Parallel Scenario Decomposition of Risk Averse 0-1 Stochastic Programs

Shabbir Ahmed
ISyE, Georgia Tech

joint work with
Yan Deng, Siqian Shen (IOE, U of Michigan)

2016 ICSP

Outline

- Risk-Averse Stochastic 0-1 Program
  - Dual representation of coherent risk measure
  - Dual decomposition
  - Distributionally robust counterpart
- Parallelization of Decomposition Method
  - Motivation
  - Parallel Schemes

Risk Averse 0-1 Program

$$\begin{aligned} \min \quad & \rho(f(x,\xi)) \\ \text{s.t.} \quad & x \in X \subseteq \{0,1\}^d \end{aligned}$$

- ξ: a random vector with finite support $\{\xi^1, \ldots, \xi^K\}$ and probabilities $p_1, \ldots, p_K$.

$$p \in \mathcal{A} = \left\{ (p_1, \ldots, p_K) : \sum_{k=1}^{K} p_k = 1,\ p_k \geq 0,\ \forall k = 1, \ldots, K \right\}$$

- $f(x,\xi)$: cost function, e.g.,

$$f(x,\xi) = c^\top x + \min_y \{\theta(y) : y \in Y(x,\xi)\}$$

- $\rho(\cdot)$: coherent risk measure.

Coherent Risk Measure

$$\begin{aligned} \min \quad & \rho(f(x,\xi)) \\ \text{s.t.} \quad & x \in X \subseteq \{0,1\}^d \end{aligned}$$

- Positive homogeneity:

  $$\rho(0) = 0, \text{ and } \rho(\epsilon w) = \epsilon \rho(w) \text{ for any } \epsilon > 0$$

- Sub-additivity:

  $$\rho(w^1 + w^2) \leq \rho(w^1) + \rho(w^2)$$

- Monotonicity:

  $$\rho(w^1 \geq w^2), \text{ if } w^1 \geq w^2 \text{ in all scenarios}$$

- Translation invariance:

  $$\rho(w + C) = \rho(w) + C, \text{ for any constant } C.$$

Coherent Risk Measure

$$\begin{aligned} \min \quad & \rho(f(x,\xi)) \\ \text{s.t.} \quad & x \in X \subseteq \{0,1\}^d \end{aligned}$$

- Artzner et al. (1999), Shapiro and Ahmed (2004), Shapiro (2013): For some uncertainty set $\mathcal{Q}(p) \subseteq \mathcal{A}$,

$$\rho(f(x,\xi)) = \max_{q \in \mathcal{Q}(p)} \left\{ \mathbb{E}_q \left[f(x,\xi) \right] = \sum_{k=1}^{K} q_k f(x, \xi^k) \right\}.$$

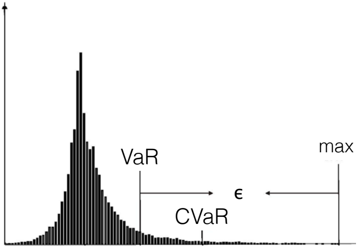
Coherent Risk Measure

$$\begin{aligned} \min \quad & \rho(f(x,\xi)) \\ \text{s.t.} \quad & x \in X \subseteq \{0,1\}^d \end{aligned}$$

- Artzner et al. (1999), Shapiro and Ahmed (2004), Shapiro (2013): For some uncertainty set $\mathcal{Q}(p) \subseteq \mathcal{A}$,

$$\rho(f(x,\xi)) = \max_{q \in \mathcal{Q}(p)} \left\{ \mathbb{E}_q \left[f(x,\xi) \right] = \sum_{k=1}^{K} q_k f(x, \xi^k) \right\}.$$

See, e.g., $\text{CVaR}_{1-\epsilon}(f(x,\xi))$



Coherent Risk Measure

$$\begin{aligned} \min \quad & \rho(f(x,\xi)) \\ \text{s.t.} \quad & x \in X \subseteq \{0,1\}^d \end{aligned}$$

- Artzner et al. (1999), Shapiro and Ahmed (2004), Shapiro (2013): For some uncertainty set $\mathcal{Q}(p) \subseteq \mathcal{A}$,

$$\rho(f(x,\xi)) = \max_{q \in \mathcal{Q}(p)} \left\{ \mathbb{E}_q \left[f(x,\xi) \right] = \sum_{k=1}^{K} q_k f(x,\xi^k) \right\}.$$

See, e.g., $\text{CVaR}_{1-\epsilon}(f(x,\xi))$

$$= \max \left\{ \sum_{k=1}^{K} q_k f(x,\xi^k) : \sum_{k=1}^{K} q_k = 1,\ 0 \le q_k \le p_k/\epsilon,\ \forall k = 1,\ldots,K \right\}$$

Coherent Risk Measure

$$\begin{aligned} \min \quad & \rho(f(x,\xi)) \\ \text{s.t.} \quad & x \in X \subseteq \{0,1\}^d \end{aligned}$$

- Artzner et al. (1999), Shapiro and Ahmed (2004), Shapiro (2013): For some uncertainty set $\mathcal{Q}(p) \subseteq \mathcal{A}$,

$$\rho(f(x,\xi)) = \max_{q \in \mathcal{Q}(p)} \left\{ \mathbb{E}_q \left[f(x,\xi) \right] = \sum_{k=1}^{K} q_k f(x,\xi^k) \right\}.$$

- **Minimax Reformulation**

$$\min_{x \in X} \max_{q \in \mathcal{Q}(p)} \left\{ \sum_{k=1}^{K} q_k f(x,\xi^k) \right\}$$

  - Collado et. al. (2012): risk averse multistage stochastic linear program
  - Ahmed (2013): 0-1 stochastic program
  - Ahmed et. al. (2015): 0-1 chance constrained program

Dual Decomposition

$$\min_{x \in X} \max_{q \in \mathcal{Q}(p)} \left\{ \sum_{k=1}^{K} q_k f(x, \xi^k) \right\}$$

Dual Decomposition

$$\min_{x\in X} \max_{q\in\mathcal{Q}(p)} \left\{ \sum_{k=1}^{K} q_k f(x, \xi^k) \right\}$$

- Clone x for each scenario $\Rightarrow x^1, \ldots, x^K$.
- Force $x^1 = \cdots = x^K$ by non-anticipativity constraint:

$$\sum_{k=1}^{K} \alpha_k x^k = x^1 \qquad \text{(NAC)}$$

where $\alpha_1, \ldots, \alpha_K$ are positive constants that sum to 1.

Dual Decomposition

$$\min_{x^1,\dots,x^K \in X} \max_{q \in \mathcal{Q}(p)} \sum_{k=1}^{K} q_k f(x^k, \xi^k)$$

$$\text{s.t.} \quad \sum_{k=1}^{K} \alpha_k x^k = x^1 \qquad \text{(NAC)}$$

Dual Decomposition

$$\begin{aligned} \min_{x^1,\dots,x^K \in X} \ \max_{q \in \mathcal{Q}(p)} \quad & \sum_{k=1}^{K} q_k f(x^k, \xi^k) \\ \text{s.t.} \quad & \sum_{k=1}^{K} \alpha_k x^k = x^1 \end{aligned} \qquad \text{(NAC)}$$

- Relax (NAC) and punish violation by $\lambda \in \mathbb{R}^d$.

$$\begin{aligned} g(\lambda) &= \min_{x^1,\dots,x^K \in X} \ \max_{q \in \mathcal{Q}(p)} \left\{ \lambda^\top \left(\sum_{k=1}^{K} \alpha_k x^k - x^1 \right) + \sum_{k=1}^{K} q_k f(x^k, \xi^k) \right\} \\ &= \min_{x^1,\dots,x^K \in X} \ \max_{q \in \mathcal{Q}(p)} \left\{ \sum_{k=1}^{K} \Big((\alpha_k - \delta_k) \lambda^\top x^k + q_k f(x^k, \xi^k) \Big) \right\} \end{aligned}$$

where $\delta_1 = 1$ and $\delta_k = 0$ for $k = 2, \dots, K$.

Dual Decomposition

$$\min_{x^1,\dots,x^K\in X}\ \max_{q\in\mathcal{Q}(p)}\ \sum_{k=1}^{K} q_k f(x^k,\xi^k)$$

$$\text{s.t.}\quad \sum_{k=1}^{K} \alpha_k x^k = x^1 \qquad \text{(NAC)}$$

- Relax (NAC) and punish violation by $\lambda \in \mathbb{R}^d$.

$$\begin{aligned} g(\lambda) &= \min_{x^1,\dots,x^K\in X}\ \max_{q\in\mathcal{Q}(p)} \left\{ \lambda^\top \left(\sum_{k=1}^{K} \alpha_k x^k - x^1 \right) + \sum_{k=1}^{K} q_k f(x^k,\xi^k) \right\} \\ &= \min_{x^1,\dots,x^K\in X}\ \max_{q\in\mathcal{Q}(p)} \left\{ \sum_{k=1}^{K} \Big((\alpha_k-\delta_k)\lambda^\top x^k + q_k f(x^k,\xi^k) \Big) \right\} \\ &\geq \max_{q\in\mathcal{Q}(p)}\ \min_{x^1,\dots,x^K\in X} \left\{ \sum_{k=1}^{K} \Big((\alpha_k-\delta_k)\lambda^\top x^k + q_k f(x^k,\xi^k) \Big) \right\} \\ &= \max_{q\in\mathcal{Q}(p)} \left\{ \sum_{k=1}^{K} \min_{x^k\in X} \left\{ (\alpha_k-\delta_k)\lambda^\top x^k + q_k f(x^k,\xi^k) \right\} \right\} = \underline{g}(\lambda) \end{aligned}$$

Dual Decomposition

$$\min_{x^1,\dots,x^K \in X} \ \max_{q \in \mathcal{Q}(p)} \ \sum_{k=1}^{K} q_k f(x^k, \xi^k)$$

$$\text{s.t.} \quad \sum_{k=1}^{K} \alpha_k x^k = x^1 \qquad \text{(NAC)}$$

- Relax (NAC) and punish violation by $\lambda \in \mathbb{R}^d$.

$$\begin{aligned}
g(\lambda) &= \min_{x^1,\dots,x^K \in X} \ \max_{q \in \mathcal{Q}(p)} \left\{ \lambda^\top \left(\sum_{k=1}^{K} \alpha_k x^k - x^1 \right) + \sum_{k=1}^{K} q_k f(x^k, \xi^k) \right\} \\
&= \min_{x^1,\dots,x^K \in X} \ \max_{q \in \mathcal{Q}(p)} \left\{ \sum_{k=1}^{K} \left((\alpha_k - \delta_k) \lambda^\top x^k + q_k f(x^k, \xi^k) \right) \right\} \\
&\geq \max_{q \in \mathcal{Q}(p)} \ \min_{x^1,\dots,x^K \in X} \left\{ \sum_{k=1}^{K} \left((\alpha_k - \delta_k) \lambda^\top x^k + q_k f(x^k, \xi^k) \right) \right\} \\
&= \max_{q \in \mathcal{Q}(p)} \left\{ \sum_{k=1}^{K} \min_{x^k \in X} \left\{ (\alpha_k - \delta_k) \lambda^\top x^k + q_k f(x^k, \xi^k) \right\} \right\} = \underline{g}(\lambda) \qquad \text{(LB, } \forall \lambda \text{)}
\end{aligned}$$

LB Computation

$$\underline{g}(\lambda) = \max_{q \in \mathcal{Q}(p)} \left\{ \sum_{k=1}^{K} \min_{x^k \in X} \left\{ (\alpha_k - \delta_k)\lambda^\top x^k + q_k f(x^k, \xi^k) \right\} \right\}$$

LB Computation

$$\underline{g}(\lambda) = \max_{q \in \mathcal{Q}(p)} \left\{ \sum_{k=1}^{K} \min_{x^k \in X} \left\{ (\alpha_k - \delta_k)\lambda^\top x^k + q_k f(x^k, \xi^k) \right\} \right\}$$

- Approach 1: LB $\leftarrow \underline{g}(0)$.

  $\underline{g}(0) = \max_{q \in \mathcal{Q}(p)} \left\{ \sum_{k=1}^{K} q_k \min_{x \in X} f(x, \xi^k) \right\}$

1: **for** $k = 1, \ldots, K$ **do**
2: $\quad \beta_k \leftarrow \min\{f(x, \xi^k) : x \in X\}$
3: **end for**
4: $\ell \leftarrow \max \left\{ \sum_{k=1}^{K} \beta_k q_k : q \in \mathcal{Q}(p) \right\}$

LB Computation

$$\underline{g}(\lambda) = \max_{q \in \mathcal{Q}(p)} \left\{ \sum_{k=1}^{K} \min_{x^k \in X} \left\{ (\alpha_k - \delta_k)\lambda^\top x^k + q_k f(x^k, \xi^k) \right\} \right\}$$

- Approach 2: LB $\leftarrow \max_\lambda \underline{g}(\lambda)$.

MP: $$\max_{q \in \mathcal{Q}(p), \lambda, \phi} \left\{ \phi : \phi \leq \sum_{k=1}^{K} \min_{x \in X} \left\{ (\alpha_k - \delta_k)\lambda^\top x + q_k f(x, \xi^k) \right\} \right\}$$

1: **repeat**
2: $(\hat{\phi}, \hat{\lambda}, \hat{q}) \leftarrow$ MP
3: **for** $k = 1, \ldots, K$ **do**
4: $\beta_k \leftarrow \min \left\{ (\alpha_k - \delta_k)\hat{\lambda}^\top x + \hat{q}_k f(x, \xi^k) : x \in X \right\}$
5: **end for**
6: add cut $\phi \leq \sum_{k=1}^{K} \left((\alpha_k - \delta_k)\lambda^\top \hat{x}^k + q_k f(\hat{x}^k, \xi^k) \right)$ to MP
7: **until** $\hat{\phi} \leq \sum_{k=1}^{K} \beta_k$

Slow convergence: stop after some iterations and return the best-found $\sum_{k=1}^{K} \beta_k$.

LB Computation

$$\underline{g}(\lambda) = \max_{q \in \mathcal{Q}(p)} \left\{ \sum_{k=1}^{K} \min_{x^k \in X} \left\{ (\alpha_k - \delta_k)\lambda^\top x^k + q_k f(x^k, \xi^k) \right\} \right\}$$

- Approach 1 & 2:

$$\begin{aligned} \min_{x^1,\ldots,x^K \in X} \ \max_{q \in \mathcal{Q}(p)} \quad & \sum_{k=1}^{K} q_k f(x^k, \xi^k) \\ \text{s.t.} \quad & \sum_{k=1}^{K} \alpha_k x^k = x^1 & \sim \lambda \in \mathbb{R}^d \end{aligned}$$

- Approach 3:

$$\begin{aligned} \min_{x^1,\ldots,x^K \in X} \ \max_{q \in \mathcal{Q}(p)} \quad & \sum_{k=1}^{K} q_k f(x^k, \xi^k) \\ \text{s.t.} \quad & \sum_{k=1}^{K} \alpha_k x^k = x^i, \quad \forall i = 1, \ldots, K & \sim q_i \lambda^i \in \mathbb{R}^d \end{aligned}$$

LB Computation

$$\underline{g}(\lambda) = \max_{q \in \mathcal{Q}(p)} \min_{x^1,\ldots,x^K \in X} \Big\{ \sum_{k=1}^{K} q_k \Big(f(x^k, \xi^k) - (\lambda^k)^\top x^k \Big) + \Big(\sum_{k=1}^{K} \alpha_k x^k \Big)^\top \Big(\sum_{k=1}^{K} q_k \lambda^k \Big) \Big\}$$

LB Computation

$$\underline{g}(\lambda) = \max_{q \in \mathcal{Q}(p)} \min_{x^1,\ldots,x^K \in X} \Big\{ \sum_{k=1}^{K} q_k \left(f(x^k, \xi^k) - (\lambda^k)^\top x^k \right) + \left(\sum_{k=1}^{K} \alpha_k x^k \right)^\top \left(\sum_{k=1}^{K} q_k \lambda^k \right) \Big\}$$

$$\underline{\underline{g}}(\lambda) = \max_{q \in \mathcal{Q}(p) \bigcap Q(\lambda)} \Big\{ \sum_{k=1}^{K} q_k \min_{x \in X} \left\{ f(x, \xi^k) - (\lambda^k)^\top x \right\} \Big\},$$

where $Q(\lambda) = \left\{ q : \sum_{k=1}^{K} q_k \lambda^k = 0 \right\}$

- Approach 3: LB $\leftarrow \max_\lambda \underline{\underline{g}}(\lambda)$.

initialize $\lambda^1, \ldots, \lambda^K$
repeat
  for $k = 1, \ldots, K$ **do**
    $\beta_k \leftarrow \min \left\{ f(x, \xi^k) - (\lambda^k)^\top x : x \in X \right\}$
  end for
  $\ell \leftarrow \max \left\{ \sum_{k=1}^{K} \beta_k q_k : q \in \mathcal{Q}(p) \bigcap Q(\lambda) \right\}$
  update $\lambda^1, \ldots, \lambda^K$
until ℓ converges

Slow convergence: stop after some iterations and return the best-found ℓ.

Serial Algorithm

- LB:

	Subproblem of Scenario k
Approach 1	$\min_{x \in X} \{f(x, \xi^k)\}$
Approach 2	$\min_{x \in X} \left\{(\alpha_k - \delta_k)\lambda^\top x + q_k f(x, \xi^k)\right\}$
Approach 3	$\min_{x \in X} \left\{f(x, \xi^k) - (\lambda^k)^\top x\right\}$

- UB: evaluate subproblem solutions.
- Algorithm overview:

```
initialize LB ℓ and UB u
repeat
    compute ℓ and collect subproblem solutions in S, by Approach 1/2/3
    for x̂ ∈ S do
        u ← min{u, ρ(f(x̂, ξ))}
    end for
    X ← X \ S
until u − ℓ ≤ ϵ
```

- No-good Cut to exclude evaluated $\hat{x}$: $\sum_{j:\hat{x}_j=1}(1 - x_j) + \sum_{j:\hat{x}_j=0} x_j \geq 1.$

Distributionally Robust Risk-Averse 0-1 Program

- Known probability distribution p,

$$\min_{x \in X} \rho(f(x, \xi)) = \min_{x \in X} \max_{q \in \mathcal{Q}_\rho(p)} \mathbb{E}_q[f(x, \xi)]$$

- If p is not known exactly, but an uncertainty set U is given,

$$\min_{x \in X} \max_{p \in U} \rho(f(x, \xi))$$

Distributionally Robust Risk-Averse 0-1 Program

- Known probability distribution p,

$$\min_{x \in X} \rho(f(x,\xi)) = \min_{x \in X} \max_{q \in \mathcal{Q}_\rho(p)} \mathbb{E}_q[f(x,\xi)]$$

- If p is not known exactly, but an uncertainty set U is given,

$$\begin{aligned} &\min_{x \in X} \max_{p \in U} \rho(f(x,\xi)) \\ &= \min_{x \in X} \max_{p \in U} \max_{q \in \mathcal{Q}_\rho(p)} \mathbb{E}_q[f(x,\xi)] \\ &= \min_{x \in X} \max_{q \in \{q : q \in \mathcal{Q}_\rho(p),\ p \in \mathcal{P}\}} \mathbb{E}_q[f(x,\xi)] \end{aligned}$$

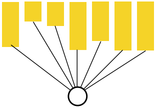
- All the proposed dual decomposition methods are still applicable.

Parallelization

- Parallel jobs, e.g., Sub(k), Eva(x).

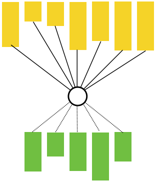
Parallelization

- Parallel jobs, e.g., Sub(k), Eva(x).
- Synchronization and communication in between iterations



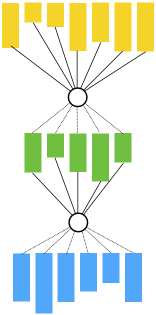
Parallelization

- Parallel jobs, e.g., Sub(k), Eva(x).
- Synchronization and communication in between iterations



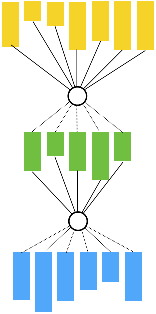
Parallelization

- Parallel jobs, e.g., Sub(k), Eva(x).
- Synchronization and communication in between iterations



Parallelization

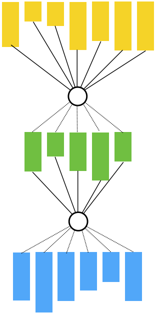
- Parallel jobs, e.g., Sub(k), Eva(x).
- Synchronization and communication in between iterations
- Similarly-structured methods:
  - Dual decomposition [Carøe and Schultz (1999), ...]
  - Benders decomposition [Benders (1962), ...]
  - Progressive hedging [Rockafellar and Roger (1991), ...]
  - Multi-stage decomposition [Slyke and Wets (1969), ...]
  - Scenario decomposition [Higle and Sen (1991), ...]



Existing Work

- Synchronous: barriers after job solving and before reiteration.

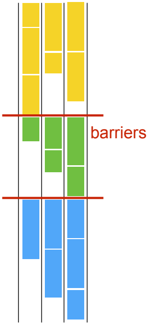
  e.g., Nielsen and Zenios (1997), Ahmed (2013), Lubin et al. (2013), ...



Existing Work

- Synchronous: barriers after job solving and before reiteration.

  e.g., Nielsen and Zenios (1997), Ahmed (2013), Lubin et al. (2013), ...



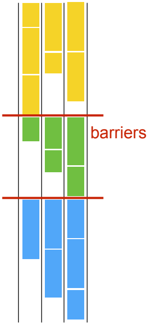
Existing Work

- Synchronous: barriers after job solving and before reiteration.

  e.g., Nielsen and Zenios (1997), Ahmed (2013), Lubin et al. (2013), ...

- Master-Worker: dedicate one processor to collect and compile distributed information.

  e.g., Ruszczyński (1993), Birge et al. (1996), Ryan, et al. (2015), ...

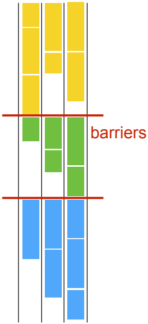


Existing Work

- Synchronous: barriers after job solving and before reiteration.

  e.g., Nielsen and Zenios (1997), Ahmed (2013), Lubin et al. (2013), ...
- Master-Worker: dedicate one processor to collect and compile distributed information.

  e.g., Ruszczyński (1993), Birge et al. (1996), Ryan, et al. (2015), ...
- Dynamic assignment: jobs queue for available processors.



Existing Work

- Synchronous: barriers after job solving and before reiteration.

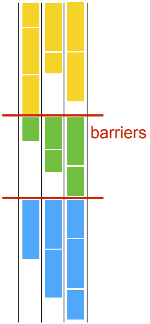
  e.g., Nielsen and Zenios (1997), Ahmed (2013), Lubin et al. (2013), ...

- Master-Worker: dedicate one processor to collect and compile distributed information.

  e.g., Ruszczyński (1993), Birge et al. (1996), Ryan, et al. (2015), ...

- Dynamic assignment: jobs queue for available processors.
- Force reiteration:

  e.g., Linderoth and Wright (2003), ...



Existing Work

- Synchronous: barriers after job solving and before reiteration.

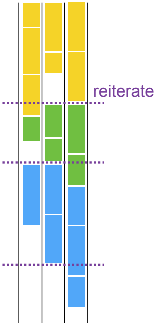
  e.g., Nielsen and Zenios (1997), Ahmed (2013), Lubin et al. (2013), ...

- Master-Worker: dedicate one processor to collect and compile distributed information.

  e.g., Ruszczyński (1993), Birge et al. (1996), Ryan, et al. (2015), ...

- Dynamic assignment: jobs queue for available processors.
- Force reiteration:

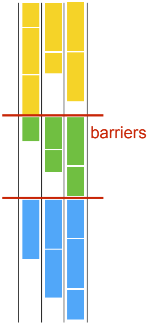
  e.g., Linderoth and Wright (2003), ...



Our Approaches

- Basic Parallel (**BP**): synchronous.

scenario subproblem ⇨ evaluation ⇨ exchange result



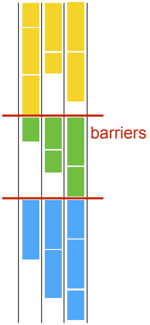
Our Approaches

- Basic Parallel (**BP**): synchronous.

  scenario subproblem ⇨ evaluation ⇨ exchange result

  - Duplicate efforts on evaluation, e.g.,

Processor 1		Processor 2		Processor 3
$\text{Sub}(1) \Rightarrow (0,1,0)$	|	$\text{Sub}(2) \Rightarrow (1,1,1)$	|	$\text{Sub}(3) \Rightarrow (0,1,0)$
$\text{Eva}((0,1,0))$	|	$\text{Eva}((1,1,1))$	|	$\text{Eva}((0,1,0))$



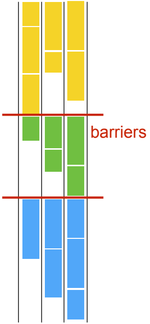
Our Approaches

- Basic Parallel (**BP**): synchronous.

  scenario subproblem ⇨ evaluation ⇨ exchange result

  - Duplicate efforts on evaluation, e.g.,

Processor 1	Processor 2	Processor 3
$\text{Sub}(1) \Rightarrow (0,1,0)$	$\text{Sub}(2) \Rightarrow (1,1,1)$	$\text{Sub}(3) \Rightarrow (0,1,0)$
$\text{Eva}((0,1,0))$	$\text{Eva}((1,1,1))$	$\text{Eva}((0,1,0))$



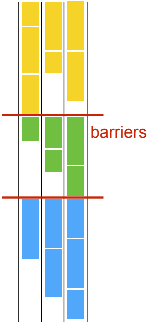
Our Approaches

- Basic Parallel (**BP**): synchronous.

  scenario subproblem ⇨ evaluation ⇨ exchange result

  - Duplicate efforts on evaluation, e.g.,

Processor 1		Processor 2		Processor 3
$\text{Sub}(1) \Rightarrow (0, 1, 0)$	|	$\text{Sub}(2) \Rightarrow (1, 1, 1)$	|	$\text{Sub}(3) \Rightarrow (0, 1, 0)$
$\text{Eva}((0, 1, 0))$	|	$\text{Eva}((1, 1, 1))$	|	$\text{Eva}((0, 1, 0))$



Our Approaches

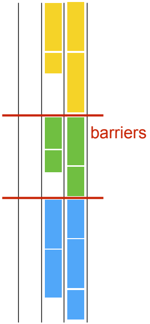
- Basic Parallel (**BP**): synchronous.

  scenario subproblem ⇒ evaluation ⇒ exchange result

  - Duplicate efforts on evaluation, e.g.,

Processor 1		Processor 2		Processor 3
$\text{Sub}(1) \Rightarrow (0,1,0)$	|	$\text{Sub}(2) \Rightarrow (1,1,1)$	|	$\text{Sub}(3) \Rightarrow (0,1,0)$
$\text{Eva}((0,1,0))$	|	$\text{Eva}((1,1,1))$	|	$\text{Eva}((0,1,0))$

- Master-Worker with Barriers (**MWB**): master keep solutions.



Our Approaches

- Basic Parallel (**BP**): synchronous.

  scenario subproblem ⇒ evaluation | ⇒ exchange result

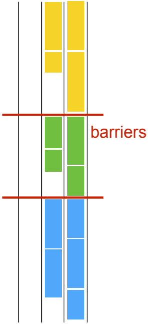
  - Duplicate efforts on evaluation, e.g.,

Processor 1		Processor 2		Processor 3
$\text{Sub}(1) \Rightarrow (0,1,0)$	|	$\text{Sub}(2) \Rightarrow (1,1,1)$	|	$\text{Sub}(3) \Rightarrow (0,1,0)$
$\text{Eva}((0,1,0))$	|	$\text{Eva}((1,1,1))$	|	$\text{Eva}((0,1,0))$

- Master-Worker with Barriers (**MWB**): master keep solutions.

  Worker: scenario subproblem | evaluation

  Master: remove duplicates



Our Approaches

- Basic Parallel (**BP**): synchronous.

  scenario subproblem ⇨ evaluation ⇨ exchange result

  - Duplicate efforts on evaluation, e.g.,

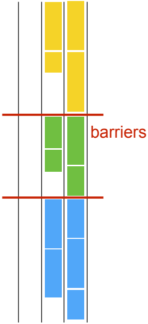
Processor 1		Processor 2		Processor 3
$\text{Sub}(1) \Rightarrow (0,1,0)$	|	$\text{Sub}(2) \Rightarrow (1,1,1)$	|	$\text{Sub}(3) \Rightarrow (0,1,0)$
$\text{Eva}((0,1,0))$	|	$\text{Eva}((1,1,1))$	|	$\text{Eva}((0,1,0))$

- Master-Worker with Barriers (**MWB**): master keep solutions.

  Worker: scenario subproblem | evaluation

  Master: remove duplicates

- Master-Worker without Barriers (**MWN**): master creates jobs and updates every worker individually with results from the others.



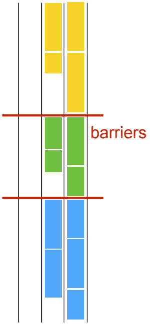
Our Approaches

- Basic Parallel (**BP**): synchronous.

  scenario subproblem ⇨ evaluation ⇨ exchange result

  - Duplicate efforts on evaluation, e.g.,

Processor 1		Processor 2		Processor 3
$\text{Sub}(1) \Rightarrow (0,1,0)$	|	$\text{Sub}(2) \Rightarrow (1,1,1)$	|	$\text{Sub}(3) \Rightarrow (0,1,0)$
$\text{Eva}((0,1,0))$	|	$\text{Eva}((1,1,1))$	|	$\text{Eva}((0,1,0))$

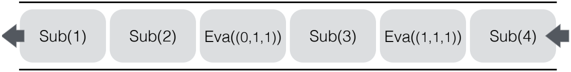


- Master-Worker with Barriers (**MWB**): master keep solutions.

  Worker: scenario subproblem evaluation

  Master: remove duplicates

- Master-Worker without Barriers (**MWN**): master creates jobs and updates every worker individually with results from the others.



Our Approaches

- Basic Parallel (**BP**): synchronous.

  scenario subproblem ⇨ evaluation | ⇨ exchange result

  - Duplicate efforts on evaluation, e.g.,

Processor 1		Processor 2		Processor 3
$\text{Sub}(1) \Rightarrow (0,1,0)$	|	$\text{Sub}(2) \Rightarrow (1,1,1)$	|	$\text{Sub}(3) \Rightarrow (0,1,0)$
$\text{Eva}((0,1,0))$	|	$\text{Eva}((1,1,1))$	|	$\text{Eva}((0,1,0))$

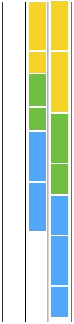
- Master-Worker with Barriers (**MWB**): master keep solutions.

  Worker: scenario subproblem | evaluation

  Master: remove duplicates

- Master-Worker without Barriers (**MWN**): master creates jobs and updates every worker individually with results from the others.

  ← Sub(1) | Sub(2) | Eva((0,1,1)) | Sub(3) | Eva((1,1,1)) | Sub(4) ←



Computational Results

- CPLEX 12.6 & C++ on a Linux workstation with four 3.4GHz processors and 16GB memory.
- Parallel: OpenMPI, Flux HPC Cluster
- Test risk measure ρ: CVaR$_{1-0.1}$
- Instances from SIPLIB†

	SSLP	SMKP
	stochastic server location problem	multi 0-1 knapsack problem
Stage 1	10 binary var 1 constr	240 binary var 50 constr
Stage 2 (per scenario)	500 binary var, 10 continuous var 60 constr	120 binary var 5 constr

	SSLP Instances				SMKP Instances			
	_50	_100	_500	_1000	_1	_2	_3	_4
# scen	50	100	50	1000	20	40	80	160

†: S. Ahmed, R. Garcia, N. Kong, L. Ntaimo, G. Parija, F. Qiu, S. Sen. SIPLIB: A Stochastic Integer Programming Test Problem Library. `http://www.isye.gatech.edu/~sahmed/siplib`, 2015.

Computational Efficiency

- `MIP`: call solver to solve the LP reformulation of CVaR (Rockafellar et al., 2002):

$$\min_{x \in X} \text{CVaR}_\alpha(f(x,\xi)) = \min_{x \in X, \eta} \left\{ \eta + \frac{1}{1-\alpha} \sum_{k=1}^{K} p_k \left[f(x,\xi^k) - \eta \right]^+ : \eta \in \mathbb{R} \right\}.$$

- `DD-`i: dual decomposition using different methods for computing bounds.

Computational Efficiency

- MIP: call solver to solve the LP reformulation of CVaR (Rockafellar et al., 2002):

$$\min_{x\in X} \text{CVaR}_\alpha(f(x,\xi)) = \min_{x\in X,\eta} \left\{ \eta + \frac{1}{1-\alpha} \sum_{k=1}^{K} p_k \left[f(x,\xi^k) - \eta \right]^+ : \eta \in \mathbb{R} \right\}.$$

- DD−i: dual decomposition using different methods for computing bounds.

Table : Solution time in seconds (optimality gap if not solved in 6hrs)

	SSLP				SMKP			
	_50	_100	_500	_1000	_20	_40	_80	_160
MIP	195	201	(100%)	(100%)	299	(0.09%)	(0.11%)	(0.16%)
DD−2S	415	602	7231	(9%)	3496	9080	(0.01%)	(0.01%)
DD−2C	1276	2570	(10%)	(16%)	(0.02%)	(0.01%)	(0.02%)	(0.02%)
DD−1	248	502	4663	12750	2692	9866	11249	18774

#: fastest among the comparison groups.

Computational Efficiency

- MIP: call solver to solve the LP reformulation of CVaR (Rockafellar et al., 2002):

$$\min_{x \in X} \text{CVaR}_\alpha(f(x,\xi)) = \min_{x \in X, \eta} \left\{ \eta + \frac{1}{1-\alpha} \sum_{k=1}^{K} p_k \left[f(x,\xi^k) - \eta \right]^+ : \eta \in \mathbb{R} \right\}.$$

- DD-i: dual decomposition using different methods for computing bounds.

Table : Solution time in seconds (optimality gap if not solved in 6hrs)

	SSLP				SMKP			
	_50	_100	_500	_1000	_20	_40	_80	_160
MIP	195	201	(100%)	(100%)	299	(0.09%)	(0.11%)	(0.16%)
DD-2S	415	602	7231	(9%)	3496	9080	(0.01%)	(0.01%)
DD-2C	1276	2570	(10%)	(16%)	(0.02%)	(0.01%)	(0.02%)	(0.02%)
DD-1	248	502	4663	12750	2692	9866	11249	18774

#: fastest among the comparison groups.

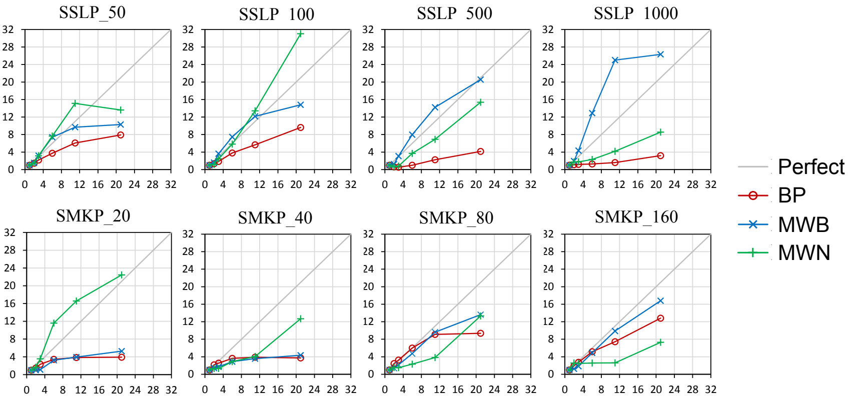
- For modest and large instances, the computational efficacy:

DD-1 (1-loop) > DD-2S (2-loop, subgradient) > DD-2C (2-loop, cutting-plane) > MIP

Parallel DD−1

Speedup = Serial Time / Parallel Time (= # processors, in perfect parallelism)

Figure : Speedup vs. Num of Processes



- MWB and BP crossover.
- MWN (MWB) scales better under a smaller (larger) num of scenarios.
- Super-linear speedup: smaller total workload in parallel than in serial.

Communication Time Tradeoff

- Communication

Communication Time Tradeoff

- Communication
  - Collective vs. Point-to-point

Communication Time Tradeoff

- Communication
  - Collective vs. Point-to-point

  - BP: collective; MWB: mixed; MWN: point-to-point.

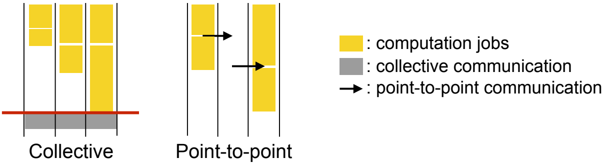
Communication Time Tradeoff

- Communication
  - Collective vs. Point-to-point

  - BP: collective; MWB: mixed; MWN: point-to-point.

Communication Time Tradeoff

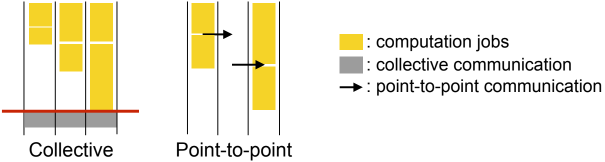
- Communication
  - Collective vs. Point-to-point



  - BP: collective; MWB: mixed; MWN: point-to-point.

Communication Time Tradeoff

- Communication
  - Collective vs. Point-to-point



  - BP: collective; MWB: mixed; MWN: point-to-point.

- Time tradeoff
  - Computation time:

$$\text{BP} > \text{MWB} \approx \text{MWN}$$

  - Collective communication time:

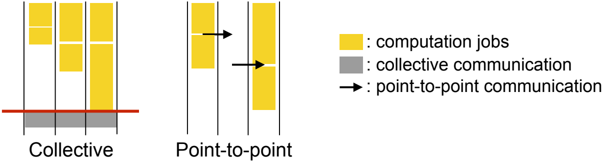
$$\text{BP} > \text{MWB} \gg \text{MWN} = 0$$

  - Point-to-point communication time:

$$\text{MWN} > \text{MWB} \gg \text{BP} = 0$$

Communication Time Tradeoff

- Communication
  - Collective vs. Point-to-point



  - BP: collective; MWB: mixed; MWN: point-to-point.

- Time tradeoff
  - Computation time:

$$\text{BP} > \text{MWB} \approx \text{MWN}$$

  - Collective communication time: $\nearrow$ with num of processors

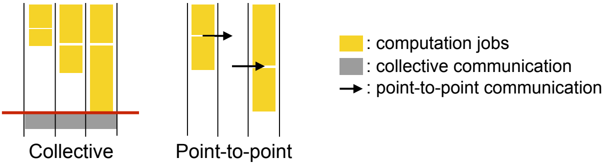
$$\text{BP} > \text{MWB} \gg \text{MWN} = 0$$

  - Point-to-point communication time:

$$\text{MWN} > \text{MWB} \gg \text{BP} = 0$$

Communication Time Tradeoff

- Communication
  - Collective vs. Point-to-point



  - BP: collective; MWB: mixed; MWN: point-to-point.

- Time tradeoff
  - Computation time:

$$\text{BP} > \text{MWB} \approx \text{MWN}$$

  - Collective communication time: $\nearrow$ with num of processors

$$\text{BP} > \text{MWB} \gg \text{MWN} = 0$$

  - Point-to-point communication time: $\nearrow$ with num of scenarios

$$\text{MWN} > \text{MWB} \gg \text{BP} = 0$$

Conclusion

Thank you!
Questions?