Corn	Sugar beets
Average return (T)	2.5	3	20
Plantation cost ($/acre)	150	230	260
Selling price ($/T)	170	150	36 ($\leq$ 6000T), 10
Buying price ($/T)	238	210	-
Minimum required (T)	200	240	-

Notations:

- x_1, x_2, x_3: acres for wheat, corn, sugar beets;
- y_1, y_2: tons of bought wheat and corn;
- w_1, w_2: tons of sold wheat and corn;
- w_3, w_4: tons of sold sugar beets, at high price and at low price.

How to decide the surface to allocate to each plant?

Linear program:

$$
\begin{aligned}
\min\ & 150x_1 + 230x_2 + 260x_3 + \\
& 238y_1 - 170w_1 + 210y_2 - 150w_2 - 36w_3 - 10w_4 \\
\text{t.q. } & x_1 + x_2 + x_3 \leq 500; \\
& 2.5x_1 + y_1 - w_1 \geq 200; \\
& 3x_2 + y_2 - w_2 \geq 240; \\
& w_3 + w_4 \leq 20x_3; \\
& w_3 \leq 6000; \\
& x_1, x_2, x_3, y_1, y_2, w_1, w_2, w_3, w_4 \geq 0.
\end{aligned}
$$

The farmer problem: deterministic solution

Total profit: \$118600. Details:

Culture	Wheat	Corn	Sugar beets
Surface (acres)	120	80	300
Production (T)	300	240	6000
Sales (T)	100	-	6000
Purchase (T)	-	-	-

The production is however dependant on the weather, and can increase or decrease by 20% to 25%.

In a very simplified setting, assume three possible cases: good year (for every plant, the production is 20% higher), average year, and bad year (for every plant, the production is 20% lower). The prices do not change.

The farmer problem: scenario solutions

New optimal solutions?

Good year. Total profit: \$167667.

Culture	Wheat	Corn	Sugar beets
Surface (acres)	183.33	66.67	250
Production (T)	550	240	6000
Sales (T)	350	-	6000
Purchases (T)	-	-	-

Bad year. Total profit: \$59950.

Culture	Wheat	Corn	Sugar beets
Surface (acres)	100	25	375
Production (T)	200	240	6000
Sales (T)	-	-	6000
Purchases (T)	-	180	-

The farmer problem: scenarios

The decisions considerably change with the weather conditions, but how to know them when deciding what to plant?
The decisions (x_1, x_2, x_3) have to be made now, but sales and purchases $(w_i, i = 1, \dots, 4, y_j, j = 1, 2)$ depend on yields.

Scenarios.
Index $s = 1, 2, 3$, designing yields higher than the average, equal to the average, and lower than the average, respectively.
New variables w_{is} and y_{is}.

The farmer problem: extended form

We now want to maximize the expected profit. Assuming that the 3 scenarios are equiprobable, we can form the new program

$$
\begin{aligned}
\min\ & 150x_1 + 230x_2 + 260x_3 + \\
& + \sum_{s=1}^{3} \frac{1}{3}(238y_{1s} - 170w_{1s} + 210y_{2s} - 150w_{2s} - 36w_{3s} - 10w_{4s}) \\
\text{t.q. } & x_1 + x_2 + x_3 \leq 500; \\
& 3x_1 + y_{11} - w_{11} \geq 200; 2.5x_1 + y_{12} - w_{12} \geq 200; 2x_1 + y_{13} - w_{13} \geq 200; \\
& 3.6x_2 + y_{21} - w_{21} \geq 240; 3x_2 + y_{22} - w_{22} \geq 240; \\
& \quad 2.4x_2 + y_{23} - w_{23} \geq 240; \\
& w_{31} + w_{41} \leq 24x_3; w_{31} + w_{41} \leq 20x_3; w_{31} + w_{41} \leq 16x_3; \\
& w_{31} \leq 6000; w_{32} \leq 6000; w_{33} \leq 6000; \\
& x, y, w \geq 0.
\end{aligned}
$$

$\rightarrow$ extended form.

The farmer problem: stages

The seeding decisions are called first-stage decisions, while the sale and purchase decisions are called second-stage decisions.

Total profit: \$108390.

	Culture	Wheat	Corn	Sugar beets
First stage	Surface (acres)	170	80	250
$s = 1$	Productions (T)	510	288	6000
	Sales (T)	310	48	6000
	Purchases (T)	-	-	-
$s = 2$	Productions (T)	425	240	5000
	Sales (T)	225	-	5000
	Purchases (T)	-	-	-
$s = 3$	Productions (T)	340	192	4000
	Sales (T)	140	-	4000
	Purchases (T)	-	48	-

Observations

The optimal decision has changed!!!

Decision under perfect information: if the farmer could know the scenario in advance, or wait to observe the realization of the random variables (wait-and-see approach), the average annual profit would be $115406. The difference with the optimal decision under uncertainty is called expected value of perfect information: profit loss due to uncertainty.

On the opposite, if the farmer only uses the average information, i.e. he replaces the random variables by their expectations, the average profit would be $107240 (expected value solution), leading to a loss of $1150 with respect to the solution of the stochastic problem. This difference is known as value of the stochastic solution.

Formulation with recourse

More generally, we consider the (linear) program

$$\begin{aligned} \min\ & c^T x + E_\omega[q^T y] \\ \text{t.q. } & Ax = b, \\ & T(\omega)x + Wy(\omega) = h(\omega),\ \forall \omega \in \Omega \\ & x \in X, \\ & y \in Y,\ \forall \omega. \end{aligned}$$

Fixed recourse: W does not change with the scenario.

How to decide over y?

Some definitions

$$\min_{x \in X | Ax = b} \left\{ c^T + E_\omega \left[\min_{y \in Y} q^T y | Wy = h(\omega) - T(\omega)x \right] \right\}.$$

- Second stage function, or recourse function $v : \mathbb{R}^m \to \mathbb{R}$:

$$v(z) \stackrel{def}{=} \min_{y \in Y} \{q^T y | Wy = z\};$$

- Expected value function, or recourse of minimum expectation $\mathcal{Q} : \mathbb{R}^n \to \mathbb{R}$:

$$Q(x) = E_\omega[v(h(\omega) - T(\omega)x)].$$

It describes the expected recourse cost, for any firs-stage decision $x \in \mathbb{R}^n$.

The two-stage (linear) stochastic program

One can reformulate our optimization problem as

$$\min_{x \in X} \{c^T x + \mathcal{Q}(x) \mid Ax = b\}.$$

It is a (nonlinear) optimization problem in $\mathbb{R}^n$.

In terms of y's:

$$\begin{aligned}
\min_{x,\, y(\omega)} \ & E_{\omega}[c^T x + q^T y(\omega)] & \\
\text{s.t. } & Ax = b & \text{first-stage constraints} \\
& T(\omega)x + Wy(\omega) = h(\omega),\ \forall \omega \in \Omega & \text{second-stage constraints} \\
& x \in X,\ y(\omega) \in Y.
\end{aligned}$$

Consider the (discrete) case where $\Omega = \{\omega_1,\ \omega_2, \ldots, \omega_S\} \subset \mathbb{R}^r$.

$$P(\omega = \omega_s) = p_s,\ s = 1, 2, \ldots ...., S$$
$$T_s = T(\omega),\ h_s = h(\omega)$$

Deterministic equivalent

$$
\begin{aligned}
\min_{x,y_1,\ldots,y_S} \quad & c^T x + p_1 q^T y_1 + p_2 q^T y_2 + \ldots p_S q^T y_S \\
\text{t.q.} \quad & \\
& \begin{array}{lllll}
Ax & & & & = b \\
T_1 x & +W y_1 & & & = h_1 \\
T_2 x & & +W y_2 & & = h_2 \\
\vdots & & & \ddots & \\
T_S x & & & +W y_s & = h_s
\end{array} \\
& x \in X,\ y_1 \in Y,\ y_2 \in Y, \ldots, y_s \in Y.
\end{aligned}
$$

Deterministic equivalent (II)

- $y_s = y(\omega_s)$ is the recourse action to take if the scenario ω_s occurs.
- Advantage: it is a linear program.
- Disadvantage: it is a linear program of (very) high dimension:
  - $n + pS$ variables;
  - $m_1 + mS$ constraints.
- Advantage: the constraints matrix has a staircase structure.
  It is possible to exploit it (L-Shaped algorithm).

General principle: the nonlinear term in the objective, that is the recourse function $\mathcal{Q}(x)$, requires to solve all the linear second-stage programs.

Is it possible to avoid the repeated second-stage functions evaluations?

Idea: build a master problem in x, but compute the complete objective function (involving first- and second-stage decision) only as a subproblem.

Multistage stochastic programming

Similar to dynamic programming: one has to take a sequence of decisions.

Consider the general problem

$$\inf_{x \in \mathcal{N}} E\left[\sum_{t=0}^{T} f_{t+1}(\xi, x^t(\xi), x_{t+1}(\xi))\right],$$

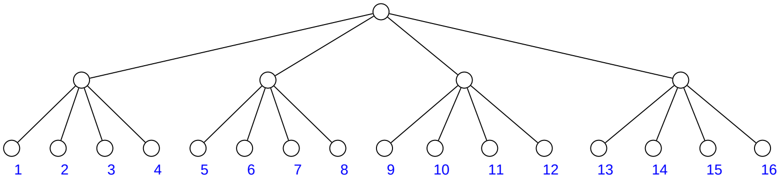
$$\text{t.q. } x_t(\xi) \in X_t(\xi) \text{p.s.},$$

where x_t is the decision at stage t, and x^t is the history of previous decisions, i.e. $x^t = \{x_1, \dots, x_t\}$.

As in the two-stage case, it is possible to form the extended form is the support of the random vector at each stage is finite. But the program quickly becomes too big!

Scenarios tree

Conceptually, the sequence of random events realizations can be organized in a tree. There are as many leaves as scenarios.



The scenarios are not necessarily equiprobable, the random terms can be correlated,. . .

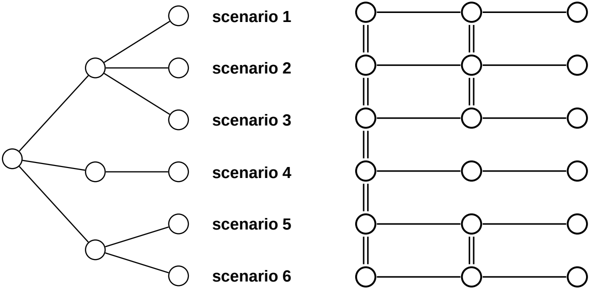
Nodal approach

- We create a vector of variables at each node of the tree.
- This vector corresponds to what should be our decision given the (known) realization of the random vectors at the previous stages.
- Let index the nodes by $l = 1, 2, \ldots, \mathcal{L}$. We have to know the parent of each node.
- A scenario corresponds to a path from the root to a leaf of the scenarios tree.

Nonanticipativity: if at some node two scenarios are identical, they should carry the same decision vector. Implicitly enforced using the nodal approach.

Scenarios approach

Instead of working on the complete problem, it is tempting to decompose it by scenarios.



Nonanticipativity

Let S_s^t be the set of scenarios that are identical at stage t. We have to ensure that

$$x_{its} = x_{its'}, \ \forall i \in N, \ \forall t \in T, \ \forall s \in S, \ \forall s' \in S_s^t.$$

In other terms, the scenarios should be associated to the same decisions as long as their history is the same. From the time they diverge, the corresponding decisions can be different.

Idea: explicitly introduce the nonanticpativity constraints, but put them in the objective rather them in the constraints.

Markovian decision process over finite horizon

At stage k, one observes the state x_k and takes a decision $u_k \in U_k(x_k)$.
Then a random variable ω_k is generated from a probability law $\mathbb{P}_k(\cdot \mid x_k, u_k)$ that can depend on (k, x_k, u_k). One assumes that the previous values $\{(x_n, u_n, \omega_n), n < k\}$ do not deliver additional information on $\mathbb{P}_k$ when one knows (k, x_k, u_k).

One observes ω_k, pays a cost $g_k(x_k, u_k, \omega_k)$, and the state at the next stage is $x_{k+1} = f_k(x_k, u_k, \omega_k)$. Total (random) cost:

$$g_N(x_N) + \sum_{k=0}^{N-1} g_k(x_k, u_k, \omega_k).$$

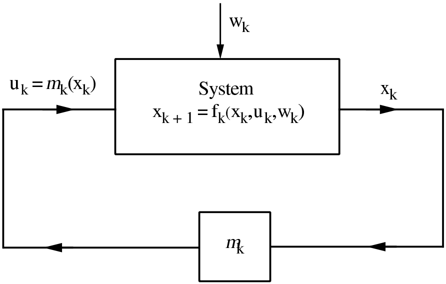
A stochastic dynamic programming model (II)

An admissible policy is a suite of functions $\pi = (\mu_0, \ldots, \mu_{N-1})$ such that $\mu_k : X_k \to U_k$ and $\mu_k(x) \in U_k(x)$ for all $x \in X_k$, $0 \leq k \leq N-1$

At stage k, one has:

- X_k = state space;
- $U_k(x)$ = set of admissible decisions in state x;
- D_k = perturbations ω_k space;
- g_k = cost function;
- f_k = transition function;
- x_k = system stage at stage k;
- u_k = decision at stage k.
- ω_k = perturbation (random variable) occurring at stage k.

A stochastic dynamic programming model (III)



Cost

For $0 \leq k \leq N+1$ and $x \in X_k$, let

$$
\begin{aligned}
J_{\pi,k}(x) &= \text{total expected cost from stage } k \text{ to the end, given} \\
&\quad \text{state } x \text{ at stage } k \text{ and under policy } \pi \\
&= \mathbb{E}_{\pi,x}\left[g_N(x_N) + \sum_{n=k}^{N-1} g_n(x_n, u_n, \omega_n) \right]
\end{aligned}
$$

where $\mathbb{E}_{\pi,x}$ indicates the expectation when $x_k = x$, $u_n = \mu_n(x_n)$ and $x_{n+1} = f_n(x_n, u_n, \omega_n)$ for $n = k, \ldots, N-1$.

Given a policy π, one has the recurrence equation

$$
\begin{aligned}
J_{\pi,N}(x) &= g_N(x) \quad \text{for all } x \in X_N \\
J_{\pi,k}(x) &= \mathbb{E}_{\pi,x}\left[g_k(x, \mu_k(x), \omega_k) + J_{\pi,k+1}(f_k(x, \mu_k(x), \omega_k)) \right] \\
&\quad \text{for } 0 \leq k \leq N,\ x \in X_k.
\end{aligned}
$$

where the expectation is taken with respect to ω_k that follows the law $\mathbb{P}_k(\cdot \mid x, \mu_k(x))$.

We are looking for a policy π that minimizes $J_{\pi,0}(x_0)$, the mathematical expectation of the sum of costs from stage 0 to stage N.

Let $\pi^* = (\mu_0^*, \mu_1^*, \ldots, \mu_{N-1}^*)$ such an optimal policy. Let

$$\begin{aligned} J_k^*(x) &= \text{total optimal expected cost from stage } k \text{ to the end,} \\ &\quad \text{given the state } x \text{ at step } k \\ &= \min_{\pi} J_{\pi,k}(x) \\ &= \min_{\mu_k,\ldots,\mu_{N-1}} J_{\mu_k,\ldots,\mu_{N-1},k}(x). \end{aligned}$$

Proposition.

(A) One has $J_k^* \equiv J_k$, where the J_k's are defined by recurrence equations (or equations of dynamic programming):

$$\begin{aligned} J_N(x) &= g_N(x) \quad \forall x \in X_N \\ J_k(x) &= \min_{u \in U_k(x)} \mathbb{E}\left[g_k(x,u,\omega_k) + J_{k+1}(f_k(x,u,\omega_k))\right] \\ &\qquad \text{for } 0 \leq k \leq N-1,\ x \in X_k, \end{aligned}$$

where expectation $\mathbb{E}$ is taken with respect to ω_k, following the law $\mathbb{P}_k(\cdot \mid x,u)$.

(B) A value of u that allows to attain the infimum is an optimal decision to take when one is in state x at stage k. One can define an optimal policy (if it exists) by

$$\mu_k^*(x) = \arg\min_{u \in U_k(x)} \mathbb{E}\left[g_k(x,u,\omega_k) + J_{k+1}(f_k(x,u,\omega_k))\right].$$

One then has $J_k \equiv J_{\pi^*,k}$ for all k.

$$
\begin{aligned}
J_k^*(x_k) & \\
&= \min_{(\mu_k,\pi^{k+1})} \mathbb{E}_{\pi^k,x}\left[g_k(x_k,\mu_k(x_k),\omega_k) + g_N(x_N) + \sum_{i=k+1}^{N-1} g_i(x_i,\mu_i(x_i),\omega_i)\right] \\
&= \min_{\mu_k} \mathbb{E}_{\pi^k,x_k}\Bigg(g_k(x_k,\mu_k(x_k),\omega_k) \\
&\qquad + \min_{\pi^{k+1}}\left[\mathbb{E}_{\pi^{k+1},x_{k+1}}\left[g_N(x_N) + \sum_{i=k+1}^{N-1} g_i(x_i,\mu_i(x_i),\omega_i) \mid \omega_k\right]\right]\Bigg) \\
&= \min_{\mu_k} \mathbb{E}_{\pi^k,x_k}\left(g_k(x_k,\mu_k(x_k),\omega_k) + J_{k+1}^*(f_k(x_k,\mu_k(x_k),\omega_k))\right) \\
&= \min_{u_k\in U_k(x_k)} \mathbb{E}_{\pi^k,x_k}\left(g_k(x_k,u_k,\omega_k) + J_{k+1}^*(f_k(x_k,u_k,\omega_k))\right) \\
&= J_k(x_k).
\end{aligned}
$$

Backward chaining procedure

For all $x \in X_N$, $J_N(x) \leftarrow g_N(x)$;
For $k = N-1, \ldots, 0$ do
for all $x \in X_k$ do

$$J_k(x) \leftarrow \min_{u \in U_k(x)} \mathbb{E}\left[g_k(x,u,\omega_k) + J_{k+1}(f_k(x,u,\omega_k))\right];$$
$$\mu_k^*(x) \leftarrow \arg\min_{u \in U_k(x)} \mathbb{E}\left[g_k(x,u,\omega_k) + J_{k+1}(f_k(x,u,\omega_k))\right];$$

Bellman optimality principle (probabilistic case)

If $\pi^* = (\mu_0^*, \dots, \mu_{N-1}^*)$ is an optimal policy for the initial problem and if $0 < k < N$, then the truncated policy $\pi_k^* = (\mu_k^*, \dots, \mu_{N-1}^*)$ is an optimal policy for the subproblem

$$\min_{u_k, \dots, u_{N-1}} \mathbb{E}\left[g_N(x_N) + \sum_{n=k}^{N-1} g_n(x_n, u_n, \omega_n) \mid x_k \right].$$

Assumptions: discrete time, Markovian model, additive costs.

The principle does not hold for the subproblem

$$\min_{u_k, \dots, u_j} \mathbb{E}\left[\sum_{n=k}^{j} g_n(x_n, u_n, \omega_n) \mid x_k \right]$$

if $j < N$ and x_j is not determined.

Open vs closed loop

Closed loop control: one takes each decision the latest possible, when one has the maximum of information.
Open loop control: all the decisions $u_0, \ldots, u_{N-1}$ are taken from the start.

The difference between the expected costs is the value of additional information.

This model of SDP has a lot of possible generalisations:
— introduction of an actualisation factor;
— infinite horizon;
— infinite states or decisions space;
— continuous time;
— etc.

Markovian reformulation

If f_k, g_k or the law of ω_k depends on the previous states x_k or decisions u_k, one usually can redefine the states, decisions, and perturbations space to come back to the Markovian model. One has to incorporate enough information in state x_k ("state augmentation").

References

- John R. Birge and François Louveaux, *Introduction to Stochastic Programming*, 2^{nd} edition, Springer-Verlag, 2011.
- Alexander Shapiro, Darinka Dentcheva and Andrzej Ruszczynski, *Lectures on Stochastic Programming: Modeling and Theory*, SIAM, 2009.
- Stein W. Wallace and William T. Ziemba (editors), *Applications of stochastic programming*, SIAM, 2005.
- Andrzej Ruszczynski and Alexander Shapiro (editors), *Stochastic Programming*, Elsevier, 2003.
- Peter Kall and Stein W. Wallace, *Stochastic Programming*, John Wiley & Sons, 1994.

Stochastic programming society: `http://www.stoprog.org`