Progressive Hedging for Multi-stage Stochastic Optimization Problems

David L. Woodruff Jean-Paul Watson

Graduate School of Management
University of California, Davis
Davis, CA 95616, USA
dlwoodruff@ucdavis.edu

Sandia National Laboratories
Discrete Math and Complex Systems Department
Albuquerque, NM 87185, USA
jwatson@sandia.gov

(*Stochastic Programming, Mixed Integer Programming, Progressive Hedging, Parallel Computing,*)

If You Say to Us:

"I have an optimization problem and I can find solutions, but now I want to address the fact that many of the parameters are uncertain."

Introduction

- **We are interested in large scale, multi-stage, (perhaps mixed-integer), stochastic programs.**

- **We are primarily interested in stochastic objectives that can be expressed as an expectation, which includes CVaR.**

- **Progressive hedging (PH) is a method based on scenario decomposition (Rockafellar and Wets).**

- **We have a (parallel) implement ion of PH that we are using to test ideas for getting good speed-up. It is available for download.**

Sample Applications

- **Energy system design (e.g., Yueyue Fan's poster)**

- **Sensor placement in water networks**

- **Forest harvest planning**

Scenarios

- **The modeler has some scenarios and often we really don't care where they got them (see Dr. Römisch).**

- **For each scenario $s \in \mathcal{S}$, we denote the probability of occurrence by $\Pr(s)$.**

- **If there multiple decision stages, then we need a tree.**

- **Aside: sometimes the scenarios are obtained by solving PDEs.**

Scenario Problem

If you somehow know the scenario s that will occur, then the goal is to minimize:

$$f_s(X(s)) \quad (\mathrm{P}_s)$$

subject to the constraint

$$X \in \Omega_s.$$

- **We use Ω_s to express all constraints for scenario s.**
- **Stage subscripts not shown.**

Non-anticipativity

- ***Admissible*** **solution systems satisfy constraints for all scenarios.**

- **A system of solution vectors is** ***implementable*** **if for all pairs of scenario s and s' that are indistinguishable up to time t, $x_i(s, t') = x_i(s', t')$ for all $1 \leq t' \leq t$ and each i in each $N(t)$.**

- **We refer to the set of all implementable solution systems as $\mathcal{N}_{\mathcal{S}}$ for a given set of scenarios, $\mathcal{S}$.**

Expected Value Minimization

$$\min \sum_{s \in \mathcal{S}} [\Pr(s) f(s; X(s))] \qquad \text{(P)}$$

subject to

$$\begin{aligned} X &\in \Omega_s \\ X &\in \mathcal{N}_{\mathcal{S}}. \end{aligned}$$

1 Progressive Hedging

The basic idea of PH for the linear case is as follows:

1. **For each scenario s, solutions are obtained for the problem of minimizing, subject to the problem constraints, the deterministic f_s (Formulation P_s).**

2. **The variable values for an implementable – but likely not admissible – solution are obtained by averaging over all scenarios at a scenario tree node.**

3. **For each scenario s, solutions are obtained for the problem of minimizing, subject to the problem constraints, the deterministic f_s (Formulation P_s) plus terms that penalize the lack of implementability using a sub-gradient estimator for the non-anticipativity constraints and a squared proximal term.**

4. **If the solutions have not converged sufficiently and the allocated compute time is not exceeded, goto Step 2.**

5. **Post-process, if needed, to produce a fully admissible and implementable solution.**

Tree Notation

- **Scenarios: each leaf is connected to exactly one node at time $t \in \mathcal{T}$ and each of these nodes represents a unique realization up to time t.**

- **Two scenarios whose leaves are both connected to the same node at time t have the same realization up to time t.**

- **Consequently, in order for a solution to be implementable it must be true that if two scenarios are connected to the same node at some time t, then the values of $x_i(t')$ must be the same under both scenarios for all i and for $t' \leq t$.**

More Tree Notation

- $\Pr(\mathcal{A})$ denotes the sum of $\Pr(s)$ over all s for scenarios emanating from node $\mathcal{A}$

- $t(\mathcal{A})$ to indicates the time index for node $\mathcal{A}$.

- $X(t;\mathcal{A})$ on the left hand side of a statement indicates assignment to the vector $(x_1(s,t),\ldots,x_{N(t)}(s,|\mathcal{T}|))$ for each $s \in \mathcal{A}$.

- We refer to vectors at each iteration of PH using a superscript; e.g., $w^{(0)}(s)$ is the multiplier vector for scenario s at PH iteration zero.

PH

Taking $\rho > 0$ as a parameter.

1. $k := 0$
2. **For all scenario indexes, $s \in \mathcal{S}$:**

$$X^{(0)}(s) := \text{argmin}\, f_s(X(s)) : X(s) \in \Omega_s \tag{1}$$

   and

$$w^{(0)}(s) := 0$$

3. $k := k + 1$
4. **For each node, $\mathcal{A}$, in the scenario tree, and for all $t = t(\mathcal{A})$:**

$$\overline{X}^{(k-1)}(t; \mathcal{A}) := \sum_{s \in A} \Pr(s) X(t; s)^{(k-1)} / \Pr(\mathcal{A})$$

5. **For all scenario indexes, $s \in \mathcal{S}$:**

$$w^{(k)}(s) := w^{(k-1)}(s) + (\rho) \left(X^{(k-1)}(s) - \overline{X}^{(k-1)} \right)$$

   and

$$X^k(s) := \text{argmin}\, f_s(X(s)) + w^{(k)}(s) X(s) + \rho/2 \left\| X(s) - \overline{X}^{k-1} \right\|^2 : X(s) \in \tag{2}$$

6. **If the termination criteria are not met (e.g., solution discrepancies quantified via a metric $g^{(k)}$), then goto Step 3.**

Important Details

- **Termination**

- **Forcing Convergence**

- **Setting and changing $\boldsymbol{\rho}$ or $\boldsymbol{\rho_i}$.**

- **Solution quality or resolution need not be as high during early iterations.**

CVaR

- The expected value of the $1 - \alpha$ tail (see Dr. Rockafellar on Thursday).
- Bounds VaR and probably the way to get VaR if you need it.

CVaR can be implemented by augmenting this problem. Following the implementation given by Schultz and Tiedemann we add a real-valued, scalar variable, η, and for every scenario $s \in \mathcal{S}$ add a non-negative, real-valued, scalar variable ν_s.

A constraint is needed:

$$\nu_s \geq f_s(x_s) - \eta \quad \forall s \in \mathcal{S} \tag{3}$$

and the CVaR objective function is:

$$\left(\eta + \frac{1}{1-\alpha} \sum_{s \in \mathcal{S}} Pr(s) \nu_s \right). \tag{4}$$

Chance Constraints

- **The "100-year flood protection" model: some fraction of the constraints can be ignored.**

- **Almost the opposite of CVaR**

- **Can be implemented as a stochastic program with integer variables to select the constraints that will be ignored.**

- **This can be incorporated in the PH algorithm (Watson, Wets, Woodruff)**

Confidence Intervals

- **If the scenario problems can be bounded, then consider the multiple replication procedure of Mak, Morton and Woods described yesterday**
- **(See Dr. Shapiro and/or Dr. Morton)**