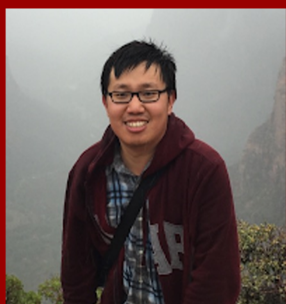
Compromise Decisions for Validating Stochastic Programming Policies and Decisions

Suvrajeet Sen in Collaboration with Jiajun Xu and Yifan Liu
https://sites.google.com/site/uscdatadrivendecisions

Thank You to Sponsors and Collaborators
Sponsors: NSF, AFOSR, and ONR

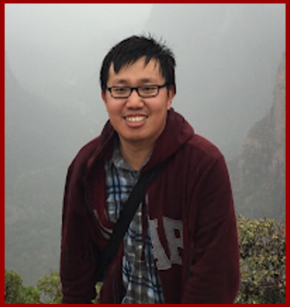


Liu, Yifan
(84.51)

Xu, Jiajun
(Meta-Facebook)

Thank You to Sponsors and Collaborators

Sponsors: NSF (30), AFOSR (15), and ONR(3)



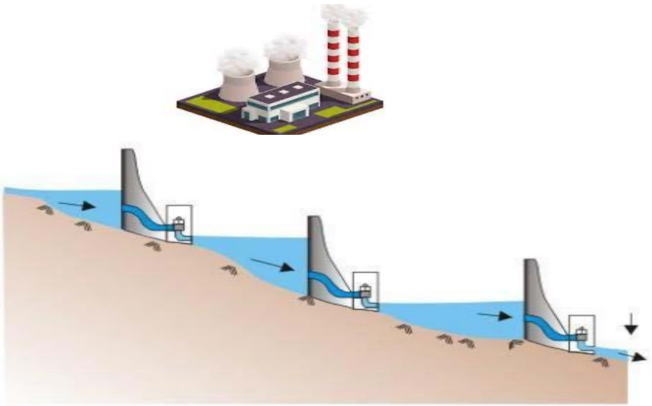
Liu, Yifan
(84.51)

Xu, Jiajun
(Meta-Facebook)

In this talk Dr. Xu cautions us against over-confidence from "small" sample sizes

Traditional SP Applications

Finance



Hydro-Thermal
Power Generation

Logistics and
Transportation

Plan for this Presentation

- **The Setting of Two-Stage Stochastic LP**
- **Brief Tour of Sampling Methods for SLP**
- **Validation of Sampling Methods**
- **Compromise Decisions: Computational View of Model Validation**
  I. Two-Stage Stochastic Linear Programming (based in Sen and Liu: appeared in *Operations Research, 2016*)
  II. *Multi-stage Stochastic Linear Programming (Xu, 2022)
  III. *Two-Stage Stochastic Combinatorial Programming (Xu, 2022)
- **Conclusions**

*Xu, J. (2022) "Computational Validation of Stochastic Programming Models and Applications," Ph.D. Dissertation, ECE Dept., Univ. of Southern California.

The Two-Stage Setting of Stochastic Linear Programming

- The overall problem: $\text{Min}\,\{c(x) + \boldsymbol{E_P}[h(x, \tilde{\omega}]: x \in X\}$
- For a given $\omega_n = (\xi_n, C_n)$ denotes a scenario/outcome

$$h(\omega_n) = \left\{ \begin{array}{c} \text{Min} \quad \boldsymbol{d}_n^\top u_n \\ \text{s.t. } u_n \in U_n(x), \\ U_n(x) = \{u_n \mid \boldsymbol{D} u_n \leq \xi_n - C_n x\} \end{array} \right\}$$

Remarks:

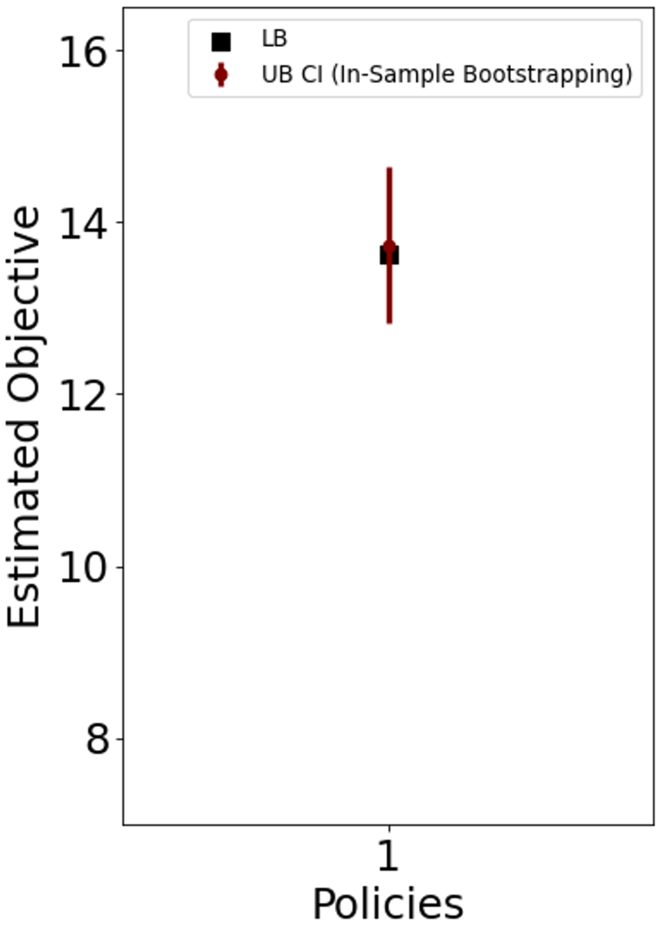
- For high-dimensional sample spaces, $E_P[h(x, \tilde{\omega})]$ is not computable in reasonable time (SP is #P hard (Hanasusanto, Kuhn and Weisemann (15) and Dyer and Stougie (06)) … Hence SP Algorithms use Sampling
- For SVM models, the first stage of an SVM chooses the pair of half-spaces specified by a first stage vector, and the second stage identifies whether a point belongs to one side of the separating hyperpane or the other.
- Kernel SVMs lead to Two-stage Stochastic Quadratic Programs

Brief Tour of Sampling-Based Methods

- Sample Average Approximation (aka External Sampling)
  - Kleywegt, Shapiro, Homem-de-Mello (02)
  - Periera, Pinto (**91**) … Stochastic Dual Dynamic Programming (SDDP)
- Adaptive/Incremental/Sequential Sampling
  - Stochastic Quasi-Gradient (Ermoliev, Gaivoronski, Norkin, Uryasiev… 60’s-80’s)
  - Importance Sampling (Dantzig, Glynn, Infanger, **91**)
  - Robust Stochastic Approx. (Nemirovski, Juditsky, Lan, Shapiro 09)
  - Stochastic Decomposition (Higle, Sen **91,** 94 … Sampling and Regularization)
  - Royset (13)
  - Royset and Szechtman (13)
  - Pasupathy and Song (21)
  - Chen, Menickelly, Scheinberg (16)
  - Blanchet, Cartis, Menickelly, Scheinberg (19)
- Statistical Stopping (Bootstrapping and Replications)
  - Higle, Sen (91b, 96, 99)
  - Shapiro and Homem-de-Mello (98)
  - Mak, Morton and Wood (99)
  - Nesterov and Vial (00/08)
  - Bayraksan and Morton (06, 11)
  - Bayraksan and Pierre-Louis (12)
  - Sen and Y. Liu (16) …. Genesis of today’s talk
  - J. Liu and Sen (19)
- Machine Learning (Explosive Literature)

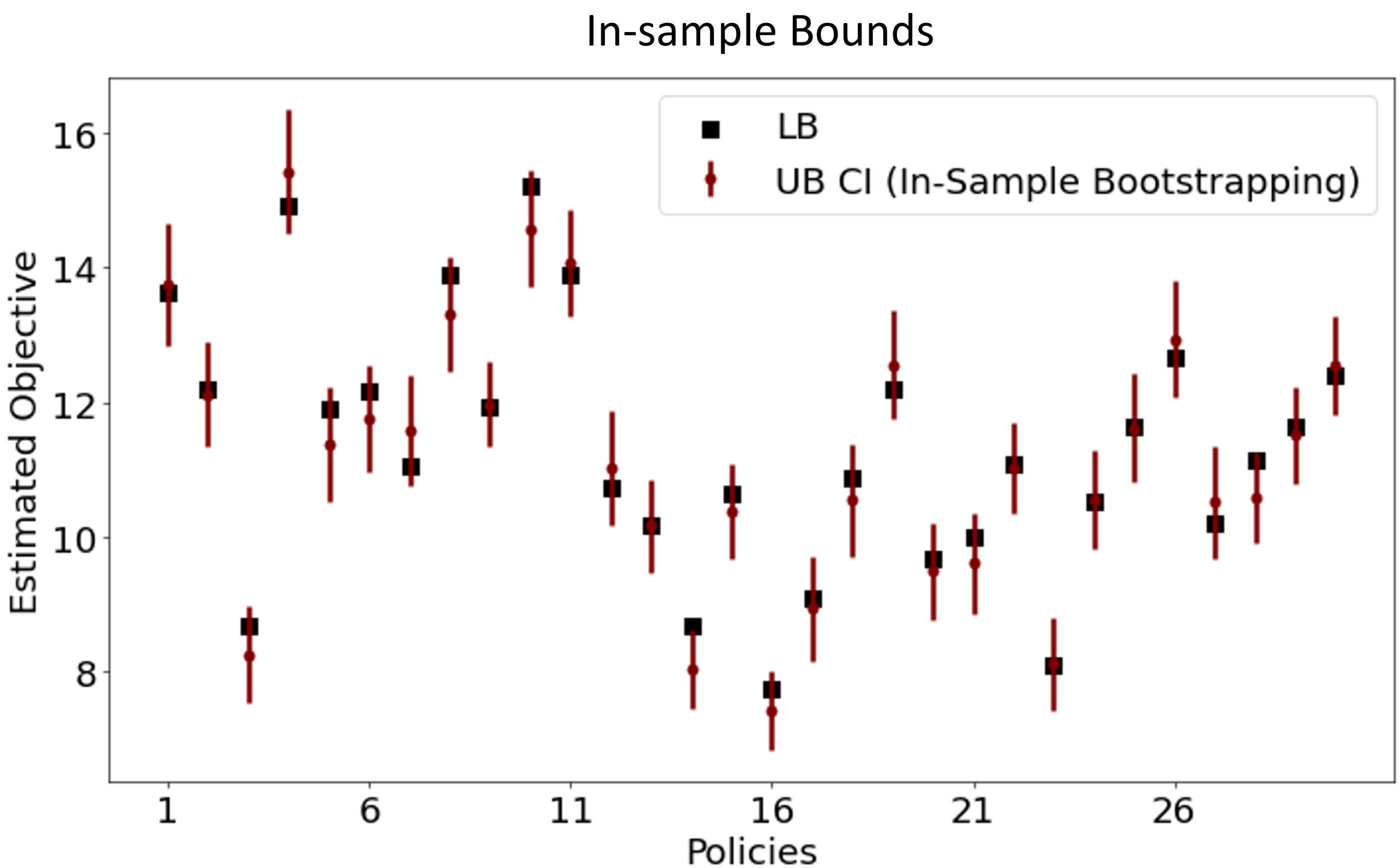
Common Strategy for SP: SAA

1. Formulate a sample average approximation (SAA) problem using either given data or a simulator (e.g., daily wind, or monthly precipitation)
2. Use Some SP algorithm to solve the SAA problem
3. Replicate if necessary



What can happen if we use different seeds

- Hydro-thermal scheduling problem
- 10^8 scenarios



What should we do to obtain Decisions/Policies ?

The Two-Stage Setting of Stochastic Linear Programming

- The overall problem: $\text{Min}\ \{c(x) + \boldsymbol{E_P}[h(x, \tilde{\omega}]: x \in X\}$
- For a given $\omega_n = (\xi_n, C_n)$ denotes an outcome

$$h(x, \omega_n) = \left\{ \begin{array}{c} \text{Min} \quad \boldsymbol{d}_n^\top u_n \\ \text{s.t. } u_n \in U_n(x), \\ U_n(x) = \{u_n \,|\, \boldsymbol{D} u_n \le \xi_n - \ C_n x\} \end{array} \right\}$$

- In our studies we make the "Fixed-Recourse" assumption ($\boldsymbol{D}$ is fixed)
- Randomness can be allowed for second stage costs $\boldsymbol{d}$
- For high-dimensional sample spaces, $E_P[h(x, \tilde{\omega})]$ is not computable in reasonable time (SP is #P hard (Hanasusanto, Kuhn and Weisemann (15) and Dyer and Stougie (06)) … Hence SP Algorithms use Sampling

Two-Stage "Compromise Decision"

– **Replicate Algorithmic Process**
  - $x^s \in \varepsilon - \text{argmin.} \left\{ f_s \ (x) + \frac{\sigma_s}{2} \|x - x^s\|^2 \ | \ x \in X \right\}$
  - Here f_s denotes the **terminating value function approximation**
– **Obtain a Compromise Decision**
  - $\boldsymbol{x^c} = \text{argmin}\ \{\text{"Grand Mean" Value Function}\} =$ $\text{argmin} \left\{ \sum_s \frac{1}{|S|} \left[f_s(x) + \frac{\bar{\sigma}}{2} \|y^s\|^2 \right] : x - y^s = x^s, x \in X \right\}$
    – Here $\boldsymbol{\bar{\sigma}}$ sample average Sample Average **Proximal** Term
– **Let $\boldsymbol{\bar{x}}$** denote sample average of **replication solutions** x^s . (ML refers to $\bar{x}$ as the bagging solution. No Compromise Decision in ML)
– If $x^c \approx \bar{x}$ stop. Else, each replication is run for more samples.

“Compromise Solution” Reduces Bias

- A tighter lower bound estimate

Theorem The optimal solution of the compromise problem is called the compromise solution, defined as $x^c = \operatorname*{argmin}_{x \in X} \bar{f}_M(x)$, and the corresponding optimal value is denoted as v^c. Then,

$$\mathbb{E}[\bar{v}_N^M] \le \mathbb{E}[v^c] \le v^*$$

Ordinary SAA lower bound estimate

optimal value

Compromise lower bound estimate

Compromise Solution Reduces Variance

Theorem: Let M denote the number of replications, and $\bar{x}$ the sample average solution. If $x^c = \bar{x}$, then both are Optimal

and $|f_s(x^s) - f(x^s)| = O_p(N^{-1/2})$, where N is the common sample size among all runs.

And, $|f(x^c) - \bar{F}_M(x^c)| = O_p\big((NM)^{-1/2}\big)$ where, $\bar{F}_M(x^c)$ is the value of the Grand Sample Mean Function

Some “concrete” instances

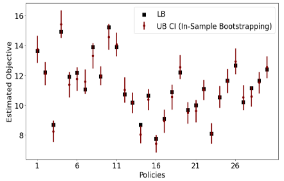
Table 1: SP Test Instances

Problem Name	Domain	# of 1st stage vars.	# of 2nd stage vars.	# of random variables	Universe of scenarios	Comment
LandS	Electric Power	4	12	3	$O(10^6)$	Made-up
20TERM	Logistics	63	764	40	$O(10^{12})$	Semi-real
SSN	Telecom	89	706	86	$O(10^{70})$	Semi-real
STORM	Logistics	121	1259	117	$O(10^{81})$	Semi-real

Sampling Approaches: External Approach (SAA) and Internal Approach (SA/SD)

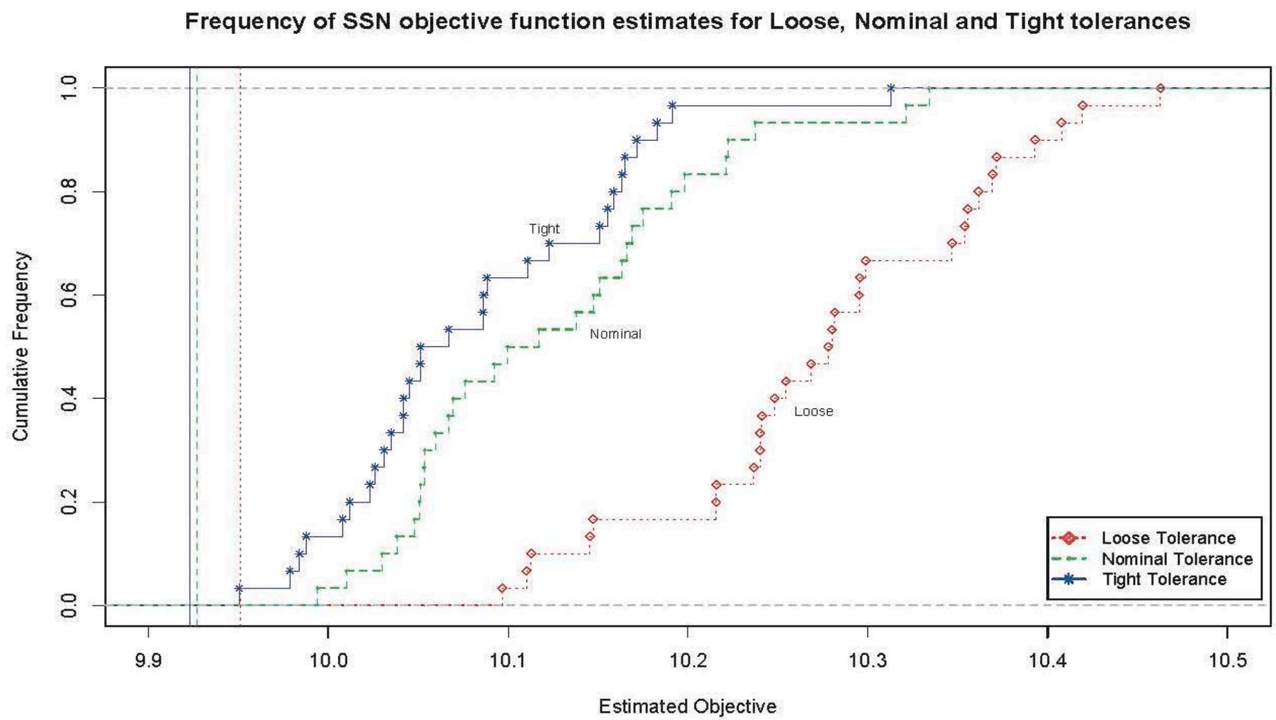
Some Comments on SSN Experiments - Anomalies Reported in the Literature

- Scenarios in SSN were generated using data from an actual network planning application from Atlanta (many years ago)
- For large sample size (*N* = 5000), Latin Hypercube Sampling, the solutions ($\hat{x}_N$) are very far apart, even though the objective functions are close to being the same.
- Such instances are sometimes referred to as "ill-conditioned" problem where "ill conditioning" has a specific meaning for sampling-based methods (introduced by Shapiro).
- For small sample sizes (say *N* = 50), the variance is small (because many objective estimates are 0). However, as the sample size is increased, the variance also increases, although the bias reduces

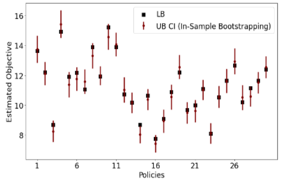


Convergence with high Probability

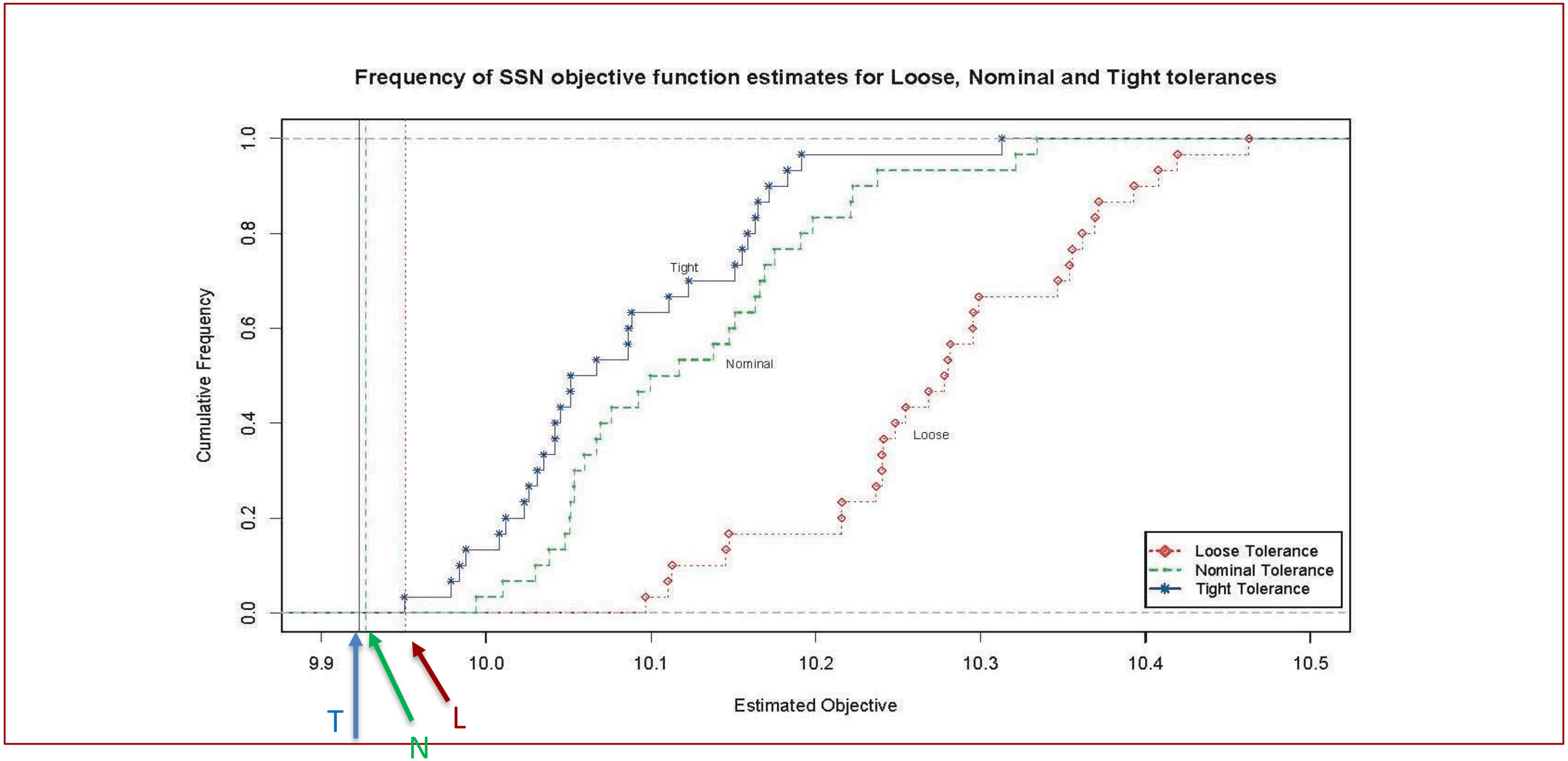
using SD From “Loose-to-Nominal-to-Tight” Tolerance



Each Series of Replications Represents a Tolerance Level: Loose, Nominal, Tight using Stochastic Decomposition (SD) Stopping Rules



After Replications Get Compromise Decisions (Two-Stage SLP)



Values of Compromise Decisions have Lower Bias and Lower Variance!

Multi-stage SLP (Stochastic Dual DP - SDDP)

- System dynamics

$$x_{t+} = \mathcal{D}(x_t, u_t, \omega_t) = a_t + A_t x_t + B_t u_t$$

- Formulation

$$\langle c_0, x_0 \rangle + \min \langle d_0, u_0 \rangle + \mathbb{E}_{\tilde{\omega}_{(0)}} \left[\langle c_1, x_1 \rangle + \langle d_1, u_1 \rangle + \mathbb{E}_{\tilde{\omega}_{(1)}} [\ldots + \mathbb{E}_{\tilde{\omega}_{T-1}} [\langle c_T, x_T \rangle + \langle d_T, u_T \rangle]] \right]$$

$$s.t.\ u_t \in \mathcal{U}_t(x_t) \coloneqq \{u_t | D_t u_t \leq b_t - C_t x_t\}, \forall t \in \mathcal{T}$$

$$x_{t+} = \mathcal{D}(x_t, u_t, \omega_t) = a_t + A_t x_t + B_t u_t, \forall t \in \mathcal{T}$$

- Value function

$$h_t(x_t) \coloneqq \langle c_t, x_t \rangle + \min \langle d_t, u_t \rangle + \mathbb{E}[h_{t+}(\tilde{x}_{t+})]$$

$$s.t.\ u_t \in \mathcal{U}_t(x_t) \coloneqq \{u_t | D_t u_t \leq b_t - C_t x_t\}$$

- Pre-decision value function (Q-function)

$$f_t(x_t, u_t) \coloneqq \langle c_t, x_t \rangle + \langle d_t, u_t \rangle + \mathbb{E}[h_{t+}(\tilde{x}_{t+})]$$

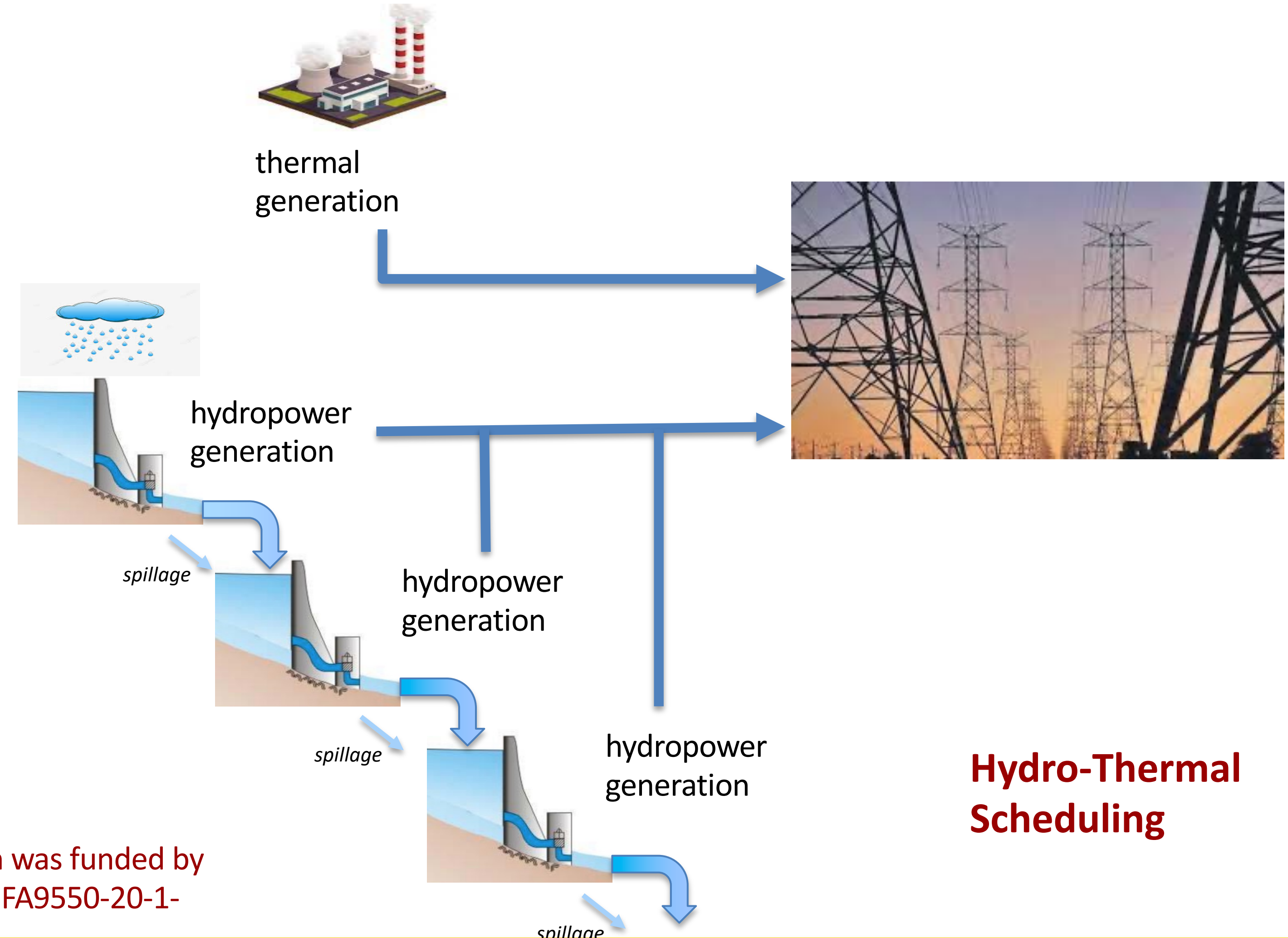
Comments on Experiments Using SDDP

- Asymptotic convergence requires

$$\{N_2, N_3, \ldots, N_T\} \rightarrow \infty,$$

But, of course, we stop in finite time.

- **As in Dynamic Programming, T-stage Compromise Reduces to a Sequence of T-1 Two-Stage Cases**
- SDDP does not use a regularizer, so we only use the generated piecewise linear approximation (PLA).
- SDDP has no cut-pruning ⇒ Large number of pieces per stage

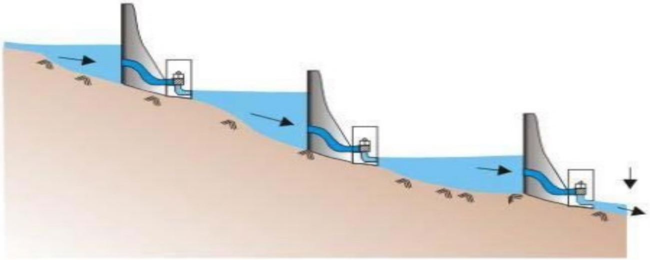


Hydro-Thermal Scheduling

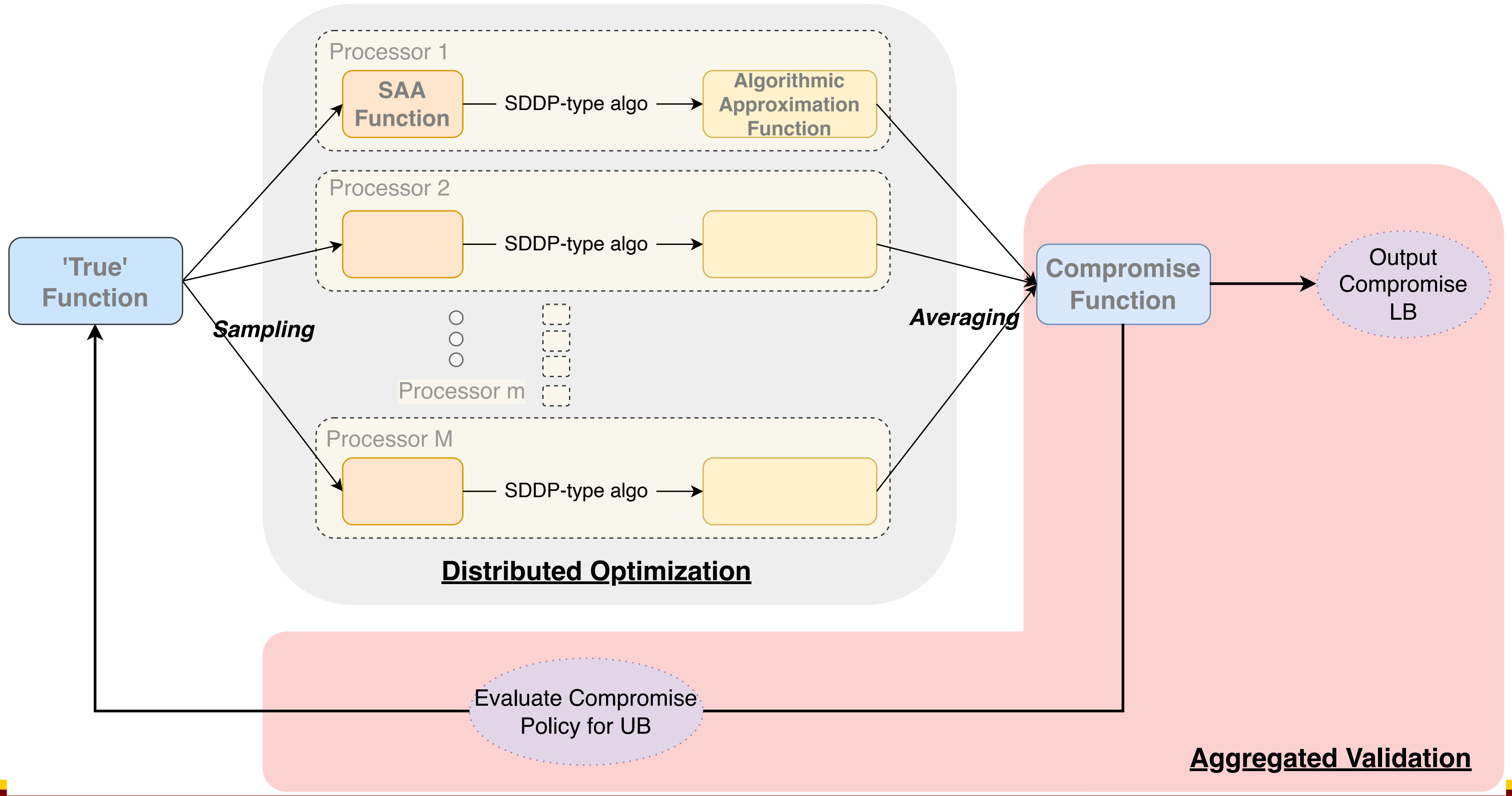
This research was funded by AFOSR grant FA9550-20-1-0006.

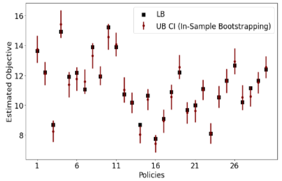
Hydro-Thermal Scheduling
O.Dowson and L. Kapelevich (SDDP.jl)

- The goal is to operate one thermal generator and N hydro generators in a valley chain over τ stages, considering the rainfall uncertainty. In this example stages – 120 (10 years, 12 months per year)
- State variables: the volume in reservoir (hydro generator) i = 1, . . . , N
- **Decision variables:**
  - the power generated by the thermal generator
  - the water from reservoir i = 1, . . . , N used for power generation
  - the water spilling out of reservoir i = 1, . . . , N;
- **Random variables: rainfall**
- The stagewise cost for power generation = thermal generator cost + hydro generators cost
- **Assuming 10 possible realizations for rainfall at any stage, the number of scenarios is 10^{120}**



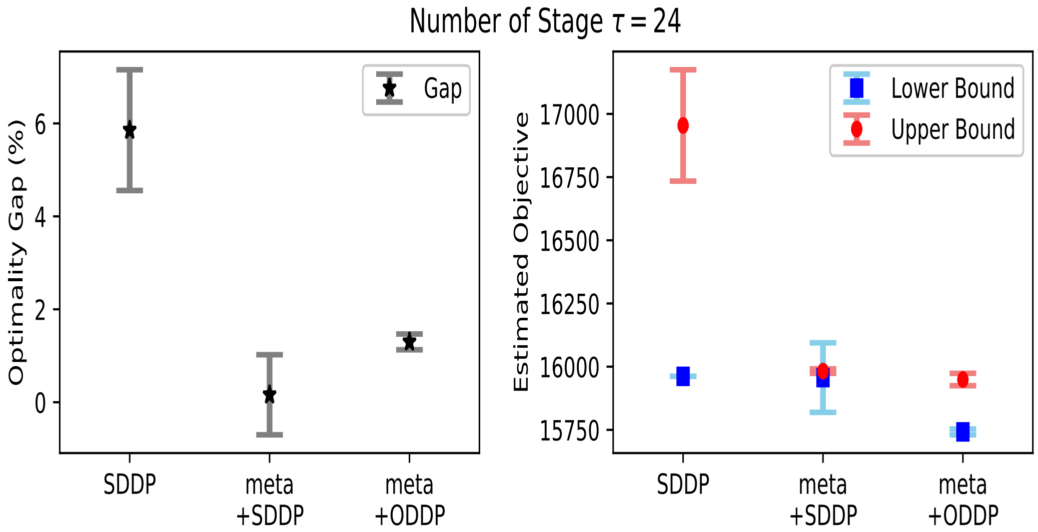
A Meta-Algorithm with Compromise Policy



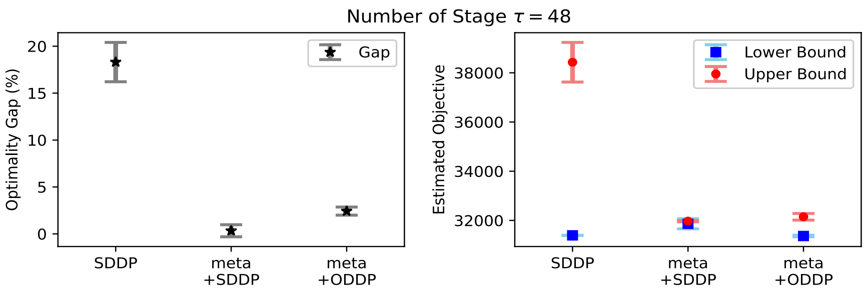


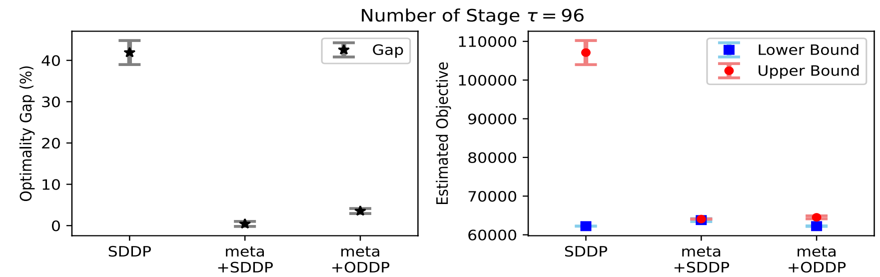
Computational Study

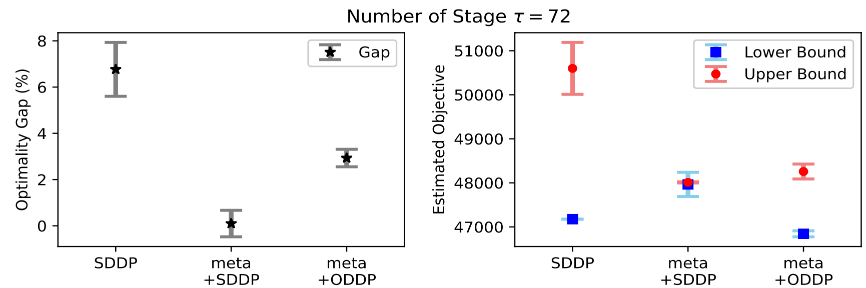
- 4 hydro generators and 1 thermal generator
- **24/48/72/96/120** stages
- Compare SDDP and Our Extensions
  - Meta-Process + SDDP
  - Meta-Process + Incremental Sampling (ODDP)

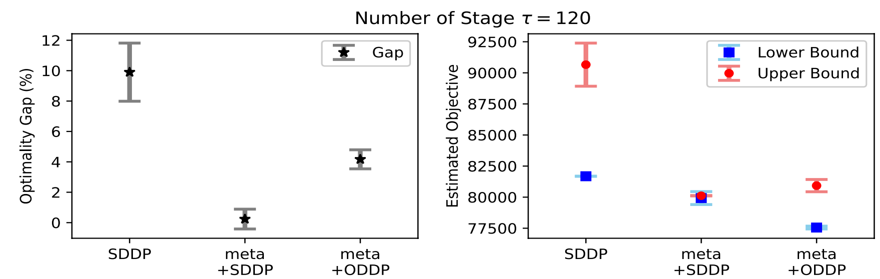


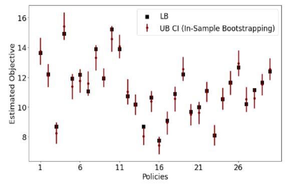
Computational Study - Contd











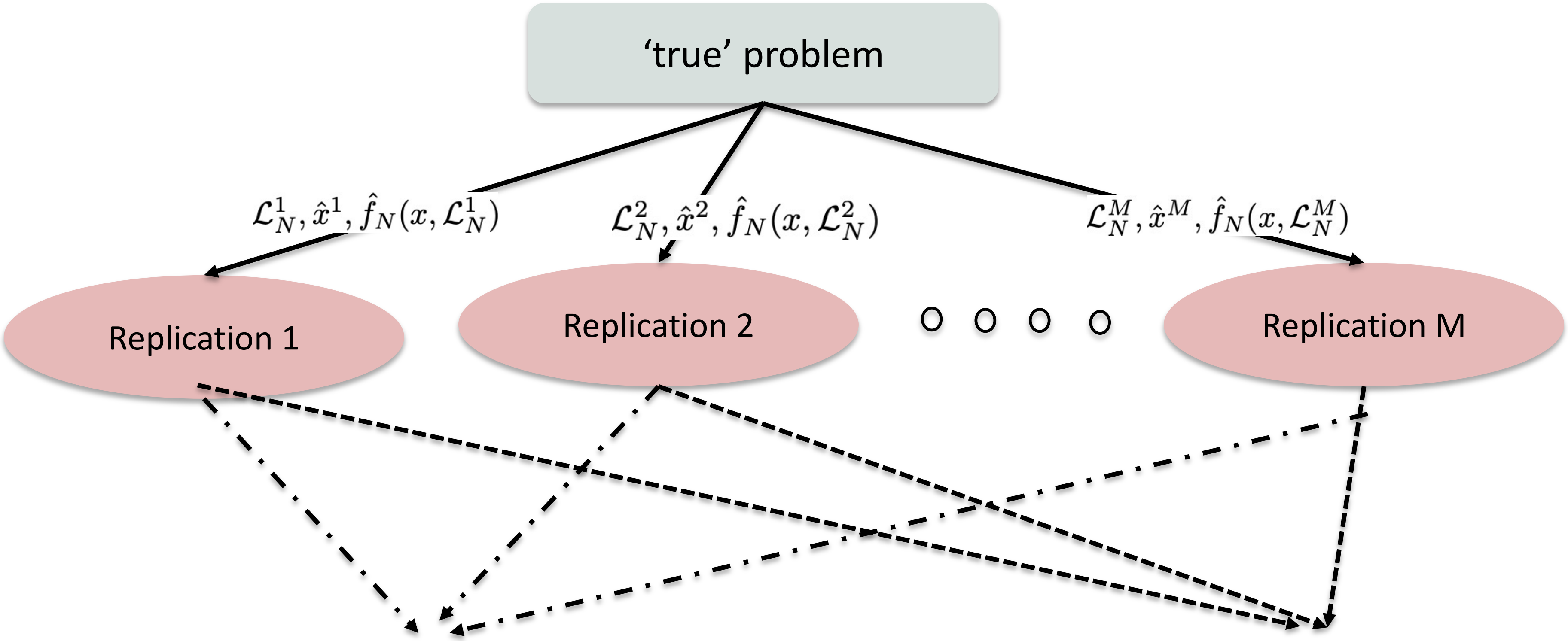
Two-Stage Combinatorial Programming: Stochastic Facility Location Problem

$$\begin{aligned}
\text{Min} \quad & \sum_{i \in P} c_i x_i + \mathbb{E}[h(x, \tilde{\omega})] \\
\text{s.t.} \quad & l \le \sum_{i \in P} x_i \le u \\
& x_i \in \{0,1\}, \quad \forall i \in P
\end{aligned}$$

$$\begin{aligned}
h(x,\omega) = \text{Min} \quad & \sum_{j \in D} \sum_{i \in F} q_j(\omega) d_{ij}(\omega) y_{ij} \\
\text{s.t.} \quad & \sum_{i \in F} y_{ij} = 1, \forall j \in D \\
& y_{ij} \le x_i, \quad \forall i \in P, \forall j \in D \\
& y_{ij} \le 1, \quad \forall i \in O, \forall j \in D \\
& y_{ij} \in \{0,1\}, \quad \forall i \in F, \forall j \in D
\end{aligned}$$

Capacity constraint: $\sum_{j \in D} q_j(\omega) y_{ij} \le k_i x_i, \forall i \in F$

Ensemble Methods for SIP



Solve
In parallel

Kernel Method for SIP: Aggregation in Space of Solution Values

- Solve multiple replications, candidate solutions $\hat{X} = \{\hat{x}^1, ..., \hat{x}^M\}$
- Define kernel function: $k(x, x') = \langle \varphi(x), \varphi(x') \rangle$
- Define Gram Matrix: $K_{ij} := k(\hat{x}^i, \hat{x}^j) = \langle \varphi(\hat{x}^i), \varphi(\hat{x}^j) \rangle$
- Define Centroid: $\bar{\varphi} = \sum_{m=1}^{M} \varphi(\hat{x}^m) / M$
- For any $\hat{x}^m \in \hat{X}$, we have:

  $$\|\varphi(\hat{x}^m) - \bar{\varphi}\|_2^2 = K_{mm} - \frac{2}{M}\sum_{i=1}^{M} K_{im} + \frac{1}{M^2}\sum_{i=1}^{M}\sum_{j=1}^{M} K_{ij}$$
- Compromise Decision: $x^b \in \arg\min_{\hat{x}^m \in \hat{X}} \|\varphi(\hat{x}^m) - \bar{\varphi}\|_2^2$

SFLP Properties

Problem	# of variables in 1st-/2nd-stage	# of Random Elements
P30_O10	30/4720	4720
P45_O15	45/7080	7080
P60_O20	60/9440	9440

Regional Demand Aggregation (Approximation):

- **Geometric Center (Hierarchical logistics)**: geometric center of all demand locations in one region
- **Sampled Center (Tele-communication)**: in each scenario, sample the demand location in one region and find that center
- **Weighted Center (Military logistics)**: : in each scenario, sample the demand locations and the associated quantity to formulate the weighted center

Computations using bagging/compromise solution

Function value estimate at compromise decision

Reduced std compared with ordinary SAA (i.e., one rep)

Model	Problem	$\hat{v}^k$	Std$[\hat{v}^k]$	Reduced Std	Agg. LB 95% CI	UB 95% CI
Geometric	P30_O10	3044.264	2.810	80.75%	[3039.026, 3049.590]	[3040.174, 3049.752]
	P45_O15	2773.942	3.260	78.18%	[2768.673, 2778.726]	[2769.912, 2778.726]
	P60_O20	2617.940	2.479	76.43%	[2614.523, 2621.620]	[2616.929, 2624.861]
Sampled	P30_O10	3053.927	3.429	80.75%	[3046.108, 3059.514]	[3046.062, 3056.142]
	P45_O15	2783.701	4.287	73.27%	[2775.521, 2787.501]	[2786.342, 2795.845]
	P60_O20	2624.868	2.539	74.18%	[2624.726, 2631.597]	[2622.932, 2631.031]
Weighted	P30_O10	3219.671	0.540	76.43%	[3218.843, 3220.621]	[3218.583, 3219.841]
	P45_O15	2779.293	0.465	77.64%	[2778.341, 2779.972]	[2778.210, 2779.428]
	P60_O20	2627.271	0.558	75.75%	[2627.459, 2629.552]	[2627.230, 2628.365]

$I_c = \{m | x^c = \underset{x \in X}{\operatorname{argmin}} \hat{f}_N(x, \mathcal{L}_N^m), m = 1, 2, ..., M\}$. k=$|I_c|$

$\hat{v}^k := \frac{1}{k} \sum_{m \in I_c} f_N(x^c, \mathcal{L}_N^m)$

Computational times for bagging/compromise solution

Model	Problem	Opt. Time (s)	Agg. Time (s)
Geometric	P30_O10	58.554	0.087
	P45_O15	117.342	0.111
	P60_O20	157.823	0.177
Sampled	P30_O10	51.067	0.075
	P45_O15	165.478	0.305
	P60_O20	365.753	0.356
Weighted	P30_O10	193.391	0.480
	P45_O15	178.064	0.345
	P60_O20	164.824	0.246

- \`Opt. Time’ :the time to solve 30 replications sequentially, where each one is solved with Benders Decomposition algorithm.
- \`Agg. Time’: the time for aggregation calculation, which includes the time to find the bagging and compromise solutions and compare whether these two are equal.

Conclusions: If you parallelize SAA

Computational View of Decision/Policy Validation

I. Two-Stage Stochastic Linear Programming
   - Use Compromise Decisions with Prox Term

II. Multi-stage Stochastic Linear Programming
   - Use Compromise Policies with Prox Term

III. Two-Stage Stochastic Combinatorial Programming
   - Use Kernels for Compromise Decisions

Consider a New Slogan
If you Parallelize, Do Compromise

Computational View of Decision/Policy Validation

I. Two-Stage Stochastic Linear Programming
   – Compromise Decisions

II. Multi-stage Stochastic Linear Programming
   – Compromise Policies

III. Two-Stage Stochastic Combinatorial Programming
   – Kernels allow Compromise Decisions