Decomposition Algorithm for Optimizing Multi-server Appointment Scheduling with Chance Constraints

Siqian Shen

joint work with Yan Deng

University of Michigan

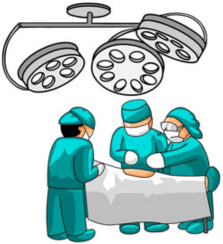
ISyE, Georgia Tech
February 13, 2015

Outline

Applications I

Health care operations management:

1. Appointment scheduling in outpatient clinics
   - How many doctors? The sequence of appointments for each doctor? Time scheduled in between the appointments?
2. Surgery planning in operating rooms (ORs)
   - Which ORs to open? How to allocate surgeries to ORs? How to schedule surgeries in their assigned ORs?



Applications II

High-cost and volatile test scheduling:

1. Crash test scheduling on prototype vehicles
   - How many prototype vehicles to use? How to allocate tests to vehicles? When to start each test?
2. Planning TAs and office hours
   - How many TAs to have? The sequence of office-hour appointments? Time allocation in between the appointments?

General Problem Structure

The multi-server appointment scheduling (MAS) problems

- decide how many/which (costly) servers to open
- allocate and schedule appointments on multiple servers
- involve uncertain service durations

General Problem Structure

The multi-server appointment scheduling (MAS) problems

- decide how many/which (costly) servers to open
- allocate and schedule appointments on multiple servers
- involve uncertain service durations

Challenges:

- Integrated mixed 0-1 planning decisions and larger-scale set of scenarios
- To coordinate staff and resources, need to specify the arrival time of each appt. cannot start before the specified time.
- All planning decisions made before realizing the uncertainty
- Recourse problem: evaluating the undesirable consequences:
  - e.g., server under-utilization, server overtime, appt. delay...
  - complete recourse if minimizing the expected penalty.

Motivation and Goals

Consider the quality of service (QoS):

- use chance constraints to restrict the risk of having overtime servers and appt. delay (given their ambiguous penalty costs)

Motivation and Goals

Consider the quality of service (QoS):

- use chance constraints to restrict the risk of having overtime servers and appt. delay (given their ambiguous penalty costs)

Goals: study the Chance-Constrained Multi-Server Appointment Scheduling (CC-MAS) problem to find out:

- Benefit of integrating allocation and scheduling decisions?
- Benefit of the chance constraints vs. minimizing the expected penalty of server overtime and appt. delay?
- How to compute the non-convex, mixed-integer, stochastic optimization model?

Sketched Model of CC-MAS

- **Decision 1:** opening servers; allocation of jobs to servers
- **Decision 2:** plan start times of jobs on individual servers
- **Objective:** minimize the costs of opening servers and allocating appt. subject to
  - each appointment starts on time
  - a chance constraint requiring the minimum joint probability of all servers finishing on time.

Computing the chance constraints:

- apply the Sample Average Approximation (SAA) method (e.g., Luedtike and Ahmed (2008))
- transform each into a set of big-M constraints with binary logic variables and a cardinality knapsack constraint that restricts values of the logic variables.
- apply **decomposition** for solving the MILP representation.

Literature Review I

Server allocation:

- Blake and Donald (2002), Ozkarahan (2000), Jebali et al. (2006), Denton et al. (2010), Shylo et al. (2012)...

Appointment scheduling under service-time uncertainty:

- Denton and Gupta (2003), Mak et al. (2014), Kong et al. (2014), Jiang and S. (2015)...

Job scheduling:

- Coffiman et al. (1978), Van den Akker et al. (2000), Savelsbergh et al. (2005), Sarin et al. (2014)...

Chance-Constrained Programming:

- Scenario Approximation: Calafiore and Campi (2005), Nemirovski and Shapiro (2006)
- Convex relaxation/approximation: Ahmed (2011), Nemirovski and Shapiro (2007)

Literature Review II

- Efficient point: Sen (1992), Dentcheva et al. (2000), Ruszczyński (2002)

Decomposition for general chance-constrained programs:

- Luedtke et al. (2010), Küçükyavuz (2012): strong valid inequalities for CC with randomness only in RHS
- Luedtke (2013): strong valid inequality and a branch-and-cut algorithm based on scenario decomposition
- Tanner and Ntaimo (2010): no recourse. branch-and-cut based on irreducible infeasible system
- Beraldi and Bruni (2010): specialized branch-and-bound
- Qiu et al. (2014), Song et al. (2014): strengthening big-M coefficients in the extended formulation
- Watson et al. (2010), Ahmed et al. (2014): dual decomposition

Parameters of CC-MAS

- I: a set of appointments.
- J: a set of servers.
- T_j: operating time limit of server $j \in J$.
- c^1_j: cost of operating server j.
- c^2_{ij}: cost of assigning appointment i to server j.
- $[\underline{a}_i, \overline{a}_i]$: earliest and latest time to start appointment i.
- W_i: maximum allowable delay time of appointment i.
- ξ_i: random service durations of appointment i.
- Ω: a discrete and finite support of the random service time ξ_i.
- $\xi^\omega = [\xi^\omega_i, \ i \in I]^\mathsf{T}$ is a realization in scenario $\omega \in \Omega$.

Decisions in CC-MAS

Binary Variables:

- x_j (open server): for $j \in J$, $x_j = 1$ if server j opens, and 0 o.w.
- y_{ij} (allocation): for $j \in J$ and $i \in I$, $y_{ij} = 1$ if appt. i is allocated to server j, and 0 o.w.
- $z_{i'i}$ (sequence): for any $i, i' \in I,\ i \neq i'$, $z_{i'i} = 1$ if appt. i' is scheduled ahead of i, and 0 o.w.

Continuous Variables:

- planned arrival time of appointments: $s_i \geq 0,\ \forall i \in I$
- actual start time of appointments: $t_i^w,\ \forall i \in I,\ w \in \Omega$

Formulation of CC-MAS I

$$\min \quad \sum_{j\in J} c_j^1 x_j + \sum_{i\in I}\sum_{j\in J} c_{ij}^2 y_{ij} \tag{1}$$

$$\text{s.t.} \quad (x, y, z, s) \in Q \tag{2}$$

$$\mathbb{P}\Big\{(x, y, z, s) \in \mathcal{Q}(\xi)\Big\} \geq 1 - \epsilon. \tag{3}$$

- Q is a fixed region, given by MILP constraints in x, y, z, s.
- $\mathcal{Q}(\xi)$ is a region parameterized by the uncertain vector ξ.

Formulation of CC-MAS II

Mixed 0-1 integer deterministic set:

$$Q = \Big\{ (x, y, z, s) \in \{0,1\}^{|J|} \times \{0,1\}^{|I|\times|J|} \times \{0,1\}^{|I|\times(|I|-1)} \times \mathbb{R}_+^{|I|} :$$
$$\sum_{j\in J} y_{ij} = 1,\ y_{ij} \le x_j \quad \forall i \in I,\ j \in J$$
$$y_{ij} + y_{i'j} - 1 \le z_{ii'} + z_{i'i} \le 1,$$
$$1 - z_{ii'} \ge y_{ij} - y_{i'j},\ 1 - z_{ii'} \ge y_{i'j} - y_{ij}, \quad \forall i, i' \in I,\ i \ne i',\ j \in J$$
$$\underline{a}_i \le s_i \le \overline{a}_i \quad \forall i \in I$$
$$s_i \ge -\mathcal{M}^1_{i'i}(1 - z_{i'i}) + s_{i'} \quad \forall i, i' \in I,\ i \ne i' \Big\}. \tag{4}$$

Formulation of CC-MAS III

$\forall w \in \Omega$:

$$\mathcal{Q}(\xi^w) = \Big\{ (x, y, z, s) : \exists t^w \in \mathbb{R}_+^{|I|} \text{ such that}$$

$$t_i^w \geq s_i, \quad \forall i \in I.$$

$$t_i^w \geq -\mathcal{M}^2_{i'iw}(1 - z_{i'i}) + t_{i'}^w + \xi_{i'}^w \quad \forall i, i' \in I,\ i \neq i'.$$

$$t_i^w + \xi_i^w \leq T_j + \mathcal{M}^3_{ijw}(1 - y_{ij}) \quad \forall i \in I,\ j \in J \Big\},$$

In the rest of the talk, we replace the joint chance constraint (3) by

$$\sum_{w \in \Omega} \mathbb{I}\left\{(x, y, z, s) \in \mathcal{Q}(\xi^w)\right\} \geq |\Omega| - \theta$$

- $\mathbb{I}\{\cdot\}$ is an indicator function; $\theta = \lfloor \epsilon |\Omega| \rfloor$.
- It can lead to the extended MIP reformulation; or we use it to evaluate the chance of a given solution $(\hat{x}, \hat{y}, \hat{z}, \hat{s})$ satisfying all constraints in $\mathcal{Q}(\xi)$.

Outline

Separate Allocation & Scheduling

1st-stage (allocation):

$$\min \left\{ c^1 x + c^2 y : \sum_{j \in J} y_{ij} = 1,\ y_{ij} \le x_j,\ (x, y) \in \mathcal{A} \cap \{0,1\}^{|J|} \times \{0,1\}^{|I| \times |J|} \right\}$$

where $\mathcal{A} = \{(x, y) :$
$\exists s, z$ satisfying other constraints in Q and the chance constraint (3).$\}$.

2nd-stage (scheduling): given $(\hat{x}, \hat{y})$, check whether $(\hat{x}, \hat{y}) \in \mathcal{A}$ by finding a feasible (z, s, t) to constraints in $\mathcal{A}$ with $y = \hat{y}$.

- If such a solution exists, $(\hat{x}, \hat{y})$ is optimal.
- Otherwise, add a cut to the 1st-stage allocation problem, e.g., no-good cuts for binary valued (x, y).

Separate Allocation & Scheduling

1st-stage (allocation):

$$\min\left\{c^1x+c^2y:\ \sum_{j\in J} y_{ij}=1,\ y_{ij}\le x_j,\ (x,y)\in\mathcal{A}\cap\{0,1\}^{|J|}\times\{0,1\}^{|I|\times|J|}\right\}$$

where $\mathcal{A}=\{(x,y):$
$\exists s,z$ satisfying other constraints in Q and the chance constraint (3).$\}$.

2nd-stage (scheduling): given $(\hat{x},\hat{y})$, check whether $(\hat{x},\hat{y})\in\mathcal{A}$ by finding a feasible (z,s,t) to constraints in $\mathcal{A}$ with $y=\hat{y}$.

- If such a solution exists, $(\hat{x},\hat{y})$ is optimal.
- Otherwise, add a cut to the 1st-stage allocation problem, e.g., no-good cuts for binary valued (x,y).

Problem: Finding a feasible schedule is hard; not much information about feasibility is known when solving the 1st-stage.

Our Approaches I

Enhancement 1: Add a proxy of the joint chance constraint to the 1st-stage problem:

$$\sum_{w \in \Omega} \mathbb{I}\left\{\sum_{i \in I} \xi_i^w y_{ij} \leq T_j x_j \ \forall j \in J\right\} \geq |\Omega| - \theta \qquad (5)$$

Enhancement 2: For a given $(\hat{x}, \hat{y})$, consider

- set $J(\hat{x}) = \{j \in J : \hat{x}_j = 1\}$ of operating servers;
- sets $I_j(\hat{y}) = \{i \in I : \hat{y}_{ij} = 1\}$ of appointments allocated on each server $j \in J(\hat{x})$.

Define variables:

- $u_{ik} \in \{0, 1\}, \ \forall i \in I_j(\hat{y})$ and $k = 1, \ldots, |I_j(\hat{y})|$, such that $u_{ik} = 1$ if appt. i is scheduled as the k^{th} one, and $u_{ik} = 0$ o.w.
- $r_k \geq 0$ and $\gamma_k \geq 0$ representing the appointed start time and the actual start time of the k^{th} appt. respectively, $\forall k = 1, \ldots, |I_j(\hat{y})|$.

Our Approaches II

The 2nd-stage feasible set $\mathcal{A}$ is equivalent to:

$$\sum_{k=1}^{|I_j(\hat{y})|} u_{ik} = 1 \quad \forall i \in I_j(\hat{y})$$

$$\sum_{i \in I_j(\hat{y})} \underline{a}_i u_{ik} \leq r_k \leq \sum_{i \in I_j(\hat{y})} \overline{a}_i u_{ik} \quad \forall k = 1, \ldots, |I_j(\hat{y})|$$

$$r_k - r_{k-1} \geq 0 \quad \forall k = 2, \ldots, |I_j(\hat{y})|$$

$$\gamma_k^w \geq r_k \quad \forall k = 1, \ldots, |I_j(\hat{y})|, \ \forall w \in \Omega$$

$$\gamma_k^w \geq \gamma_{k-1}^w + \sum_{i \in I_j(\hat{y})} \xi_i^w u_{ik-1} \quad \forall k = 2, \ldots, |I_j(\hat{y})|, \ \forall w \in \Omega$$

$$\gamma_{|I_j(\hat{y})|}^w + \sum_{i \in I_j(\hat{y})} \xi_i^w u_{i|I_j(\hat{y})|} \leq T_j, \ \forall w \in \Omega$$

$$u_{ik} \in \{0, 1\}, \forall i \in I_j(\hat{y}), \ r_k \geq 0, \ \gamma_k \geq 0, k = 1, \ldots, |I_j(\hat{y})|.$$

This reformulation does not contain the big-$\mathcal{M}^1$, -$\mathcal{M}^2$ and -$\mathcal{M}^3$ coefficients.

Outline

Strengthened Big-M Coefficients

To optimize the enhanced 1st-stage allocation problem with the added joint chance constraint (5), we work with the extended reformulation.

Strengthen the big-M coefficients using two approaches:

- Qiu et al. (2014): iteratively repeat plugging the latest-attained coefficients into an LP model to compute improved values.
- Song et al. (2014): sort scenario-based optimal objectives (much easier to compute) to derive valid coefficient thresholds.

Other Approaches for Optimizing the 1st Stage

1. Branch-and-Cut (Luedtke (2013)): Strengthen the big-M valid inequalities in Song et al. (2014) by lifting, and integrate into a branch-and-cut algorithm.

Other Approaches for Optimizing the 1st Stage

1. Branch-and-Cut (Luedtke (2013)): Strengthen the big-M valid inequalities in Song et al. (2014) by lifting, and integrate into a branch-and-cut algorithm.
2. Decomposition-based bounding: consider scenario-based subproblems:

$$v(w,\mathcal{S}) = \min\left\{c^1 x + c^2 y : (x,y) \in \mathcal{D}_w \setminus \mathcal{S}\right\} \quad \forall w \in \Omega \qquad (6)$$

where $\mathcal{S}$ is a set of (x, y) vertices violating the joint chance constraint (5). For a fixed $\mathcal{S}$, we compute $v(w,\mathcal{S})$, $\forall w \in \Omega$ to update valid upper bound $\overline{B}$ (any $v(w,\mathcal{S})$ yielding feasible $(x(w), y(w))$) and lower bound $\underline{B}$ ($= v(\sigma_{\theta+1},\mathcal{S})$ as the $\theta + 1$ largest value). We append evaluated solutions to the set $\mathcal{S}$ and add no-good cuts for excluding the corresponding (x, y).

Other Approaches for Optimizing the 1st Stage

1. Branch-and-Cut (Luedtke (2013)): Strengthen the big-M valid inequalities in Song et al. (2014) by lifting, and integrate into a branch-and-cut algorithm.

2. Decomposition-based bounding: consider scenario-based subproblems:

$$v(w, \mathcal{S}) = \min \left\{ c^1 x + c^2 y : (x, y) \in \mathcal{D}_w \setminus \mathcal{S} \right\} \quad \forall w \in \Omega \qquad (6)$$

   where $\mathcal{S}$ is a set of (x, y) vertices violating the joint chance constraint (5). For a fixed $\mathcal{S}$, we compute $v(w, \mathcal{S})$, $\forall w \in \Omega$ to update valid upper bound $\overline{B}$ (any $v(w, \mathcal{S})$ yielding feasible $(x(w), y(w))$) and lower bound $\underline{B}$ ($= v(\sigma_{\theta+1}, \mathcal{S})$ as the $\theta + 1$ largest value). We append evaluated solutions to the set $\mathcal{S}$ and add no-good cuts for excluding the corresponding (x, y).

3. Dual/scenario decomposition: make copies of x and y in all scenarios and enforce them taking the same values by using nonanticipativity constraints. Take the Lagrangian dual and optimize.

Outline

2nd Stage: Chance-Constrained Appointment Scheduling

Given $(\hat{x}, \hat{y})$ from the 1st stage, we verify whether exists feasible appt. arrivals to satisfy the server-overtime chance constraint.

- It is an MIP with a joint chance constraint.
- We can still apply the previous approaches used for solving the enhanced 1st-stage problem.
- All constraints are "server decomposable" except the joint chance constraint of server overtime.
- We use branch-and-cut and add cuts based on "scenario covers" (i.e., "cover inequalities" by identifying scenarios that cannot be *all* violated.)
- We identify the scenario covers based on irreducibly infeasible subsystem (IIS) of an LP relaxation model.
- The idea was also implemented by Tanner and Ntaimo (2010) and Codato and Fischetti (2006) in different contexts.

Outline

Model Variants of CC-MAS

We consider the following model variants and most computational methods can be generalized:

- Replace the joint chance constraint (3) by multiple chance constraints each for one server:

$$\sum_{w\in\Omega} \mathbb{I}\{(x,y,z,s)\in\mathcal{Q}_j(\xi^w)\} \geq |\Omega| - \lfloor \epsilon_j|\Omega| \rfloor.$$

  The 2nd-stage problem becomes server-wise decomposable.

Model Variants of CC-MAS

We consider the following model variants and most computational methods can be generalized:

- Replace the joint chance constraint (3) by multiple chance constraints each for one server:

$$\sum_{w\in\Omega} \mathbb{I}\left\{(x,y,z,s) \in \mathcal{Q}_j(\xi^w)\right\} \geq |\Omega| - \lfloor \epsilon_j |\Omega| \rfloor.$$

  The 2nd-stage problem becomes server-wise decomposable.

- **Hard constraints on appointment waiting**: $t_i^w - s_i \leq \mathrm{W}_i$, for all $i \in I,\ w \in \Omega$.

Model Variants of CC-MAS

We consider the following model variants and most computational methods can be generalized:

- Replace the joint chance constraint (3) by multiple chance constraints each for one server:

  $$\sum_{w\in\Omega} \mathbb{I}\left\{(x,y,z,s) \in \mathcal{Q}_j(\xi^w)\right\} \geq |\Omega| - \lfloor \epsilon_j |\Omega| \rfloor.$$

  The 2nd-stage problem becomes server-wise decomposable.

- **Hard constraints on appointment waiting**: $t_i^w - s_i \leq \mathrm{W}_i$, for all $i \in I,\ w \in \Omega$.

- **Recourse Cost in the Objective**: Define a variables $o_j^w \in \mathbb{R}_+$ as the overtime of every server j in each scenario $w \Rightarrow$ $c^1x + c^2y + (1/|\Omega|)\sum_{w\in\Omega}\sum_{j\in J} c_j^3 o_j^w$, and add constraints $o_j^w \geq t_i^w + \xi_i^w - \mathrm{T}_j - \mathcal{M}^3_{ijw}(1-y_{ij}),\ \forall i \in I,\ j \in J,\ w \in \Omega$.

Model Variants of CC-MAS

We consider the following model variants and most computational methods can be generalized:

- Replace the joint chance constraint (3) by multiple chance constraints each for one server:

$$\sum_{w\in\Omega} \mathbb{I}\{(x,y,z,s)\in\mathcal{Q}_j(\xi^w)\} \geq |\Omega| - \lfloor \epsilon_j |\Omega| \rfloor.$$

  The 2nd-stage problem becomes server-wise decomposable.

- **Hard constraints on appointment waiting**: $t_i^w - s_i \leq \mathrm{W}_i$, for all $i \in I,\ w \in \Omega$.

- **Recourse Cost in the Objective**: Define a variables $o_j^w \in \mathbb{R}_+$ as the overtime of every server j in each scenario $w \Rightarrow$ $c^1x + c^2y + (1/|\Omega|)\sum_{w\in\Omega}\sum_{j\in J} c_j^3 o_j^w$, and add constraints $o_j^w \geq t_i^w + \xi_i^w - \mathrm{T}_j - \mathcal{M}_{ijw}^3(1-y_{ij}),\ \forall i \in I,\ j \in J,\ w \in \Omega$.

- The delay of appointments can be penalized in a similar way.

Computational Setup

Problem instances: allocating and scheduling surgeries to operating rooms (ORs) under surgery time uncertainty.

Children's Hospital of Michigan
DMC DETROIT MEDICAL CENTER

ORs (servers):

- $\mathrm{T}_j = 4 \sim 15,\ j \in J$; $c_j^1 = 8 \sim 18,\ j \in J$; $c_{ij}^2 = 1,\ \forall i \in I,\ j \in J$.

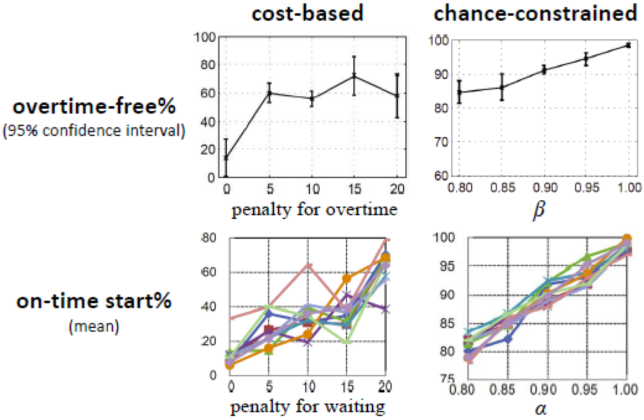
Surgeries (appointments):

- durations of the operating time of each surgery type are randomly sampled based on one-week data, following a lognormal distribution.
- $[\underline{a}_i, \overline{a}_i]$: [0, 6], [6, 12], and [0, 12].
- $\epsilon = 0.1$

Computer characteristics:

- CPU 3.20 GHz, with 8GB memory; CPLEX 12.5.1.

1. Cost-Based Model:



2. Separate Modeling: **10%↓** overtime-free%.

Results of Integrating Allocation and Scheduling

A benchmark process: CCSA $\rightarrow$ CCS:
Chance-constrained server allocation (CCSA)

- a stochastic bin packing problem where we "pack" surgeries with random durations into ORs with time limits, subject to a joint chance constraint of β on-time OR closure rate.

Chance-constrained scheduling (CCS)

- Pass an optimal solution of CCSA to CCS, where we seek feasible schedules to satisfy the two chance constraints in CC-MAS.

Results of Integrating Allocation and Scheduling

A benchmark process: CCSA $\rightarrow$ CCS:
Chance-constrained server allocation (CCSA)

- a stochastic bin packing problem where we "pack" surgeries with random durations into ORs with time limits, subject to a joint chance constraint of β on-time OR closure rate.

Chance-constrained scheduling (CCS)

- Pass an optimal solution of CCSA to CCS, where we seek feasible schedules to satisfy the two chance constraints in CC-MAS.

Table: Results of QoS level β' and cost of "integrating" and "separating" CC-MAS models.

Model used	β'(%) given β(%):					solution cost given β(%):				
	80	85	90	95	100	80	85	90	95	100
CC-MAS	87.7±1.3	89.3±1.5	91.2±1.3	94.3±1.5	96.9±1.5	38.0±0.1	38.3±0.8	38.3±0.8	41.1±1.4	44.1±1.7
CCSA→CCS	75.5±2.0	79.7±1.6	79.6±1.1	80.1±2.1	85.8±2.8	38.0±0.0	38.0±0.0	38.0±0.1	38.3±0.8	40.4±1.1

CPU Time Results of Decomposition I

Table: Total solution time and number of branched nodes

Instance	$\|\Omega\|$	Direct		MP+SP			MP*+SP		
		total	#node	total	#node	#cut	total	#node	#cut
$\|J\| = 5$	20	269.8	169356	366.2	23	8	1.3	421	3
$\|I\| = 10$	200	-	5333*	4854.3	14320	64	54.6	6503	3
	2000	-	29*	-	879*	87*	-	13199*	5*

Table: Solution time for solving MP* and big-M strengthening

Instance	$\|\Omega\|$	MP*+SP	$\text{MP}^*_{\text{iter}}$+SP			$\text{MP}^*_{\text{scen}}$+SP		
		mp (sec)	mp (sec)	str (sec)	str%	mp (sec)	str (sec)	str%
$\|J\| = 5$	20	0.4	0.3	1.1	15.4%	0.1	7.9	1.0%
$\|I\| = 10$	200	48.7	11.4	20.4	16.0%	1.5	725.4	1.1%
	2000	-	15.8	1917.7	16.0%	5.3	5325.0	1.0%

CPU Time Results of Decomposition II

Table: Comparisons of B&C, scenario-based bounding, dual decomposition for the enhanced 1st-stage problem (MP*)

Instance	$\lvert\Omega\rvert$	B&C+SP			Pbnd+SP			Dbnd+SP		
		total (sec)	#sub	sub (sec)	total (sec)	#sub	sub (sec)	total (sec)	#sub	sub (sec)
$\lvert J\rvert=5$	20	2.5	35	0.06	3.2	146	0.02	1.6	248	0.008
$\lvert I\rvert=10$	200	173.2	388	0.44	10.0	568	0.02	13.8	2480	0.007
	2000	2494.9	3754	0.66	78.5	2100	0.03	185.2	29500	0.007
$\lvert J\rvert=10$	20	6535.8	2672	2.40	115.3	751	0.12	46.8	656	0.07
$\lvert I\rvert=20$	200	-	-	-	410.5	1136	0.16	347.8	2080	0.09
	2000	-	-	-	1332.4	4000	0.13	1053.4	8324	0.10

Table: Solution time on directly computing the 2nd-stage problem (SP)

Instance	$\lvert\Omega\rvert$	MP^*_{iter}+SP		B&C+SP		Pbnd+SP		Dbnd+SP	
		sp (sec)	sp%	sp (sec)	sp%	sp (sec)	sp%	sp (sec)	sp%
$\lvert J\rvert=5$	20	0.3	18.3%	0.4	16.0%	0.8	12.8%	0.1	3.8%
$\lvert I\rvert=10$	200	3.4	9.8%	0.2	0.1%	0.2	4.5%	0.1	2.5%
	2000	42.0	4.0%	27.7	1.1%	2.4	3.7%	1.6	2.1%
$\lvert J\rvert=10$	20	4.1	6.7%	23.0	0.3%	0.1	0.1%	0.9	3.8%
$\lvert I\rvert=20$	200	-	-	-	-	2.4	1.3%	13.8	3.5%
	2000	-	-	-	-	25.3	1.9%	22.1	2.1%

CPU Time Results of Decomposition III

Table: IIS-based scenario cover inequalities for solving SP

Instance	$\|\Omega\|$	MP$^*_{\text{iter}}$+B&C'		B&C+B&C'		Pbnd+B&C'		Dbnd+B&C'	
		sp (sec)	sp%	sp (sec)	sp%	sp (sec)	sp%	sp (sec)	sp%
$\|J\| = 5$	20	3.4	70.3%	0.3	9.9%	0.5	7.9%	4.3	84.0%
$\|I\| = 10$	200	13.3	27.9%	3.9	1.7%	3.4	37.2%	2.3	26.0%
	2000	39.7	3.6%	21.6	0.8%	11.7	14.5%	9.4	9.9%
$\|J\| = 10$	20	27.0	30.5%	27.1	0.3%	12.3	54.3%	2.0	7.3%
$\|I\| = 20$	200	-	-	-	-	14.0	6.4%	5.7	1.4%
	2000	-	-	-	-	13.2	0.9%	12.1	1.1%

Table: The CC-MAS variant with overtime penalty cost

Instance	$\|\Omega\|$	Pbnd+SP		Dbnd+SP		Pbnd+B&C'		Dbnd+B&C'	
		mp (sec)	sp (sec)	mp (sec)	sp (sec)	mp (sec)	sp (sec)	mp (sec)	sp (sec)
$\|J\| = 5$	20	32.4	20.3	29.7	1.5	18.9	34.5	65.5	35.7
$\|I\| = 10$	200	50.7	248.6	110.6	449.3	64.3	104.8	47.5	76.3
penal.	2000	361.1	1242.7	249.1	3032.6	130.9	523.8	117.8	700.7
$\|J\| = 10$	20	369.8	198.5	248.4	104.8	465.3	127.4	388.1	333.0
$\|I\| = 20$	200	842.3	2036.1	502.3	3571.7	535.7	369.6	476.1	629.2
penal.	2000	1567.5	5413.7	1098.4	2499.0	795.4	749.9	731.5	166.9

CPU Time Results of Decomposition IV

Table: Multiple chance constraints vs. joint chance constraint

Instance	$\lvert\Omega\rvert$	$\texttt{MP}^{*}_{\text{iter}}\texttt{+SP (MCC)}$		$\texttt{MP}^{*}_{\text{iter}}\texttt{+SP}$	
		total	#node	total	#node
$\lvert J\rvert = 5$	20	0.3	311	1.3	421
$\lvert I\rvert = 10$	200	31.9	751	54.6	6503
	2000	4474.5	9701	-	13199*

Conclusions

- Combine multiple server/scenario-based decomposition methods for solving CC-MAS.
- The work can be generalized to problems with decomposable structures, e.g., network problems with multiple subgraphs and correlated network-flow decisions.
- The decomposition framework is also not restricted to problems with joint chance constraints.

Future research:

- Incorporate other risk measures.
- Apply to prototype vehicle test scheduling (under collaboration with Ford Motor Company).
- Introduce distribution ambiguity. Consider multiple uncertainty sources.

Thank you!

Questions?