Scenario Generation and Sampling Methods

Güzin Bayraksan
Tito Homem-de-Mello

SVAN 2016 – IMPA
May 13th, 2016

Deterministic methods for scenario generation

So far we have discussed (mostly) approaches based on random sampling to generate scenarios for the problem $\min_{x \in X} \mathbb{E}[G(x, \xi)]$.

We can do a lot with that (convergence results, confidence intervals, etc.).

BUT: This may not always be the best choice! Some possible reasons:

- Random sampling does not use any information about the problem, it only uses information about the distribution of ξ.
- Suppose the objective function is a *risk function* instead of the expectation. Then, it is quite possible that only the ξ's in some region actually 'matter".
- In the case of multi-stage problems, we cannot simply do SAA by drawing *sample paths* from the underlying stochastic process.

A stability-based approach

IDEA: Let P denote the distribution of ξ, which takes values in Ξ. Define the function

$$v(P) := \min_{x \in X} \mathbb{E}_P[G(x, \xi)].$$

and, for $\rho > 0$, define the set $X^\rho := X \cap \rho\mathbb{B}$.

Let $d_{G,\rho}(P, Q)$ denote the distance between two distributions P and Q, defined as

$$d_{G,\rho}(P, Q) := \sup_{x \in X^\rho} \left| \mathbb{E}_P[G(x, \xi)] - \mathbb{E}_Q[G(x, \tilde{\xi})] \right|$$

where $\tilde{\xi} \sim Q$.

Then, it is possible to show that there exist constants $\rho > 0$ and $\varepsilon > 0$ such that

$$|v(P) - v(Q)| \leq d_{G,\rho}(P, Q) \quad \text{whenever } d_{G,\rho}(P, Q) < \varepsilon.$$

Why is this useful?

Let P be the original distribution and Q be a discretization. So, if we can choose Q to approximate P well (in terms of $d_{G,\rho}(P, Q)$), then the optimal values will be automatically close!

How to do that?

- Computing $d_{G,\rho}(P, Q)$ seems even harder than the original problem!
- Alternatively, if we can find a more easily computable distance d such that $d_{G,\rho}(P, Q) \leq d(P, Q)$, then all we need is to find Q that makes $d(P, Q)$ small.

A class of distances

A very large class of distances between probability measures has the form

$$d_{\mathcal{H}}(P, Q) := \sup_{h \in \mathcal{H}} \left| \int_{\Xi} h(s)\, P(ds) - \int_{\Xi} h(s)\, Q(ds) \right|,$$

where $\mathcal{H}$ is a family of integrable functions with certain properties. Examples:

- $\mathcal{H}_B = \{\text{bounded measurable functions}\}$, which yields

  $$d_{\mathcal{H}_B}(P, Q) = \sup\{|P(A) - Q(A)| : A \text{ is a measurable set.}\}$$

  This is called the *variational distance* between P and Q.

- $\mathcal{H}_{BL} = \{\text{bounded Lipschitz functions with Lipschitz constant 1}\}$, i.e., $|h(s) - h(t)| \leq \|s - t\|$ for all s, t.

A class of distances

Some further important examples:

- $\mathcal{H}_L = \{\text{Lipschitz functions with Lipschitz constant } 1\}$. This is called the *Kantorovich distance*. On the real line, the corresponding distance can be written as

$$d_K(P,Q) \;=\; \int_{\mathbb{R}} \left|F_P(z) - F_Q(z)\right| dz \;=\; \int_0^1 \left|F_P^{-1}(u) - F_Q^{-1}(u)\right| du$$

  where $F_P(z) = P((-\infty, z]) = \mathbb{P}(\xi \leq z)$.

- $\mathcal{H}_{Lq} = \{\text{Lipschitz functions of order } q \text{ with Lipschitz constant } 1\}$, i,e.

$$|h(s) - h(t)| \leq \|s - t\| \max\{1, \|s\|^{q-1}, \|t\|^{q-1}\} \quad \text{for all } s, t.$$

  This is called the *Fortet-Mourier distance*.

A useful metric d

Consider now the set $\mathcal{H}_c$ defined as

$$\mathcal{H}_c := \left\{h : \left|h(s) - h(t)\right| \leq c(s,t) \quad \forall\, s,t \in \Xi\right\}$$

which yields

$$d_c(P,Q) := \sup_{h\in\mathcal{H}_c} \left|\mathbb{E}_P[h(\xi)] - \mathbb{E}_Q[h(\tilde{\xi})]\right|,$$

where $c(s,t)$ is a function measuring the "distance" between two scenarios s and t in Ξ. One example of requirements on c is that it must satisfy

- $c(s,t) = 0$ if $s = t$;
- $c(s,t) = c(t,s)$;
- $\lim_{\|s-t\|\to 0} c(s,t) = 0$;
- $c(s,t) \leq \lambda(s) + \lambda(t)$ for some function λ which is bounded on bounded sets.

A useful metric d

Suppose that the objective function $G(x,\cdot)$ has the property that

$$|G(x,s) - G(x,t)| \;\leq\; \varphi(\|x\|)\, c(s,t) \quad \forall x \in X,$$

where $\varphi : \mathbb{R}_+ \mapsto \mathbb{R}_+ \setminus \{0\}$ is a non-decreasing function. Then

$$\begin{aligned}
(\varphi(\rho))^{-1} d_{G,\rho}(P,Q) &= (\varphi(\rho))^{-1} \sup_{x \in X^\rho} \left| \mathbb{E}_P[G(x,\xi)] - \mathbb{E}_Q[G(x,\tilde{\xi})] \right| \\
&\leq \sup_{h \in \mathcal{H}_c} \left| \mathbb{E}_P[h(\xi)] - \mathbb{E}_Q[h(\tilde{\xi})] \right| \\
&= d_c(P,Q),
\end{aligned}$$

i.e.,

$$d_{G,\rho}(P,Q) \;\leq\; \varphi(\rho)\, d_c(P,Q).$$

So what?? Calculating d_c still seems pretty hard...

A transportation problem

Gaspard Monge proposed in 1781 the following problem:

> *How to transport soil from site S to site S' with minimum cost, if there is a cost $c(s, s')$ of transporting a grain of sand from position $s \in S$ to position $s' \in S'$?*

We can state the problem in more mathematical terms as follows: let μ be a finite measure on S and ν be a finite measure on S' such that $\mu(S) = \nu(S')$ ($=1$ without loss of generality).

Monge wanted to find a mapping $T : S \mapsto S'$ such that

- $\nu(B) = \mu(T^{-1}(B))$ for any measurable set B (i.e. T preserves volumes);
- T minimizes the total cost

$$\int_S c(s, T(s))\, \mu(ds).$$

A transportation problem

As posed, this is a very difficult problem, which eluded mathematicians for more than 150 years! Why?

- It is hard even to show existence of T
- The optimization problem in nonlinear in T and the feasibility set is non-convex...

The breakthrough

In 1942 L. Kantorovich proposed a solution for a related problem, unaware of Monge's work.

In terms of Monge's problem, Kantorovich's solution can be viewed as a relaxation of the requirement that T be a function (= indivisible grains).

- In Kantorovich's formulation, the transportation map T is replaced by a the wider class of transportation plans, which move masses instead of single points.
- The transportation plans are identified with the set of probability measures π on the product space $X \times Y$.
- Clearly we must have

$$\begin{aligned} \pi(A, Y) &= \mu(A) \quad \text{for any measurable set } A \subseteq S \\ \pi(X, B) &= \nu(B) \quad \text{for any measurable set } B \subseteq S' \end{aligned}$$

The Monge-Kantorovich problem

In other words, Kantorovich's problem can be formulated as

$$\inf_{\pi} \left\{ \int_{S \times S'} c(s,t)\, \pi(ds, dt) \ : \ \pi \text{ has marginals } \mu \text{ and } \nu \right\}.$$

- Note that this is a linear optimization problem over a convex set (the set of all probability measures on $S \times S'$).
- It has a lot more structure — duality, algorithms, etc.
- This is called the Monge-Kantorovich transportation problem in the literature.

The M-K problem: the finite case

Consider the particular setting where

- S has finitely many points $S = \{s_1 \ldots, s_n\}$, so μ corresponds to probabilities $\mu_1 \ldots, \mu_n$;
- S' has finitely many points $S' = \{t_1 \ldots, t_m\}$ so ν corresponds to probabilities $\nu_1 \ldots, \nu_m$.

Then, the M-K problem can be written as the finite dimensional LP

$$\min_{\pi \in \mathbb{R}^{n \times m}} \sum_{i=1}^{n} \sum_{j=1}^{m} c(s_i, t_j)\pi_{ij}$$

$$\sum_{j=1}^{m} \pi_{ij} = \mu_i \quad i = 1, \ldots, n$$

$$\sum_{i=1}^{n} \pi_{ij} = \nu_j \quad j = 1, \ldots, m$$

$$\pi_{ij} \geq 0$$

The Kantorovich-Rubinstein problem

Consider now the situation where $S = S'$. Then, we can write the M-K problem as

$$\inf_{\pi} \int_{S \times S} c(s,t)\, \pi(ds, dt)$$
$$\text{s. to} \quad \pi(A, S) - \pi(S, A) = \sigma(A) \quad \text{for any measurable set } A \subseteq S,$$
$$\pi \geq 0$$

where $\sigma = \mu - \nu$.

This is called the *Kantorovich-Rubinstein problem* in the literature.

A key duality result

By deriving the dual of the K-R problem we obtain the problem

$$\sup_h \int_S h(s)\,\sigma(ds)$$
$$\text{s. to} \quad |h(s) - h(t)| \leq c(s,t) \quad \forall s,t \in S.$$

Have we encountered this problem before?...

YES! It is precisely the Kantorovich distance between distributions P and Q (with respect to scenario distance function c), defined as

$$d_c(P,Q) = \sup_h \Big\{ \Big| \mathbb{E}_P[h(\xi)] - \mathbb{E}_Q[h(\tilde{\xi})] \Big| : |h(s) - h(t)| \leq c(s,t)\ \forall\, s,t \in \Xi \Big\}$$

Consequences of the duality result

Let $d_{\mathrm{MK}}(P, Q)$ be the distance between two distributions P and Q, given by the optimal value of the M-K problem.

- Then, by weak duality we have $d_c(P, Q) \leq d_{\mathrm{MK}}(P, Q)$.

The key question: When is there no duality gap?

- Conditions on c and/or Ξ must be imposed.
- One such condition is that
  (i) Ξ is a compact space
  (ii) c is either continuous on $\Xi \times \Xi$, or it is lower semi-continuous and satisfies the triangle inequality.

Comments

- In the particular case

$$c(\xi^1, \xi^2) = \|\xi^1 - \xi^2\|_m^\eta$$

d_c is called the *Wasserstein distance* between P and Q of order η with respect to the m-norm.

- Consider the case where $\Xi = \{\xi^1, \ldots, \xi^r\}$. Typically P is known, with support $\text{supp}(P) \subseteq \Xi$, and we want to find Q that contains fewer scenarios than P and approximates it as closely as possible.
  - Let $\{z_1, \ldots, z_n\}$ and $\{y_1, \ldots, y_m\}$ denote the support of P and Q respectively.
  - So, we want to solve the M-K problem with variables $\{\pi_{ij}\}$ and $y_1, \ldots, y_m$.

The M-K problem

Suppose we fix the support of the approximating distribution Q (i.e., the $y_1, \ldots, y_m$).

Then, the M-K problem is a simple LP, which can actually be solved analytically:

- For each $j = 1, \ldots, m$, let A_j denote the set of indices of the z_i's that are closer (in terms of the distance measured by c) to y_j than to any other y_k.
- Let

$$\pi^*_{ij} := \begin{cases} p_i & \text{if } i \in A_j \\ 0 & \textit{otherwise} \end{cases}$$

and let $q^*_j := \sum_i \pi_{ij}$.

The M-K problem (cont.)

In general, of course, we don't know the y_j's. Let us restrict supp(Q) to be a subset of supp(P). This is called the scenario reduction problem.

In that case, the resulting problem is combinatorial. Possible approaches are:

- When $|\text{supp}(Q)| = 1$ or $|\text{supp}(Q)| = |\text{supp}(P)| - 1$, the problem is easy to solve.
  - Then, we can either start with an empty set and add once scenario at a time (a *forward* heuristic) until we reach $|\text{supp}(Q)| = m$,
  - Or we can start with $n = |\text{supp}(P)|$ scenarios and eliminate one at a time until we reach $|\text{supp}(Q)| = m$ (a *backward* heuristic).

The facility location problem

Another possibility is to treat the problem as a *facility location problem*.

- However, the problem is still difficult since it involves discrete variables (for the locations) and continuous variables (for the probabilities).
- Moreover, this is a k-dimensional facility location problem, where m is the dimension of the random vector corresponding to the scenarios
- Dempster et al. (2011) propose an iterative scheme that solves alternately for the ys and the πs.

The facility location problem (cont.)

Pflug and Pichler (2015) study the following approach: suppose that $\Xi \subset \mathbb{R}^k$, and let $Y = (y_1, \ldots, y_m)$ denote an array of m points in $\mathbb{R}^k$, which are candidates to constitute $\text{supp}(Q)$.

- Given a scenario $\xi \in \Xi$, define $d(\xi, Y)$ as the distance between ξ and the element of the set $\{y_1, \ldots, y_m\}$ which is the closest to ξ.
- Then, define the function

$$D(Y) \ := \ \mathbb{E}[d(\xi, Y)].$$

  We would like to find the Y that minimizes D.

- It is possible to show that when ξ has continuous distributions the function $D(Y)$ is differentiable, so first-order algorithms can be used to find a stationary point of D.

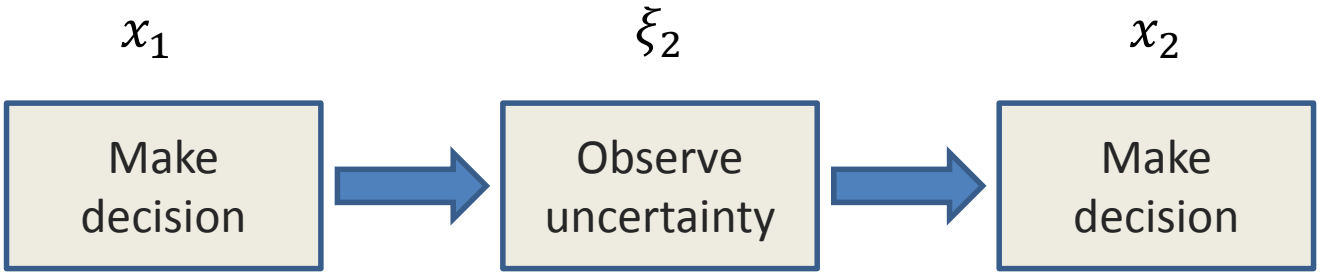
Stochastic dynamic problems

Many problems are dynamic and stochastic by nature. For example,

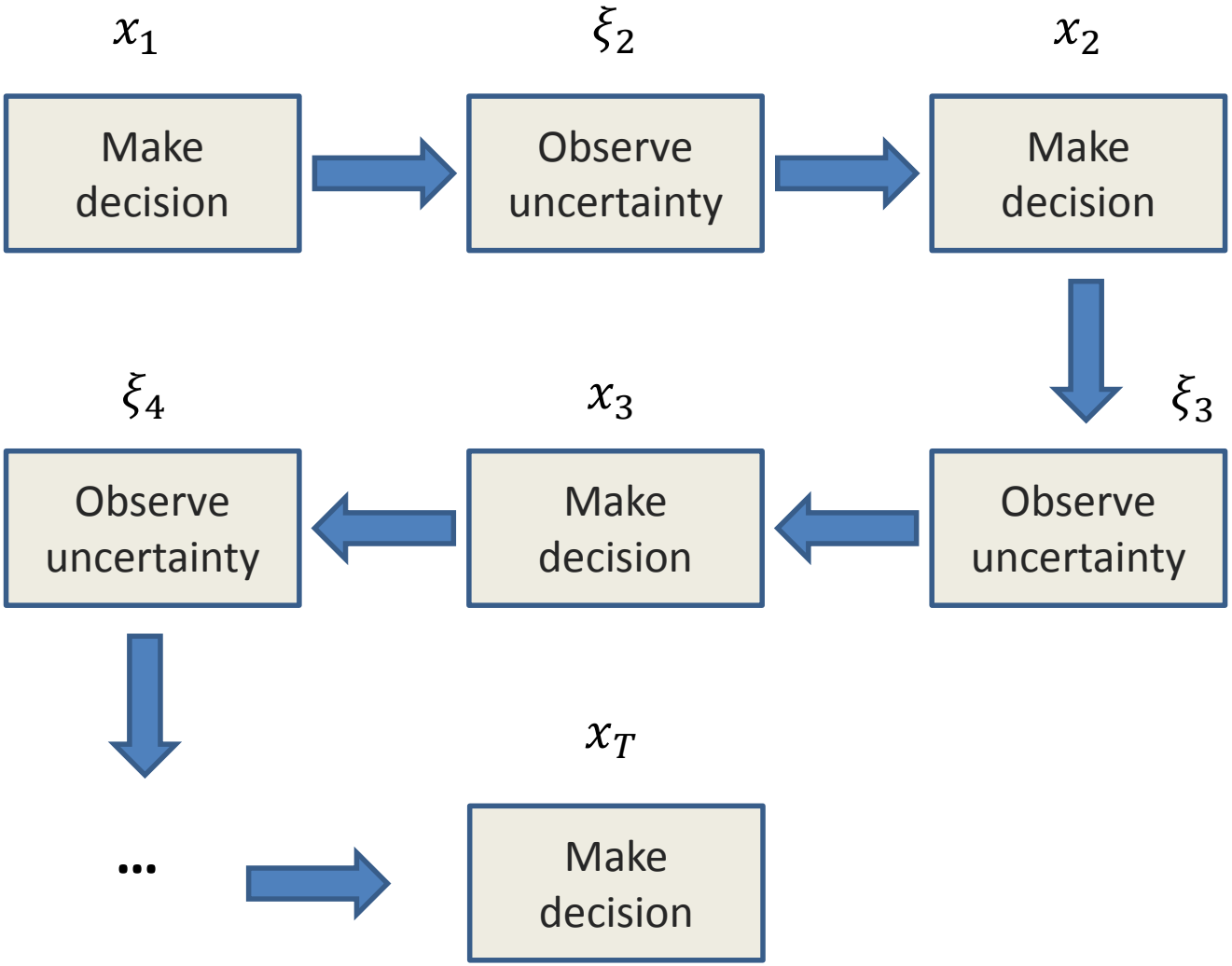
- *Hydroelectric energy planning*: How much energy to produce/store in each month, given that water inflows are uncertain?
- *Portfolio selection*: How much money should I put on each investment every month, knowing that future returns are uncertain?
- *Revenue management*: Which products (e.g., fare classes) should be made available at each time period, given that future demand is uncertain?

Issue: How to generate scenarios for such problems?

Dynamics (two stages)



Dynamics (T stages)



Multi-stage models

$$\begin{aligned} \min_{x_1,\ldots,x_T} \quad & \mathbb{E}\left[f_1(x_1,\xi_1) + \ldots + f_T(x_T,\xi_T)\right] \\ \text{s.t.} \quad & x_t \in \mathcal{X}_t\left(x_1,\ldots,x_{t-1},\xi_1,\ldots,\xi_t\right), \quad t = 1,\ldots,T \\ & x_t \lhd \sigma(\xi_1,\ldots,\xi_t) \end{aligned} \qquad \text{[MSSP]}$$

where

- $x_1,\ldots,x_T$ are the decisions made at each stage;
- ξ_t, $t = 1,\ldots,T$ is the uncertainty observed just before stage t (ξ_1 is a constant), with $\xi_t \in \Xi_t$.
- $f_t(x_t,\xi_t)$ is the cost of decision x_t given the observed uncertainty at that stage;
- $\mathcal{X}_t$ denotes the feasibility set in stage t, which may depend on previous decisions as well as on the observed uncertainty.

Recursive formulation of linear MSSPs

$$\begin{aligned} \min \;\; & c_1^T x_1 + \mathbb{E}_{\xi_2}\left[Q_2(x_1, \xi_2)\right] \\ \text{subject to} \;\; & A_1 x_1 \leq b_1. \end{aligned} \qquad \text{[MSSP-R]}$$

The function Q_2 is defined recursively as

$$\begin{aligned} & Q_t(x_1, \ldots, x_{t-1}, \xi_2, \ldots, \xi_t) = \\ & \quad \min_{x_t} \; c_t^T x_t + \mathbb{E}_{\xi_{t+1}}\left[Q_{t+1}(x_1, \ldots, x_t, \xi_2, \ldots, \xi_{t+1}) \,|\, \xi_2, \ldots, \xi_t\right] \\ & \text{subject to} \quad A_t x_t \leq b_t - \sum_{m=1}^{t-1} B_{m+1} x_m, \end{aligned}$$

$t = 2, \ldots, T$. In the above formulation, ξ_t denotes the random components of c_t, A_t, B_t, b_t. Also, $Q_{T+1} \equiv 0$.

What is different about the multi-stage case?

Can we apply the same ideas seen earlier? (e.g., Monte Carlo, facility location approaches).

- Yes...and no! The point is that it must be done carefully.
- Roughly speaking, the scenario generation technique must somehow incorporate the dependence structure of the input process (trees).

Example: SAA approach

Suppose we look at formulation [MSSP] and apply the SAA approach to it.

- That is, we take N samples of the process $\{\xi_2, \ldots, \xi_T\}$ (call them $\{\xi_2^j, \ldots, \xi_T^j\}$, $j = 1, \ldots, N$) and approximate the expectation with the sample average.
- Does that work? Let us consider for simplicity the case $T = 3$. Then, the actual function Q_2 is

$$Q_2(x_1, \xi_2^j) = \min_{x_2} (c_2^j)^T x_2 + \mathbb{E}_{\xi_3}\left[Q_3(x_1, x_2, \xi_2, \xi_3) \,|\, \xi_2 = \xi_2^j\right]$$
$$\text{subject to} \quad A_2^j x_2 \leq b_2^j - B_2^j x_1,$$

but we are calculating

$$\widehat{Q}_2(x_1, \xi_2^j) = \min_{x_2} (c_2^j)^T x_2 + Q_3(x_1, x_2, \xi_2^j, \xi_3^j)$$
$$\text{subject to} \quad A_2^j x_2 \leq b_2^j - B_2^j x_1,$$

Example: SAA approach

So, we see that we are estimating the expectation

$$\mathbb{E}_{\xi_3}\left[Q_3(x_1, x_2, \xi_2, \xi_3) \,|\, \xi_2 = \xi_2^j\right]$$

with a sample of *size one*

$$Q_3(x_1, x_2, \xi_2^j, \xi_3^j)$$

- There is no way we can obtain consistent estimators as $N \to \infty$!
- The situation can be fixed if we use conditional sampling:
  - That is, for each value of ξ_2^j we generate N samples $\xi_3^{j1}, \ldots, \xi_3^{jN}$.
  - Of course, this implies drawing N^2 samples overall!
  - For a T-stage problem, this implies drawing N^{T-1} samples. Impractical! We need to work with a fixed tree structure.

Example: processes generated by time series

Suppose that the underlying stochastic process $\xi := (\xi_1, \ldots, \xi_T)$ is driven by a time series model of the form

$$\xi_{t+1} = \psi(\xi_0, \ldots, \xi_t, U_t), \tag{1}$$

where $U_0, \ldots, U_{T-1}$ are independent uniform random vectors in $(0,1)^k$, and ξ_0 is a constant.

Example: processes generated by time series

Suppose that the underlying stochastic process $\xi := (\xi_1, \ldots, \xi_T)$ is driven by a time series model of the form

$$\xi_{t+1} = \psi(\xi_0, \ldots, \xi_t, U_t), \tag{1}$$

where $U_0, \ldots, U_{T-1}$ are independent uniform random vectors in $(0,1)^k$, and ξ_0 is a constant.

1. Let n_t denote the number of branches to be generated at each node in stage t;
2. For each $t = 0, \ldots, T-1$:
   1. Generate n_t Uniform(0,1) k-dimensional random vectors $U_t^1, \ldots, U_t^{n_t}$ using MC or QMC method;
   2. For each node in stage t, generate n_t children nodes using the relation (1) for each U_t^j, $j = 1, \ldots, n_t$.

Measuring distances in trees

Consider the input process $\{\xi_t\}$ of the MSSP, and let P be its "distribution".

As before, we would like to find another distribution Q that approximates P and and has fewer scenarios than P.

A more basic issue: What is the distribution of $\{\xi_t\}$?

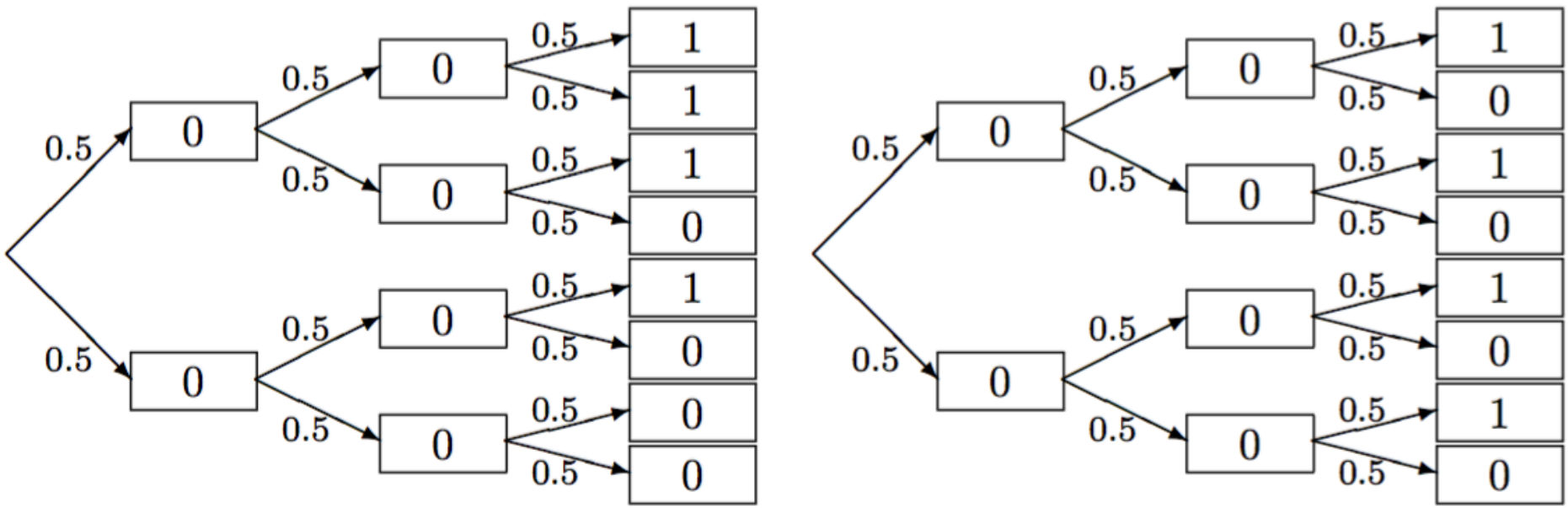
- The distribution of the stochastic process $(\xi_1, \ldots, \xi_T)$ is given by the joint distribution

  $\mathcal{P}(\xi_1 \in A_1, \ldots, \xi_T \in A_T)$ for any measurable sets $A_1, \ldots, A_T \subseteq \Xi_1 \times \ldots$

- Does that suffice?

An example

Consider the following 4-stage example, from Pflug (2009):



The processes on the left and on the right have the same distribution — but they clearly have different tree structures!

An example (cont.)

To see why this matters, consider the problem

$$\begin{aligned} \min_{x_1,x_2,x_3} \quad & \mathbb{E}\left[|x_1 - \xi_4| + |x_2 - \xi_4| + |x_3 - \xi_4|\right] \\ \text{s.t.} \quad & x_t \lhd \sigma(\xi_1, \ldots, \xi_t), \quad t = 1, 2, 3. \end{aligned}$$

What is the optimal solution for the processes in the previous slide?

- For the one on the left, an optimal solution is $x_t = 1$ on the upper side of the tree, $x_t = 0$ on the lower side.This yields an optimal value equal to 1.
- For the one on the right, any feasible solution is optimal, and corresponds to an objective value equal to 1.5.
- So, we need to define a distribution that can take the tree structure into account.

Nested distributions

Pflug (2009) define the notion of nested processes, in which the element at time t has two components: the random vector ξ_t, and the process that is conditional on each outcome of ξ_t.

The figure below illustrates the nested process for the tree on the left of the previous figure.

$$\left[\begin{array}{c} 0.5 \\ \hline 0 \end{array}\right.\quad \left[\begin{array}{cc} 0.5 & 0.5 \\ \hline 0 & 0 \end{array}\right.\quad \left[\begin{array}{cc} 0.5 & 0.5 \\ \hline 0 & 1 \end{array}\right]\quad \left[\begin{array}{cc} 0.5 & 0.5 \\ \hline 0 & 0 \end{array}\right.\quad \left[\begin{array}{cc} 0.5 & 0.5 \\ \hline 1 & 1 \end{array}\right]\left.\right]\left.\right]$$

It is possible then to define distributions of the nested processes, which are called *nested distributions*.

Nested distances

Working with nested distributions essentially means that we need not only the distribution of the stochastic process but also the conditional distributions.

We can then define a nested Kantorovich distance between two (nested) processes, which extends the M-K problem for conditional distributions.

Still, how to select the scenarios?...

An easier case

Mirkov and Pflug (2007) define *conditional Wasserstein distances* between the conditional distributions $P_t(\cdot \,|\, \xi_1, \ldots, \xi_{t-1})$ and $Q_t(\cdot \,|\, \xi_1, \ldots, \xi_{t-1})$.

- If a particular path $\xi_1, \ldots, \xi_{t-1}$ does not exist in the Q-tree, the distance is zero (assuming the Q-tree is a subset of the P-tree).
- Otherwise, the distance is the Wasserstein distance d_W for the respective conditional distributions.
- They also argue that the Wasserstein distance provide good coverage of the tails of a distribution.

An easier case (cont.)

They show that, if

$$
\begin{aligned}
d_W(P_1, Q_1) &< \varepsilon_1 \\
\sup_{\xi_1} d_W(P_2(\cdot \,|\, \xi_1),\ Q_2(\cdot \,|\, \xi_1)) &< \varepsilon_2 \\
\sup_{\xi_1,\xi_2} d_W(P_3(\cdot \,|\, \xi_1, \xi_2),\ Q_3(\cdot \,|\, \xi_1, \xi_2)) &< \varepsilon_2 \\
&\vdots \\
\sup_{\xi_1,\xi_{T-1}} d_W(P_T(\cdot \,|\, \xi_1, \ldots, \xi_{T-1}),\ Q_T(\cdot \,|\, \xi_1, \ldots, \xi_{T-1})) &< \varepsilon_T
\end{aligned}
$$

for $(\varepsilon_1, \ldots, \varepsilon_T)$ small enough, then the solutions of the problems obtained with P and Q trees are close.

It is also possible to show that, under proper assumptions, the above conditions imply that the corresponding nested distributions are close.

Scenario tree generation algorithm of Pflug and Pichler (2015)

For each stage $t = 1, \ldots, T-1$,

1. Fix a small number m_t of scenarios to be generated for each node in that stage;
2. For each node,
    1. Solve the k-dimensional facility location problem plus the M-K LP to find a conditional distribution Q_t with n_t points that approximates well the conditional distribution P_t at that node.

Alternatively, one can fix $\{\varepsilon_t\}$ and find the number of nodes m_t in each stage needed to achieve that precision.

Scenario reduction methods

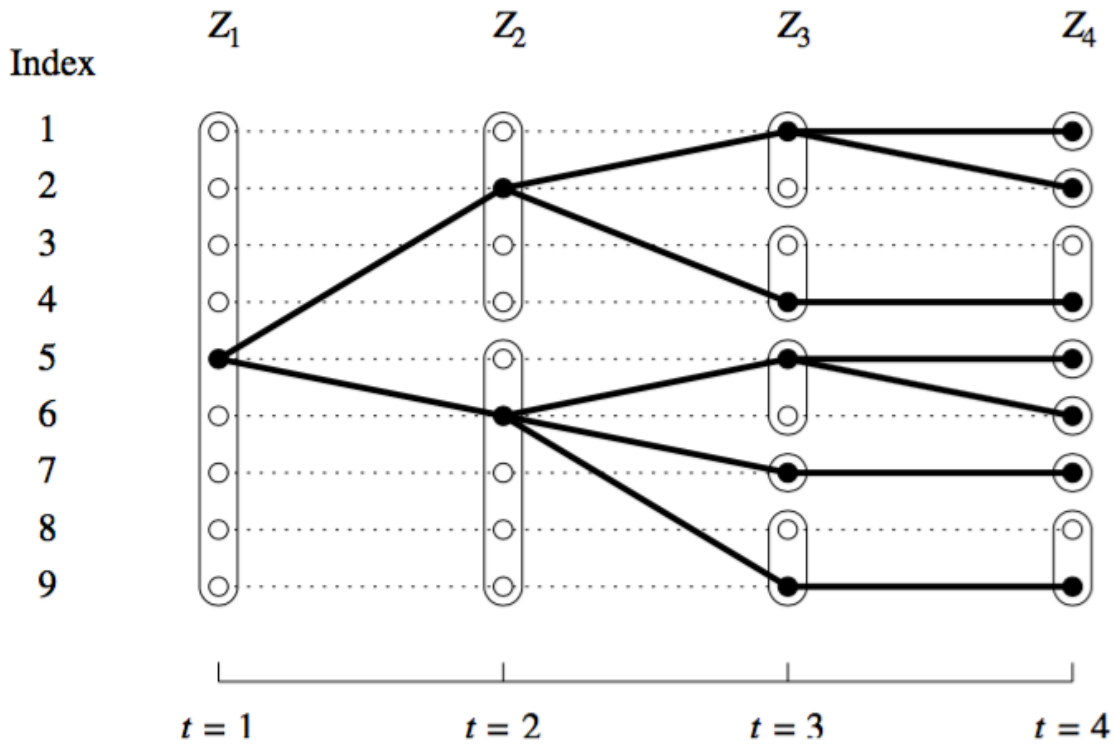
One alternative often used in practice is to start with a large representation of the process (e.g., obtained from sampling) and then apply a scenario reduction technique such as Heitsch and Römisch (2009):

- The idea is to find, for each stage, the set of nodes that better represent the conditional distributions.
- This can be done by solving a combinatorial problem, or by applying the forward/backward heuristics.
- The other nodes in that stage are then "merged" into the closest representative node.

The algorithm can be applied when we don't have direct access to the distributions, only to sample paths available from data.

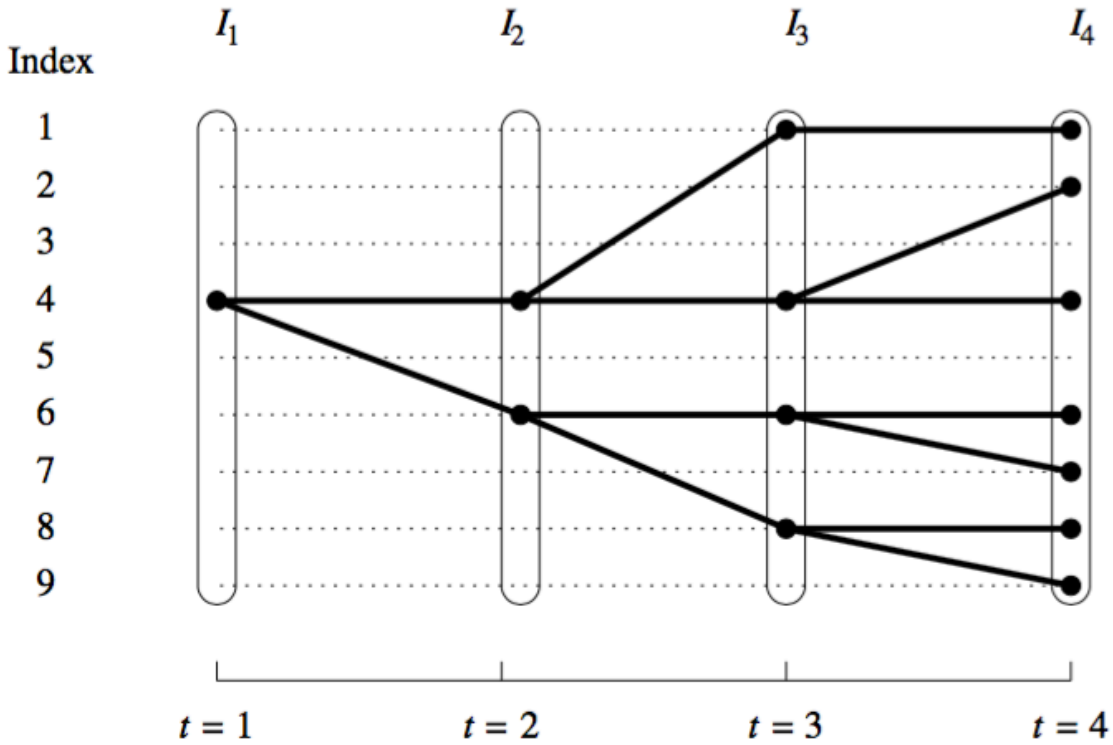
An example

Consider the following 4-stage example with forward construction, from Heitsch and Römisch (2009):



An example

Consider the following 4-stage example with backward construction, from Heitsch and Römisch (2009):



The independent case

The situation is considerably simplified when the process ξ_t is stage-wise independent.

- Although this seems a restrictive assumption, in many situations one can do a "modeling trick" to obtain such a property (see example below).
- The advantage is that in such case one only needs to apply scenario generation/reduction on each stage, with no concern for conditional distributions!

A case study: the Brazilian power system

(Work with E. Finardi and V. de Matos).

Composition:

- Hydro and Thermal power plants
- More than 70% of the power capacity is from *hydro plants*

A case study: the Brazilian power system

(Work with E. Finardi and V. de Matos).

Composition:

- Hydro and Thermal power plants
- More than 70% of the power capacity is from *hydro plants*

Characteristics of the hydrothermal scheduling:

- Stochastic, due to water inflows
- Coupled in time and space
- Nonlinear
- Large scale (139 Hydro plants, 146 Thermal plants)

The problem

- Operational decisions are periodically reviewed, and must take into account the decisions that will be made in the future.

The problem

- Operational decisions are periodically reviewed, and must take into account the decisions that will be made in the future.
- The decisions must also hedge against the uncertainty of water inflows.

The problem

- Operational decisions are periodically reviewed, and must take into account the decisions that will be made in the future.
- The decisions must also hedge against the uncertainty of water inflows.
- Goal is to determine the optimal policy that minimizes the expected total cost (consisting of thermal fuel costs and energy deficit penalties) over a horizon.

The problem

- Operational decisions are periodically reviewed, and must take into account the decisions that will be made in the future.
- The decisions must also hedge against the uncertainty of water inflows.
- Goal is to determine the optimal policy that minimizes the expected total cost (consisting of thermal fuel costs and energy deficit penalties) over a horizon.
- Models for long-, medium- and short-term planning are used by planners.

Simplifications

- Thermal units are considered to be linear.
- Demand is treated as deterministic.
- Hydro plants are aggregated into Energy Equivalent Reservoirs (EER).

Generating scenario trees

The water inflows at each EER are modeled using a calibrated periodic auto-regressive process of the form

$$I_t = \phi_1 I_{t-1} + \ldots + \phi_p I_{t-p} + \xi_t,$$

where the ϕ_k are coefficients and ξ_t is a random vector.

Generating scenario trees

The water inflows at each EER are modeled using a calibrated periodic auto-regressive process of the form

$$I_t = \phi_1 I_{t-1} + \ldots + \phi_p I_{t-p} + \xi_t,$$

where the ϕ_k are coefficients and ξ_t is a random vector.

- Note that such a structure yields random variables that are dependent across stages.

Generating scenario trees

The water inflows at each EER are modeled using a calibrated periodic auto-regressive process of the form

$$I_t = \phi_1 I_{t-1} + \ldots + \phi_p I_{t-p} + \xi_t,$$

where the ϕ_k are coefficients and ξ_t is a random vector.

- Note that such a structure yields random variables that are dependent across stages.
- Key transformation: By modeling the I_t's as state variables and assuming the random noise ξ_t does not depend on t, it follows that the input stochastic process is stagewise independent.

Generating scenario trees

The water inflows at each EER are modeled using a calibrated periodic auto-regressive process of the form

$$I_t = \phi_1 I_{t-1} + \ldots + \phi_p I_{t-p} + \xi_t,$$

where the ϕ_k are coefficients and ξ_t is a random vector.

- Note that such a structure yields random variables that are dependent across stages.
- Key transformation: By modeling the I_t's as state variables and assuming the random noise ξ_t does not depend on t, it follows that the input stochastic process is stagewise independent.
- Although this modeling choice increases the number of variables, it allows for easier scenario generation and also for the use of efficient algorithms.

Generating an initial tree

We assume that the underlying stochastic process $\xi := (\xi_1, \ldots, \xi_T)$ is such that the $\{\xi_t\}$ are independent and each ξ_t is a d-dimensional random vector with the ith component given by

$$\xi_t^i = e^{V_t^i} + \Delta_t^i. \tag{2}$$

In the above equation, the vector $V_t = (V_t^i)_{i=1,\ldots,d}$ is multivariate normal with mean vector μ_t and covariance matrix Σ_t, whereas the Δ_t^i are given parameters.

Generating an initial tree

We assume that the underlying stochastic process $\xi := (\xi_1, \ldots, \xi_T)$ is such that the $\{\xi_t\}$ are independent and each ξ_t is a d-dimensional random vector with the ith component given by

$$\xi_t^i = e^{V_t^i} + \Delta_t^i. \tag{2}$$

In the above equation, the vector $V_t = (V_t^i)_{i=1,\ldots,d}$ is multivariate normal with mean vector μ_t and covariance matrix Σ_t, whereas the Δ_t^i are given parameters.

1. Let N_t denote the number of branches to be generated at each node in stage t;
2. For each $t = 1, \ldots, T-1$:
   1. Generate N_t d-dimensional multivariate normal random vectors $V_{t,j}$, $j = 1, \ldots, N_t$ with mean μ_t and covariance matrix Σ_t;
   2. For each node in stage t, generate N_t children nodes using the relation (2) for each j and each i.

Generating smaller trees

For the initial tree, all we want is that it represents well the original process, so we could have a relatively large number of children for each each node.

Generating smaller trees

For the initial tree, all we want is that it represents well the original process, so we could have a relatively large number of children for each each node.

- These points can be generated for example by Quasi Monte Carlo to ensure representativeness.

In order to solve the problem, however, we need a much smaller tree (with, say, about 20 children per node).

We will discuss some scenario reduction approaches next.

Numerical illustration

We compared six alternatives for two problems — one with 3 stages and another with 5 stages.

- Each node has dimension 2, corresponding to two reservoirs.
- For the 3-stage problem, an initial tree with 75 children per node was generated.
- For the 5-stage problem, an initial tree with 50 children per node was generated.
- In both cases, we generated reduced trees with 5 children per node.

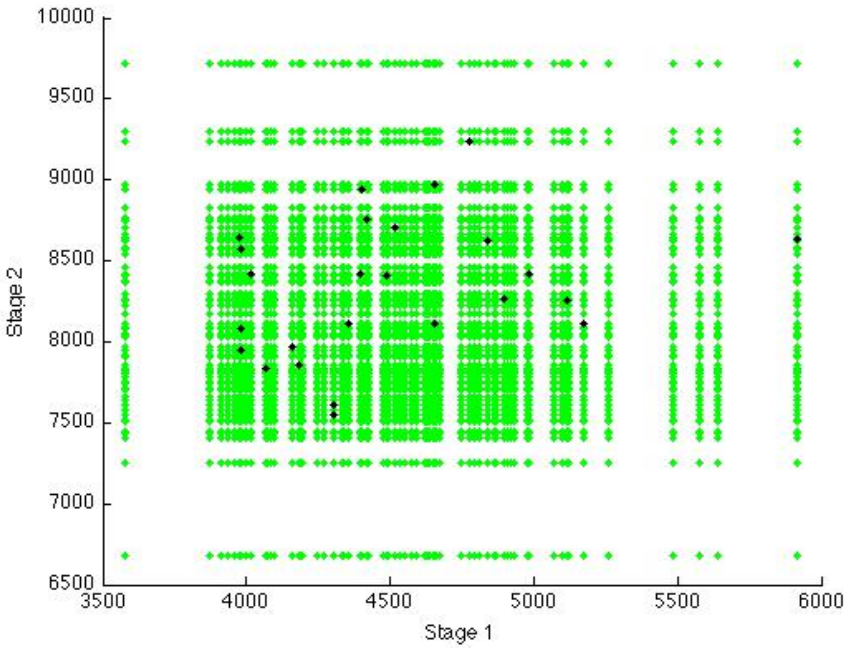
Numerical illustration

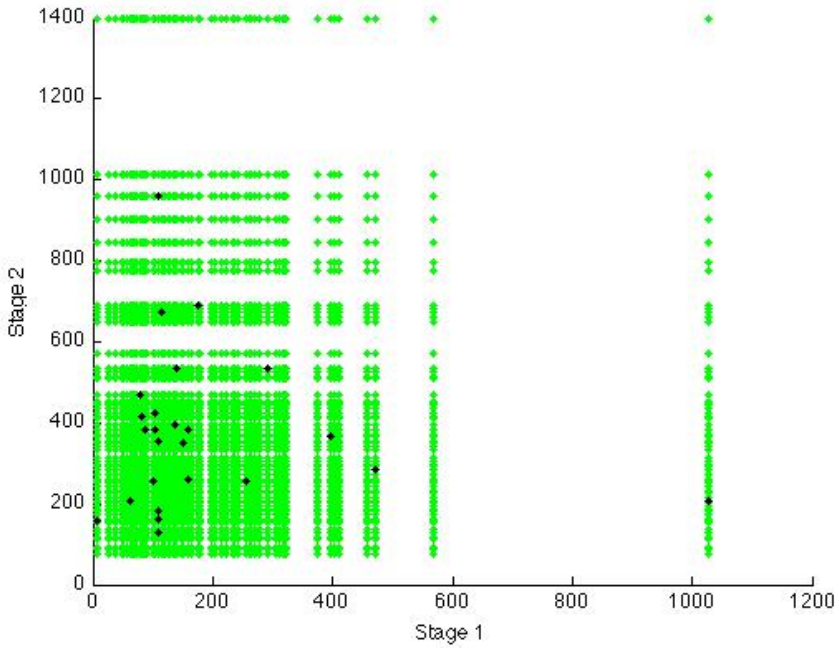
The six alternatives we compared with were:

- The full tree
- Random selection of scenarios in each stage
- Backward heuristic (only for 3-stage problem) for scenarios of the full tree
- Forward heuristic (only for 3-stage problem) for scenarios of the full tree
- Random selection of scenarios from the full tree (only for 3-stage problem)
- A facility location problem per stage (only for 3-stage problem)

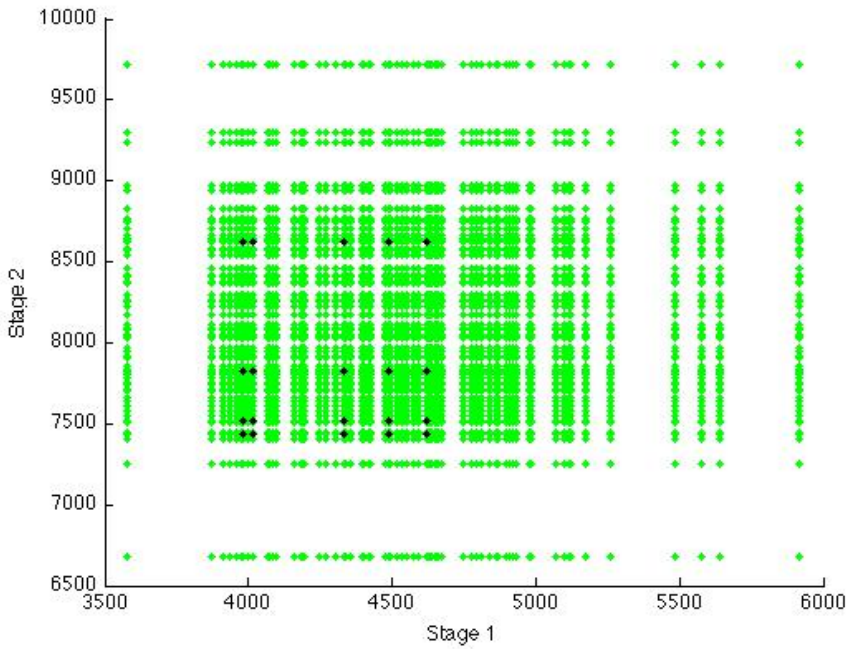
The graphs that follow depict the chosen scenarios in each case (without the probabilities).

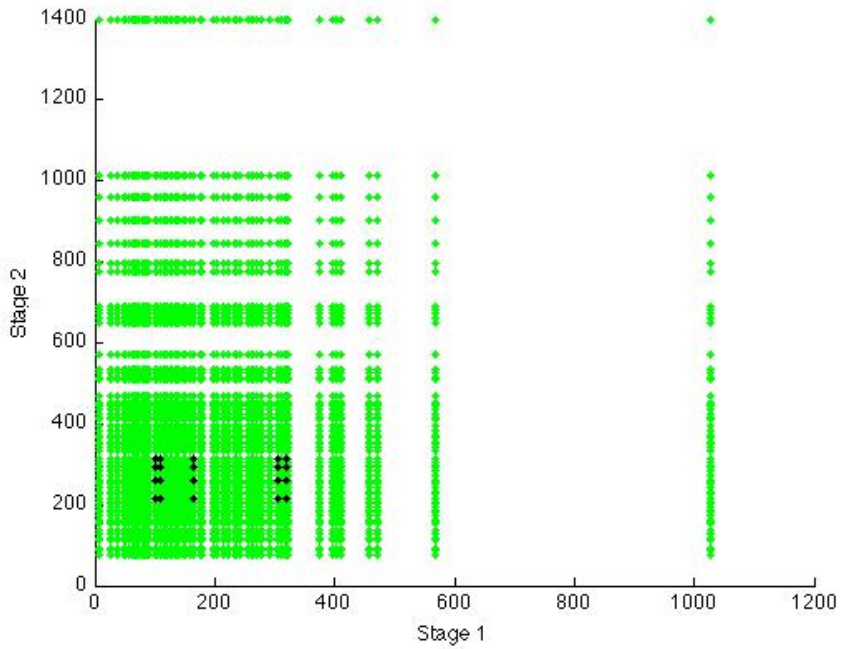
3-stage, random choice from full tree, both reservoirs



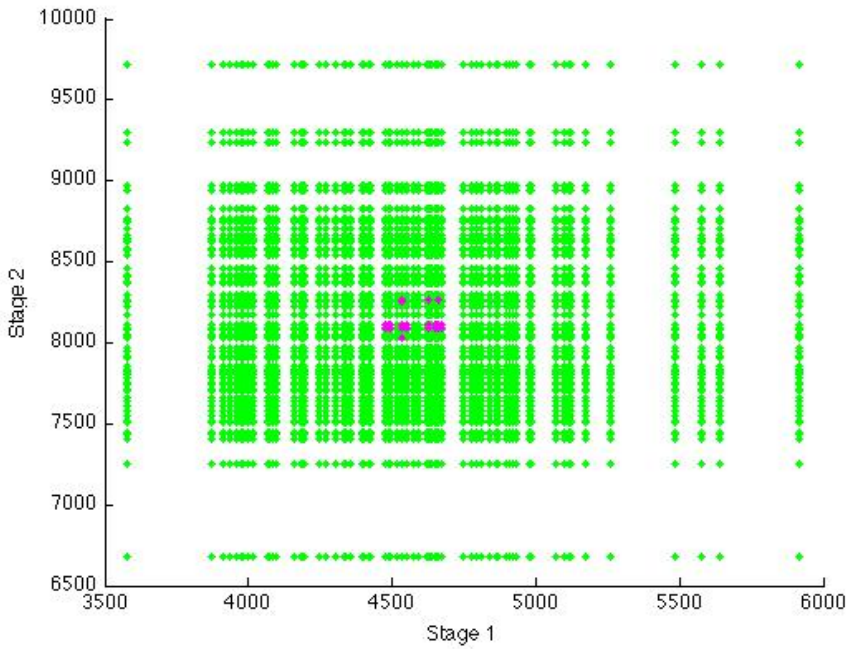


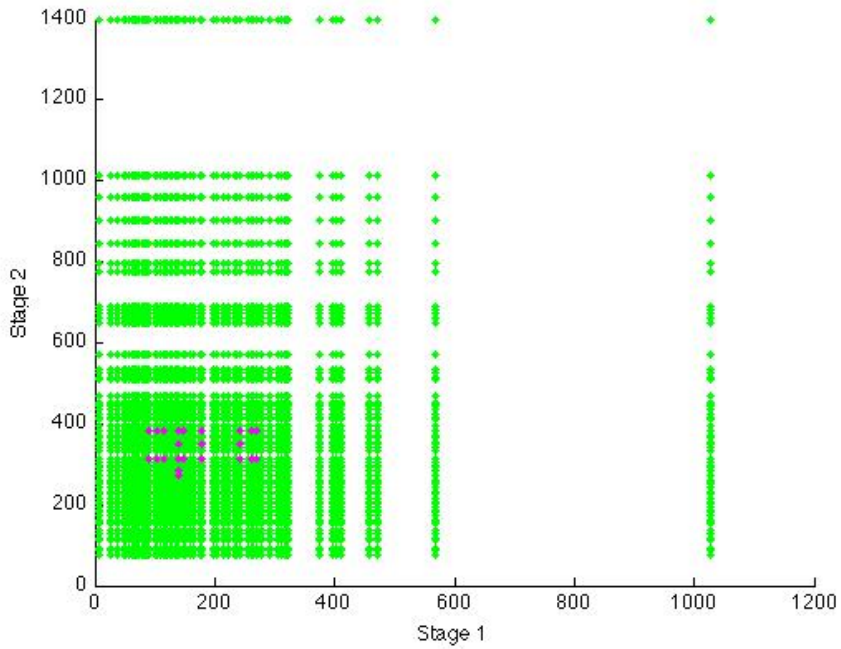
3-stage, random choice per stage, both reservoirs



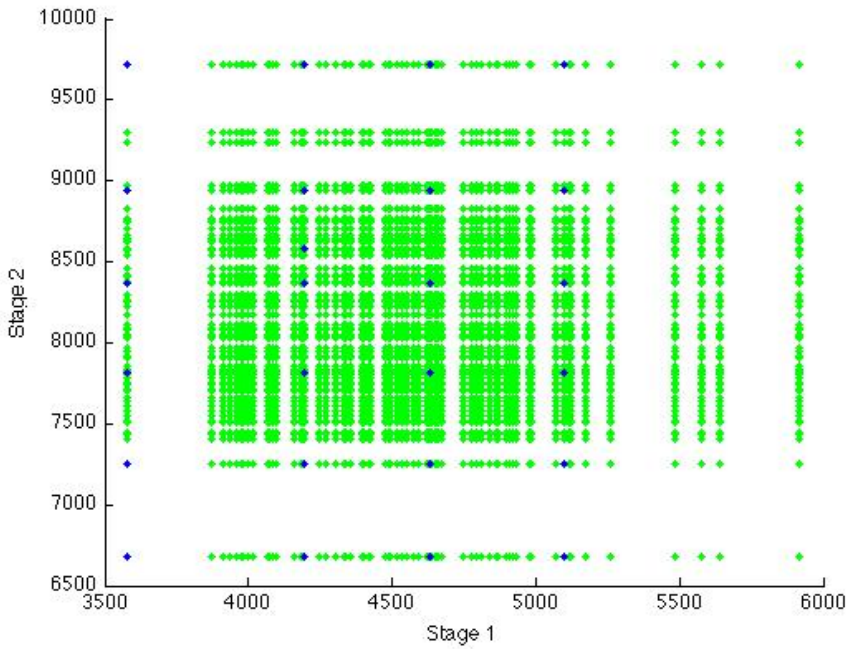


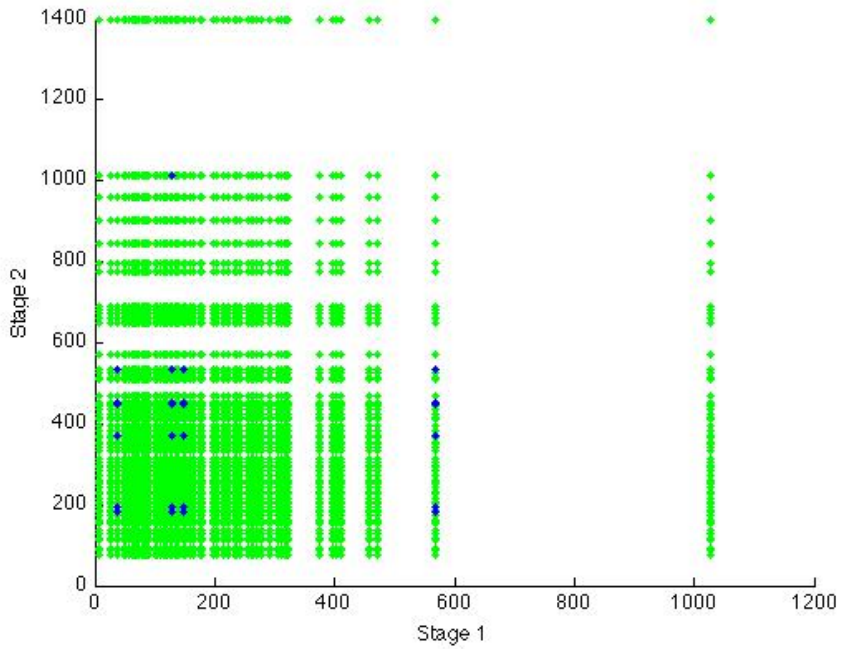
3-stage, forward heuristic, both reservoirs





3-stage, backward heuristic, both reservoirs

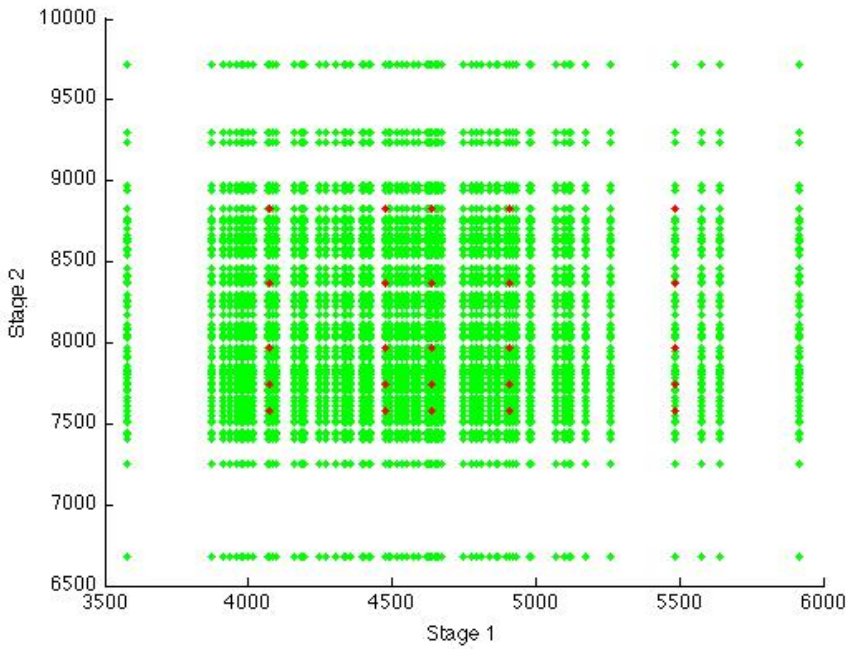


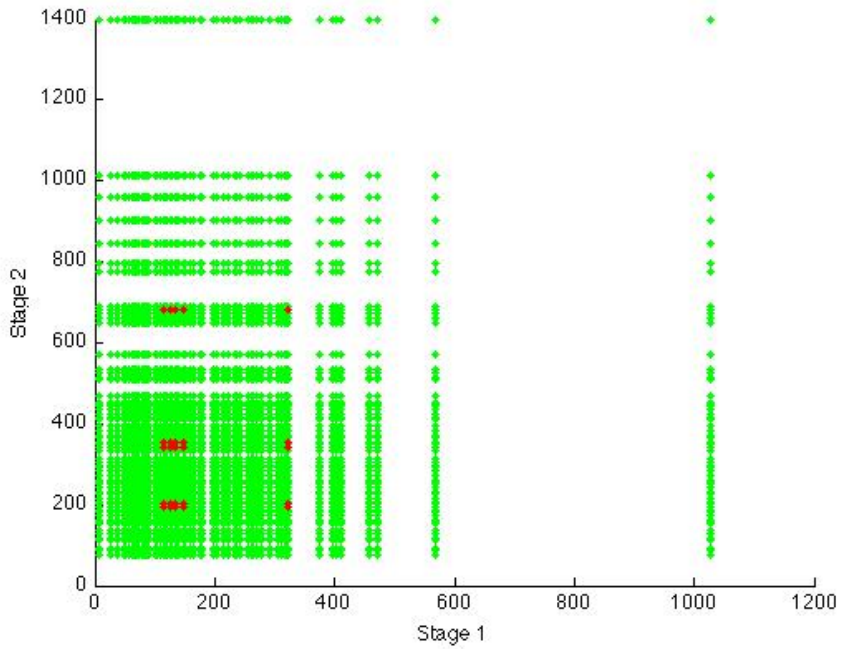


Chosen distribution for backward selection

Scenario	Probability
(3575.9, 37.0), (6683.7, 450.9)	0.000
(3575.9, 37.0), (7255.5, 193.7)	0.002
(3575.9, 37.0), (9719.4, 372.4)	0.000
(3575.9, 37.0), (7816.2, 446.4)	0.005
(3575.9, 37.0), (8365.6, 183.3)	0.005
(3575.9, 37.0), (8942.8, 536.2)	0.001
(4194.7, 128.5), (6683.7, 450.9)	0.006
(4194.7, 128.5), (7255.5, 193.7)	0.058
(4194.7, 128.5), (9719.4, 372.4)	0.013
(4194.7, 128.5), (8579.9, 1013.9)	0.042
(4194.7, 128.5), (7816.2, 446.4)	0.160
(4194.7, 128.5), (8365.6, 183.3)	0.166
(4194.7, 128.5), (8942.8, 536.2)	0.045
(4632.6, 568.6), (6683.7, 450.9)	0.002
(4632.6, 568.6), (7255.5, 193.7)	0.021
(4632.6, 568.6), (9719.4, 372.4)	0.005
(4632.6, 568.6), (7816.2, 446.4)	0.060
(4632.6, 568.6), (8365.6, 183.3)	0.063
(4632.6, 568.6), (8942.8, 536.2)	0.017
(5101.3, 148.9), (6683.7, 450.9)	0.004
(5101.3, 148.9), (7255.5, 193.7)	0.040
(5101.3, 148.9), (9719.4, 372.4)	0.009
(5101.3, 148.9), (7816.2, 446.4)	0.115
(5101.3, 148.9), (8365.6, 183.3)	0.124
(5101.3, 148.9), (8942.8, 536.2)	0.037

3-stage, facility location, Wasserstein order 1

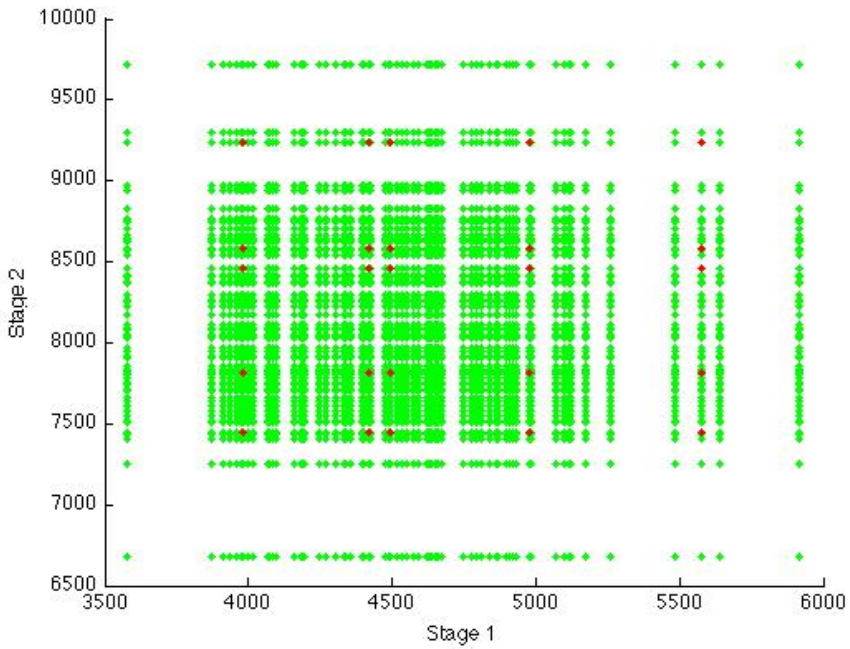


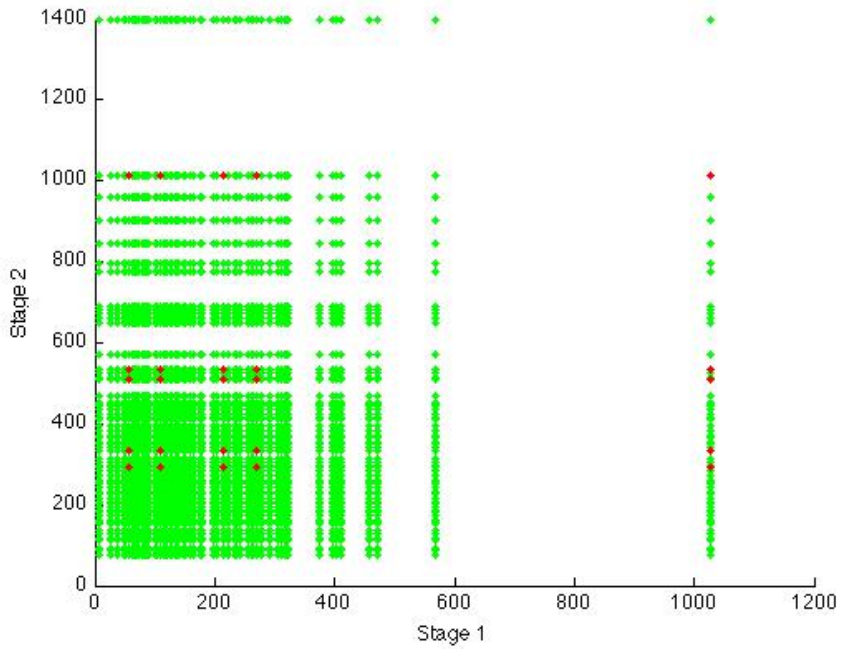


Chosen distribution for Wasserstein order 1

Scenario	Probability
(4074.5, 115.5), (8371.2, 340.1)	0.064
(4074.5, 115.5), (8830.0, 352.5)	0.054
(4074.5, 115.5), (7968.0, 194.7)	0.047
(4074.5, 115.5), (7739.0, 680.4)	0.041
(4074.5, 115.5), (7578.7, 200.7)	0.047
(4478.1, 148.8), (8371.2, 340.1)	0.068
(4478.1, 148.8), (8830.0, 352.5)	0.057
(4478.1, 148.8), (7968.0, 194.7)	0.050
(4478.1, 148.8), (7739.0, 680.4)	0.043
(4478.1, 148.8), (7578.7, 200.7)	0.050
(4639.8, 320.8), (8371.2, 340.1)	0.041
(4639.8, 320.8), (8830.0, 352.5)	0.034
(4639.8, 320.8), (7968.0, 194.7)	0.030
(4639.8, 320.8), (7739.0, 680.4)	0.026
(4639.8, 320.8), (7578.7, 200.7)	0.030
(4911.2, 134.3), (8371.2, 340.1)	0.064
(4911.2, 134.3), (8830.0, 352.5)	0.054
(4911.2, 134.3), (7968.0, 194.7)	0.047
(4911.2, 134.3), (7739.0, 680.4)	0.041
(4911.2, 134.3), (7578.7, 200.7)	0.047
(5482.6, 124.6), (8371.2, 340.1)	0.017
(5482.6, 124.6), (8830.0, 352.5)	0.014
(5482.6, 124.6), (7968.0, 194.7)	0.012
(5482.6, 124.6), (7739.0, 680.4)	0.011
(5482.6, 124.6), (7578.7, 200.7)	0.012

3-stage, facility location, Wasserstein order 3





Chosen distribution for Wasserstein order 3

Scenario	Probability
(3983.5, 109.4), (9238.8, 532.4)	0.016
(3983.5, 109.4), (8459.1, 290.9)	0.090
(3983.5, 109.4), (7445.2, 510.1)	0.035
(3983.5, 109.4), (8579.9, 1013.9)	0.013
(3983.5, 109.4), (7818.5, 333.4)	0.086
(4418.5, 1026.5), (9238.8, 532.4)	0.001
(4418.5, 1026.5), (8459.1, 290.9)	0.005
(4418.5, 1026.5), (7445.2, 510.1)	0.002
(4418.5, 1026.5), (8579.9, 1013.9)	0.001
(4418.5, 1026.5), (7818.5, 333.4)	0.005
(4495.0, 270.6), (9238.8, 532.4)	0.026
(4495.0, 270.6), (8459.1, 290.9)	0.144
(4495.0, 270.6), (7445.2, 510.1)	0.057
(4495.0, 270.6), (8579.9, 1013.9)	0.021
(4495.0, 270.6), (7818.5, 333.4)	0.139
(4976.5, 214.6), (9238.8, 532.4)	0.020
(4976.5, 214.6), (8459.1, 290.9)	0.114
(4976.5, 214.6), (7445.2, 510.1)	0.045
(4976.5, 214.6), (8579.9, 1013.9)	0.016
(4976.5, 214.6), (7818.5, 333.4)	0.110
(5576.4, 55.9), (9238.8, 532.4)	0.004
(5576.4, 55.9), (8459.1, 290.9)	0.020
(5576.4, 55.9), (7445.2, 510.1)	0.008
(5576.4, 55.9), (8579.9, 1013.9)	0.003
(5576.4, 55.9), (7818.5, 333.4)	0.019

References

[1] M. A. H. Dempster, E. A. Medova, and Y. S. Yong. Comparisons of sampling methods for dynamic stochastic programming. In M. Bertocchi, G. Consigli, and M. A. H. Dempster, editors, *Stochastic Optimization Methods in Finance and Energy*, chapter 389–425. Springer, 2011.

[2] H. Heitsch and W. Römisch. Scenario tree modeling for multistage stochastic programs. *Math Program*, 118:371–406, 2009.

[3] R. Mirkov and G. C. Pflug. Tree approximations of dynamic stochastic programs. *SIAM J. Optim.*, 18(3):1082–1105, 2007.

[4] G. C. Pflug. Version-independence and nested distributions in multistage stochastic optimization. *SIAM Journal on Optimization*, 20 (3):1406–1420, 2009.

[5] G. C. Pflug and A. Pichler. Dynamic generation of scenario trees. *Comput. Optim. Appl.*, 62(3):641–668, Dec. 2015.