Stochastic programming
Multi-stage problems

Fabian Bastin
bastin@iro.umontreal.ca
Université de Montréal – CIRRELT – IVADO – Fin-ML

Motivation: example

- Assume that we invest in the market. We would to reach a nominal goal G after Y years.

- The portfolio is rebalanced each v years, so there are $T = Y/v$ times where we have to decide what are the investments to do.
  There are T stages in our problem.

- We face a space of N investment decisions (i.e. we have N possible actions); we have a set $\mathcal{T} = \{1, 2, \ldots, T\}$ of investment periods.

- If we go over the target G, we will have an interest rate q on the surplus.

- If we do not meet the target, we have to borrow the difference at an interest r.

- The initial wealth is b.

Variables

- x_{it}, $i \in N$, $t \in \mathcal{T}$: amount of money to invest in action i during the period t;
- y: excess amount at the end of the horizon;
- w: lack of money at the end of the horizon;
- In the current framework, we aim to take the corrective actions in order to reach the target of G dollars.

Deterministic formulation

$$
\begin{aligned}
\max\ & qy - rw \\
\text{s.t.}\ & \sum_{i \in N} x_{i1} = b, \\
& \sum_{i \in N} \omega_{it} x_{i,t-1} = \sum_{i \in N} x_{it}, \quad \forall t \in \mathcal{T} \backslash 1, \\
& \sum_{i \in N} \omega_{iT} x_{iT} - y + w = G, \\
& x_{it} \geq 0, \quad \forall i \in N,\ t \in \mathcal{T}, \\
& y, w \geq 0.
\end{aligned}
$$

Random returns

- The investment returns can be viewed as random variables, as we do not know their realizations in advance.
- Imagine that for each action, at each time t, the are R possible realizations.
- The scenarios are made of all the possible sequences of realizations.

Example: assume $R = 4$ and $T = 3$. Scenarios:

| $t = 1$	$t = 2$	$t = 3$
1	1	1
1	1	2
1	1	3
1	1	4
1	2	1
$\vdots$	$\vdots$	$\vdots$
4	4	4

Stochastic program formulation

- Let x_{its}, $t \in \mathcal{T}$, $s \in S$ be the amount to invest in action i during the period t in scenario s;
- y_s: excess amount at the end of the horizon for scenario s;
- w_s: lack of money at the end of the horizon for scenario s.
- Let ω_{it} be the random return; it can be seen as function of the scenario s.

Stochastic version

$$\max \ \sum_{s=1}^{S} p_s(qy_s - rw_s)$$

$$\text{s.t.} \ \sum_{i \in N} x_{i1} = b,$$

$$\sum_{i \in N} \omega_{its} x_{i,t-1,s} = \sum_{i \in N} x_{its}, \quad \forall t \in \mathcal{T} \backslash 1, \forall s \in S$$

$$\sum_{i \in N} \omega_{iTs} x_{iT} - y_s + w_s = G, \quad \forall s \in S,$$

$$x_{its} \geq 0, \quad \forall i \in N, \ t \in \mathcal{T}, \forall s \in S,$$

$$y_s, w_s \geq 0 \quad \forall s \in S.$$

Scenario tree

Conceptually, the sequence of events can be arranged in a tree.
There is one leaf per scenario.
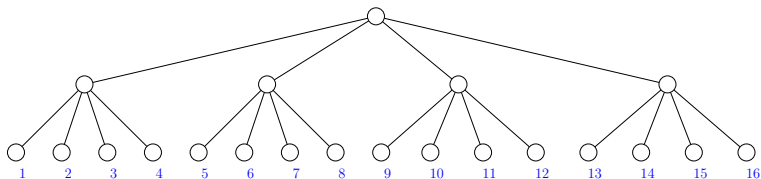
The scenarios can have different probabilities, the variables can
be correlated,...

Non-anticipativity

- The conversion between the deterministic model and the stochastic version is easy...
- but the built model is wrong!!!
- We have to enforce the non-anticipativity:

Let S_s^t be the set of scenarios that are identical to scenario s at time t. We have to guarantee that

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

Equivalently,

$$\left(\sum_{s' \in S_s^t} p_s'\right) x_{its} = \sum_{s' \in S_s^t} p_s' x_{its'} \quad \forall\, i, t, s,$$

or

$$x_{its} = \frac{\sum_{s' \in S_s^t} p_s' x_{its'}}{\sum_{s' \in S_s^t} p_s'} \quad \forall\, i, t, s$$

$$= E[x_{its'} | s' \in S_s^t] \quad \forall\, i, t, s.$$

The amount invested in scenario s, multiplied by the probability to reach a scenario equivalent to s, has to be equal to the expected amount that would be invested in any scenario equivalent to s.

There are many formulations of these conditions!

Back to the scenario tree

- We have to enforce nonanticipativity as we have created copies of the variables for each scenario, at each period. Graphically:
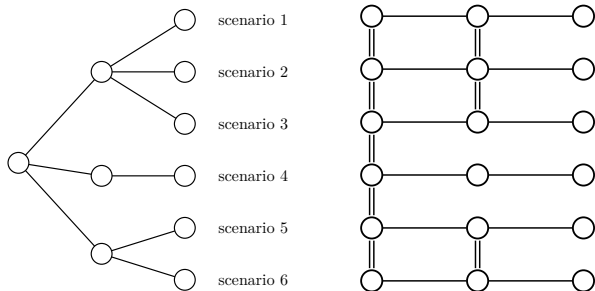
Nodal approach

- We can also create a vector of variables for each node in the tree.
- This vector corresponds to what should be our decision given the known realizations of the random variables upon this stage (i.e. at this node).
- Index the nodes by $l = 1, 2, \ldots, \mathcal{L}$.
- We have to know the ancestor of each node. Let $A(l)$ the ancestor of node $l \in \mathcal{L}$ in the scenarios tree.

Reformulation

$$\max \sum_{s \in S} p_s(qy_s - rw_s)$$

$$\text{s.t.} \sum_{i \in N} x_{i1} = b,$$

$$\sum_{i \in N} \omega_{il} x_{i,A(l)} = \sum_{i \in N} x_{il}, \quad \forall l \in \mathcal{L} \backslash 1,$$

$$\sum_{i \in N} w_{iA(s)} x_{iA(s)} - y_s + w_s = G, \quad \forall s \in S,$$

$$x_{il} \geq 0, \quad \forall i \in N, \ \forall l \in \mathcal{L},$$

$$y_s, w_s \geq 0 \quad \forall s \in S.$$

Multi-period production planification

- A firm produces various different goods.
- Ressources (for instances manual and machine work hours) are required. We assume that these requirements are known for each product.
- The demand must be met at the end of each period, but this demand is not perfectly known.
- A too large or too small inventory induces costs (storage, additional purchases to satisfy the demand,…).
- It is possible to add machine and work hours, within predefined limits.
- There is a hiring and firing cost related to changes in the workforce.

Decision problem

- We have to decide now on
    - the quantity of each product to manufacture during each period;
    - the additional capacity to use during each period;
    - to hire or to fire staff during each period;
- A random demand occurs. Conceptually, this happens at each period.
- After having observed the random demands, we can decide on how to stock the products in the inventory, or to buy from an external source.

Definitions

1. Sets:
   - T: number of periods, also seen as a set;
   - N: products;
   - M: ressources.

2. Variables:
   - x_{jt}: production of product $j \in N$ during the period $t \in T$;
   - u_{it}: additional amount of ressource $i \in M$ to buy during the period $t \in T$;
   - $z_{t-1,t}^{+}$, $z_{t-1,t}^{-}$: planned augmentation or reduction of the labor force;
   - $y_{j,t}^{+}$, $y_{j,t}^{-}$: surplus, deficit of product $j \in N$ at the end of the period $t \in T$.

Parameters

- All the above variables have associated costs (α, β, γ, δ).
- ω_{jt}: demand of product $j \in N$ during the period $t \in T$.
- U_{it}: upper bound on u_{it}.
- a_{ij}: amount of ressource $i \in M$ required to produce one unit of the good $j \in N$.
- b_{it}: amount of ressource $i \in M$ available at time $t \in T$.

Model

$$\min \sum_{j \in N} \sum_{t \in T} \alpha_{jt} x_{jt} + \sum_{i \in M} \sum_{t \in T} \beta_{it} u_{it} + \sum_{t \in T \backslash 1} \left(\gamma_{t-1,t}^{+} z_{t-1,t}^{+} + \gamma_{t-1,t}^{-} z_{t-1,t}^{-} \right)$$
$$+ E_{\boldsymbol{\omega}} \left[\min \left(\sum_{j \in N} \sum_{t \in T} (\delta_{jt}^{+} y_{jt}^{+} + \delta_{jt}^{-} y_{jt}^{-}) \right) \right],$$

such that

$$\sum_{j \in N} a_{ij} x_{jt} \leq b_{it} + u_{it} \quad \forall i \in M, \ \forall t \in T,$$

$$u_{it} \leq U_{it} \quad \forall i \in M, \ \forall t \in T,$$

$$z_{t-1,t}^{+} - z_{t-1,t}^{-} = \sum_{j \in N} a_{jt} (x_{jt} - x_{j,t-1}) \quad \forall t \in T \backslash 1,$$

$$x_{jt} + y_{j,t-1}^{+} - y_{jt}^{+} + y_{jt}^{-} = \omega_{jt} \quad \forall j \in N, \ \forall t \in T \text{ (demand satisfaction)},$$

Extended form of the recourse model

$$\min \sum_{j \in N} \sum_{t \in T} \alpha_{jt} x_{jt} + \sum_{i \in M} \sum_{t \in T} \beta_{it} u_{it} + \sum_{t \in T \setminus 1} \left(\gamma_{t-1,t}^{+} z_{t-1,t}^{+} + \gamma_{t-1,t}^{-} z_{t-1,t}^{-} \right)$$
$$+ \sum_{s \in S} \sum_{j \in N} \sum_{t \in T} p_s (\delta_{jt}^{+} y_{jts}^{+} + \delta_{jt}^{-} y_{jts}^{-}),$$

such that

$$\sum_{j \in N} a_{ij} x_{jt} \le b_{it} + u_{it} \quad \forall i \in M, \ \forall t \in T,$$

$$u_{it} \le U_{it} \quad \forall i \in M, \ \forall t \in T,$$

$$z_{t-1,t}^{+} - z_{t-1,t}^{-} = \sum_{j \in N} a_{jt} (x_{jt} - x_{j,t-1}) \quad \forall t \in T \setminus 1,$$

$$x_{jt} + y_{j,t-1,s}^{+} - y_{jts}^{+} + y_{jts}^{-} = \omega_{jts} \quad \forall j \in N, \ \forall t \in T,$$

nonanticipativity constraints.

Potential solution techniques

Adapted from Jalal Kazempour,
https://www.jalalkazempour.com/teaching

- Direct solution: computational issues
- Linear decision rules: a non-decomposition approach with approximation. Assumption of linear dependency between decision variables and uncertain parameters. Using this approximation, the number of constraints and variables grows only linearly with the number of stages.
- Nested Benders decomposition: a decomposition approach based on fixing complicating variables.
- Lagrangian relaxation: a decomposition approach based on relaxing the complicating (non-anticipativity) constraints.

A nested Benders decomposition

We transform a multi-stage problem to a nested series of two-stage problems

Similar to Benders decomposition, nested Benders decomposition is an iterative solution technique. However, in each iteration, we need to solve several two-stage stochastic problems.
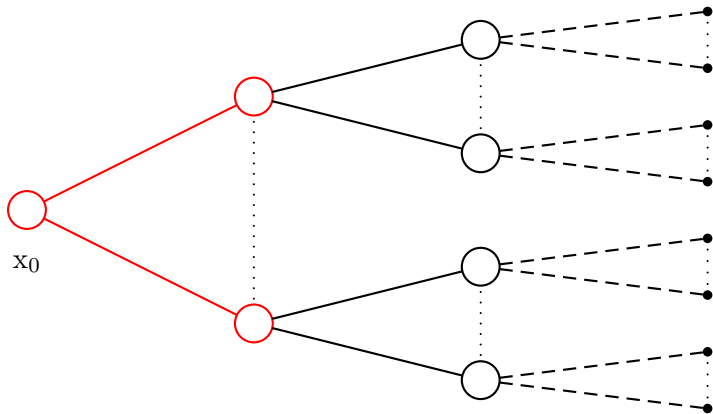
Illustration
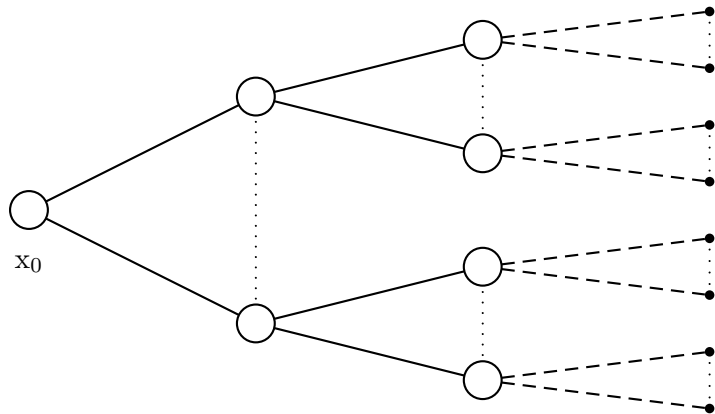
First stage   Second stage   Third stage   ...

Step 1: forward pass

First stage   Second stage   Third stage   ...

Step 1: forward pass

Solve

$$\min_{x_0, \theta_0} \ c_0^{\mathrm{T}} x_0 + \theta_0$$
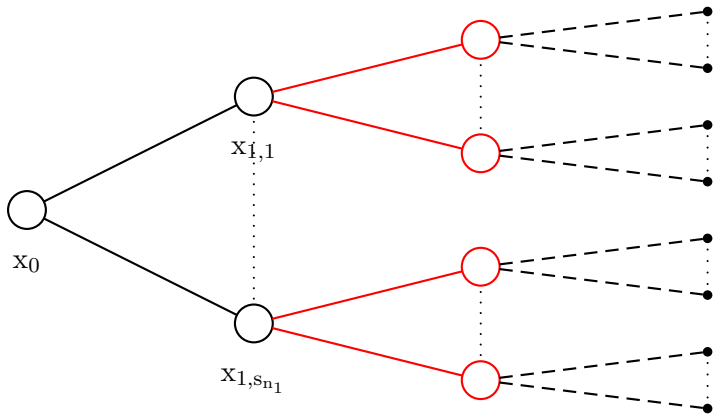
$$\text{s.t. } A_0 x_0 = b_0$$

$$\theta_0 \geq \theta_{0,\mathsf{min}}$$

feasibility and optimality cuts

$$x_0 \geq 0.$$

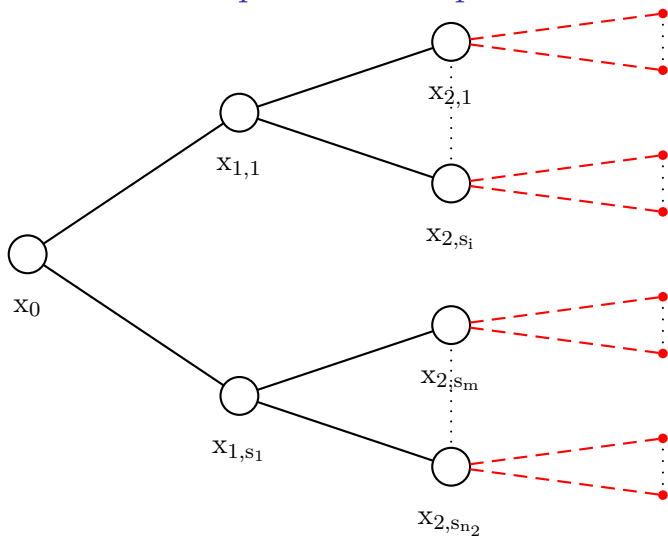# Step 2: forward pass

First stage   Second stage   Third stage   ...

Solve

$$\min_{x_{1,i}, \theta_{1,i}} \ c_{1,i}^{T} x_{1,i} + \theta_{1,i}$$

$$\text{s.t.} \ A_{1,i} x_{1,i} = b_{1,i}$$

$$W_{1,i} x_{1,i} = h_{1,i} - T_{1,i} x_0$$

$$\theta_{1,i} \geq \theta_{1,i,\text{min}}$$

feasibility and optimality cuts

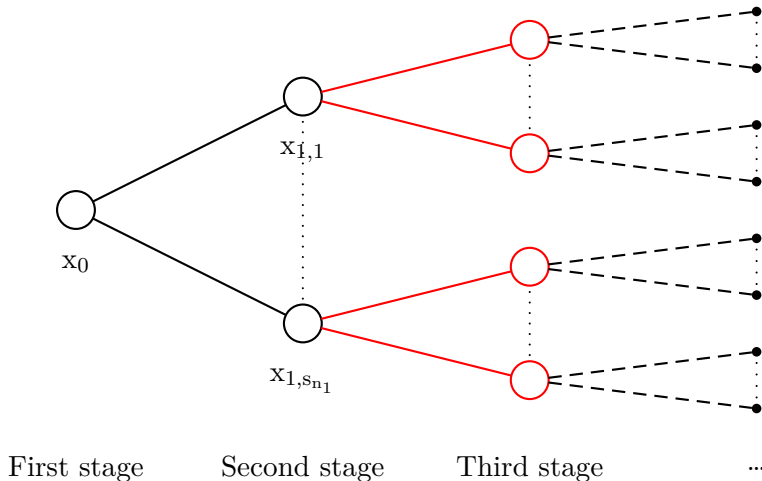$$x_{1,i} \geq 0.$$

Step 3: forward pass

First stage   Second stage   Third stage

Step 2: backward pass

The newly formed cuts are added to the Step 2 program and we solve again the program to determine cuts on the second-stage program.

Step 2: backward pass

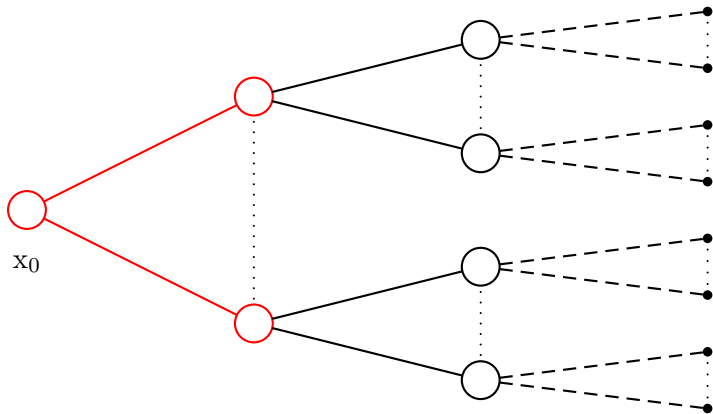
First stage   Second stage   Third stage   ...

Step 1: backward pass

The newly formed cuts are added to the Step 1 program.

This forms a new master program for the first stage, that is solved again in the forward pass.

Step 1: backward pass

First stage   Second stage   Third stage   ...

Scenario tree

The main issue is the construction of the scenario tree that is growing exponentially at each stage, and so the number of subproblems.

The stochastic dual dynamic programming algorithm aims to solve the problem on a lattice rather than a tree, under the assumption that the underlying random process is Markovian.

The augmented Lagrangian approaches will still rely on the scenario tree, but will consider the relaxation on the nonanticipativity constraints rather than solve a succession a two-stage problems. This limit the number of subproblems to the number of scenarios and can handle non-Markovian processes. The drawback is that it can take many iterations to restore the nonanticativity.