MULTISTAGE ROBUST MIXED INTEGER OPTIMIZATION WITH ADAPTIVE PARTITIONS

15.094J: Robust Modeling, Optimization, Computation

Iain Dunning

March 4, 2015

MIT Operations Research Center

OUTLINE

1. The problem & past work
2. Motivation
3. Algorithm
4. Bounds & extension to multistage
5. Computational results & JuMPeR implementation

Based on:

> Berstimas, Dunning. Multistage Robust Mixed Integer Optimization with Adaptive Partitions. Submitted to Operations Research.
> `http://www.optimization-online.org/DB_FILE/2014/11/4658.pdf`

MOTIVATION

> ...the original problem that started my research is still outstanding - namely the problem of planning or scheduling dynamically over time, particularly planning dynamically under uncertainty. If such a problem could be successfully solved it could eventually through better planning contribute to the well-being and stability of the world. - George Dantzig, in “History of Mathematical Programming”, 1991

- Planning decisions across time under uncertainty is at the core of operations research.
- Decisions can be
  - **continuous** - e.g. how much stock to order
  - **discrete** - e.g. whether to operate a coal-fired power plant

The difficulty arises from the **uncertainty** in our problem

- Must make modeling decision about how to represent it:
  - May have good short-term estimates of uncertainty, but long-term?
- Must model adaptability:
  - We need to decide some things **here-and-now**
  - But can **wait-and-see** for later decisions
- Must be **tractable**

Operations management: inventory control, supply chain flexibility, project management...

Industrial: electricity unit commitment, facility location/expansion, air traffic control...

Financial: portfolio construction, financial instruments...

Take a **robust optimization** view of uncertainty

- Assume little about uncertainty
- Good evidence of tractability

Alternative would be a **stochastic optimization** view

- Multistage discrete decisions difficult
- Need distributions

FULLY-ADAPTIVE MULTISTAGE ROBUST OPTIMIZATION PROBLEM

$$z_{\text{full}} = \min_{\mathbf{x}} \max_{\boldsymbol{\xi}\in\Xi} \sum_{t=1}^{T} \mathbf{c}^t(\boldsymbol{\xi}) \cdot \mathbf{x}^t\left(\boldsymbol{\xi}^1, \ldots, \boldsymbol{\xi}^{t-1}\right)$$

$$\text{subject to} \quad \sum_{t=1}^{T} \mathbf{A}^t(\boldsymbol{\xi}) \cdot \mathbf{x}^t\left(\boldsymbol{\xi}^1, \ldots, \boldsymbol{\xi}^{t-1}\right) \leq \mathbf{b}(\boldsymbol{\xi}) \quad \forall \boldsymbol{\xi} = \left(\boldsymbol{\xi}^1, \ldots, \boldsymbol{\xi}^T\right) \in \Xi$$

$$\mathbf{x} \in \mathcal{X}$$

- T time stages, $t = 1$ is here-and-now
- Uncertain parameters $\boldsymbol{\xi}^t$ for each time t
  - Uncertainty set Ξ, captures correlation across time
- Adaptive decisions $\mathbf{x}^t$ for each time t
  - **Fully adaptive** because policy is arbitrary function of complete history
- Deterministic & integrality constraints $\mathcal{X}$

A HIERACHY OF ADAPTABILITY

One extreme: **static policy**

- Future decisisions cannot adapt - all here & now
- Very conservative, but very tractable

Other extreme: **fully adaptive policy**

- Generally intractable
- Some success for unit commitment problem (Berstimas et al 2013)

In-between: assume simpler policy

Linear decision rules, a.k.a. affine adaptability

- Applied to RO in (Ben-tal et. al. 2004)
- Good: problem class OK, simple, sometimes optimal
- Bad: no discrete recourse, changes problem structure, numerical problems
- Extensions: deflected linear decision rules (Chen et. al 2008), polynomial adaptability (Bertsimas et. al. 2010)

Piecewise linear decision rules

- Relatively new, (Bertsimas & Georghiou 2013, 2014)
- Piecewise linear for continuous decisions, piecewise constant for integer
- (2013) uses a cutting-plane method, scaling issues
- (2014) shows good results for multistage.

Finite adaptability

- Partition the uncertainty set, associate decision for each
- Effectively: piecewise constant policy, works well for discrete
- Preserves problem structure better than LDRs
- But can combine with LDRs to get piecewise LDRs!

How to pick the partitions?

- A priori, e.g. (Vayanos et. al. 2011)
- Fix number of partitions and optimize, e.g. (Bertsimas & Caramanis 2010), (Hanasusanto et. al. 2014)
- Optimizing directly results in very difficult MIO

OUR APPROACH

Solve problem with static policy

Indentify good partitions heuristically

Solve partitioned problem

Identify more partitions, or stop

TWO-STAGE PROBLEMS

THE TWO-STAGE PROBLEM

$$
\begin{aligned}
\min_{\mathbf{x},z} \; & z \\
\text{subject to} \quad & \mathbf{c}^1(\boldsymbol{\xi}) \cdot \mathbf{x}^1 + \mathbf{c}^2(\boldsymbol{\xi}) \cdot \mathbf{x}^2(\boldsymbol{\xi}) \leq z \qquad \forall \boldsymbol{\xi} \in \Xi \\
& \mathbf{a}_i^1(\boldsymbol{\xi}) \cdot \mathbf{x}^1 + \mathbf{a}_i^2(\boldsymbol{\xi}) \cdot \mathbf{x}^2(\boldsymbol{\xi}) \leq b_i(\boldsymbol{\xi}) \forall \boldsymbol{\xi} \in \Xi, \; i \in \{1, \dots, m\} \\
& \mathbf{x} \in \mathcal{X},
\end{aligned}
$$

- Consider solving the static policy, continuous two-stage problem
- Using cutting-planes, we will add constraints until solution is feasible
- For every “uncertain constraint” we might add multiple cuts
- Each cut is associated with a value of $\boldsymbol{\xi}$
- If we remove all cuts that have slack $s > 0$, **solution will not change**
- Active constraints = active uncertain parameters, or “samples”

- Let $\hat{\Xi}$ be set of active uncertain parameters, 0 or 1 per uncertain constraint.
- Construct arbitrary partition of Ξ to create Ξ_1 and Ξ_2

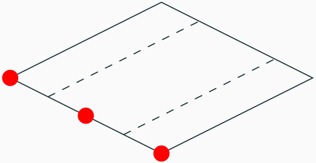
Theorem

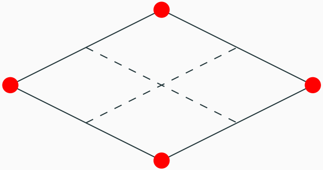
If either $\hat{\Xi} \subseteq \Xi_1$ or $\hat{\Xi} \subseteq \Xi_2$, then there will be no improvement in the objective.

Proof.

If $\hat{\Xi} \subseteq \Xi_1$ then the solution associated with that partition can't be better than for Ξ, so overall solution is the same. $\square$

- We have to split the active uncertain parameters to improve the solution
- Given a set of N points, a **Voronoi diagram** defines a partition for each point such that each point in the partition is closer to that point than any other
- Use active uncertain parameters as the points





Partitions defined by hyperplanes:

$$
\begin{aligned}
\Xi\left(\hat{\boldsymbol{\xi}}_i\right) &= \Xi \cap \left\{ \boldsymbol{\xi} \,\middle|\, \left\| \hat{\boldsymbol{\xi}}_i - \boldsymbol{\xi} \right\|_2 \leq \left\| \hat{\boldsymbol{\xi}}_j - \boldsymbol{\xi} \right\|_2, \forall j \in I,\ i \neq j \right\} \\
&= \Xi \cap \left\{ \boldsymbol{\xi} \,\middle|\, \sum_k \left(\hat{\xi}_{i,k} - \xi_k \right)^2 \leq \sum_k \left(\hat{\xi}_{j,k} - \xi_k \right)^2, \forall j \in I,\ i \neq j \right\} \\
&= \Xi \cap \left\{ \boldsymbol{\xi} \,\middle|\, \sum_k \left(\frac{\hat{\xi}_{i,k} - \hat{\xi}_{j,k}}{2} \right) \xi_k \geq \sum_k \left(\hat{\xi}^2_{i,k} - \hat{\xi}^2_{j,k} \right), \forall j \in I,\ i \neq j \right\},
\end{aligned}
$$

- Each partition defined by Ξ + $N - 1$ linear constraints
- Polyhedral Ξ gives polyhedral partitions
- Computational complexity of Voronoi diagrams bad if enumerating, but we don't need to

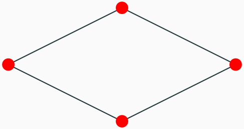
REPEATED PARTITIONING

Solve $\rightarrow$ active parameters $\rightarrow$ partition $\rightarrow$ solve

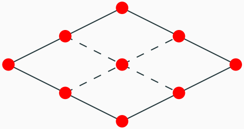
Two options for further partitioning:

- Collect new active parameters, and reconstruct partitions **non-nested**
- Associate active parameters with partitions, create **nested** partitions

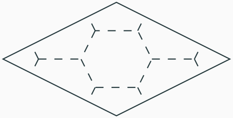
NON-NESTED



$$\mathcal{F}^1 = \{\}$$

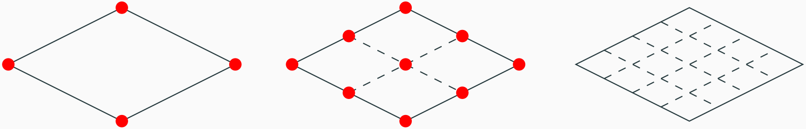


$$\mathcal{F}^2 = \{(10, 10), (20, 5), (30, 10), (20, 15)\}$$



$$\mathcal{F}^3 = \{(10, 10), (15, 12.5), (20, 15), (25, 12.5), (30, 10), (20, 10), (15, 7.5)\,(20, 5)\,(25, 7.5)\}$$

NESTED



$\mathcal{T}^1 = \text{root}$

$\mathcal{T}^2 = \text{root}$
- 10,10
- 20,15
- 30,10
- 20,5

$\mathcal{T}^3 = \text{root}$
- 10,10
  - 10,10
  - 15,12.5
  - 20,10
  - 15,7.5
- 20,15
  - ...
- 30,10
  - ...
- 20,5
  - 20,10
  - 25,12.5
  - 30,10
  - 25,7.5

$$\text{Siblings}\left(\hat{\boldsymbol{\xi}}\right) = \text{Children}\left(\text{Parent}\left(\hat{\boldsymbol{\xi}}\right)\right).$$

$$\begin{aligned}\Xi\left(\hat{\boldsymbol{\xi}}_{\mathrm{i}}\right) = &\left\{\boldsymbol{\xi} \,\middle|\, \left\|\hat{\boldsymbol{\xi}}_{\mathrm{i}} - \boldsymbol{\xi}\right\|_2 \leq \left\|\hat{\boldsymbol{\xi}}_{\mathrm{j}} - \boldsymbol{\xi}\right\|_2 \quad \forall \hat{\boldsymbol{\xi}}_{\mathrm{j}} \in \text{Sibls}\left(\hat{\boldsymbol{\xi}}_{\mathrm{i}}\right)\right\} \\ &\cap \left\{\boldsymbol{\xi} \,\middle|\, \left\|\text{Parent}\left(\hat{\boldsymbol{\xi}}_{\mathrm{i}}\right) - \boldsymbol{\xi}\right\|_2 \leq \left\|\hat{\boldsymbol{\xi}}_{\mathrm{j}} - \boldsymbol{\xi}\right\|_2 \quad \forall \hat{\boldsymbol{\xi}}_{\mathrm{j}} \in \text{Sibl}\left(\text{Parent}\left(\hat{\boldsymbol{\xi}}_{\mathrm{i}}\right)\right)\right\} \\ &\vdots \\ &\cap \Xi,\end{aligned}$$

EXAMPLE: TWO STAGE INVENTORY CONTROL

- Must order stock to meet future unknown demand
- Can order any amount **now**, cost \$50 per unit
- Realize demand, then can order
  - One bulk shipment of 25 units at \$60 per unit
  - Another bulk shipment of 25 units at \$75 per unit
- Holding costs of \$65 per unit

$$
\begin{aligned}
& \min z \\
\text{subject to}\ & 50x^1 + 65I^2(\xi) + 1500y_A^2(\xi) + 1875y_B^2(\xi) \leq z && \forall \xi \in \Xi \\
& I^2(\xi) \geq 0 && \forall \xi \in \Xi \\
& x^1 \geq 0 \\
& y_A^2(\xi),\ y_B^2(\xi) \in \{0, 1\} && \forall \xi \in \Xi
\end{aligned}
$$

$$
I^2(\xi) = x^1 - \xi + 25y_A^2(\xi) + 25y_B^2(\xi), \qquad \Xi = \{\xi \,|\, 5 \leq \xi \leq 95\}
$$

Solve static policy:

- $z = 10600$, $x^1 = 95$, $y_A^2 = 0$, $y_B^2 = 0$
- Worst cases are $\hat{\xi} = 5$ and $\hat{\xi} = 95$
- Create two partitions:
    - $\Xi\left(\hat{\xi} = 5\right) = \{\xi\,|5 \le \xi \le 50\}$
    - $\Xi\left(\hat{\xi} = 95\right) = \{\xi\,|50 \le \xi \le 95\}$

Solve with new partitions:

- $z = 7926$, $x^1 = 70$
    - $y_{A,1}^2 = 0$, $y_{B_1}^2 = 0$
    - $y_{A,2}^2 = 1$, $y_{B_2}^2 = 0$
- Worst cases are $\hat{\xi} = 5, \hat{\xi} = 50$ and $\hat{\xi} = 50, \hat{\xi} = 95$

Non-nested version:

- $\Xi\left(\hat{\xi}=5\right)=\{\xi\,|5\leq\xi\leq35\}$
- $\Xi\left(\hat{\xi}=50\right)=\{\xi\,|35\leq\xi\leq65\}$
- $\Xi\left(\hat{\xi}=95\right)=\{\xi\,|65\leq\xi\leq95\}$
- $z=7575$, $x^1=45$, and

$$y_A^2(\xi)=\begin{cases}0, & 5\leq\xi<35,\\ 1, & 35\leq\xi\leq95,\end{cases}$$

and

$$y_B^2(\xi)=\begin{cases}0, & 5\leq\xi<35,\\ 1, & 65\leq\xi\leq95.\end{cases}$$

Nested partitions:

- Parent $\hat{\xi} = 5$:
  - $\Xi\left(\hat{\xi} = 5\right) = \{\xi \,|5 \leq \xi \leq 27.5\}$
  - $\Xi\left(\hat{\xi} = 50\right) = \{\xi \,|27.5 \leq \xi \leq 50\}$
- Parent $\hat{\xi} = 95$:
  - $\Xi\left(\hat{\xi} = 50\right) = \{\xi \,|50 \leq \xi \leq 72.5\}$
  - $\Xi\left(\hat{\xi} = 95\right) = \{\xi \,|72.5 \leq \xi \leq 95\}$
- $z = 7375$, $x^1 = 47.5$, and

$$y_A^2(\xi) = \begin{cases} 0, & 5 \leq \xi < 27.5, \\ 1, & 27.5 \leq \xi \leq 95, \end{cases}$$

and

$$y_B^2(\xi) = \begin{cases} 0, & 5 \leq \xi < 72.5, \\ 1, & 72.5 \leq \xi \leq 95. \end{cases}$$

Summary of results:

- Static: $z = 10600$
- First iteration: $z = 7926$
- Second iteration, non-nested: $z = 7575$
- Second iteration, nested: $z = 7375$
- Fully adaptive: $z = 7250$

Can easily incorporate linear decision rules

Make substitution

$$\mathsf{x}^2(\boldsymbol{\xi}) = \mathsf{F}\boldsymbol{\xi} + \mathsf{g},$$

for continuous decisions

Equivalent to piecewise affine once we partition, but with breaks heuristically determined, i.e.

$$\mathsf{x}^2(\boldsymbol{\xi}) = \begin{cases} \mathsf{F}_1\boldsymbol{\xi} + \mathsf{g}_2, & \boldsymbol{\xi} \in \Xi\left(\hat{\boldsymbol{\xi}}_1\right), \\ \mathsf{F}_2\boldsymbol{\xi} + \mathsf{g}_2, & \boldsymbol{\xi} \in \Xi\left(\hat{\boldsymbol{\xi}}_2\right), \\ \vdots & \vdots \end{cases}$$

There is an objective value associated with each partition

- Define active partition as partition that is binding overall objective
- Only look at uncertain parameters for active partitions
- e.g. in example, partition objectives are 5137.5, 6800, 5337.5, and 7375

We have been using uncertain parameters with minimum slack

- Could use only $s = 0$ uncertain parameters
- Situation dependent: may be hard to get 0 slack in many problems

MULTISTAGE PROBLEMS

One thing: must satisfy **non-anticipativity**

With affine, automatically satisfied

With finite adaptability, **easy to violate!**

Consider the same inventory problem as before, but $T = 3$

Uncertainty set

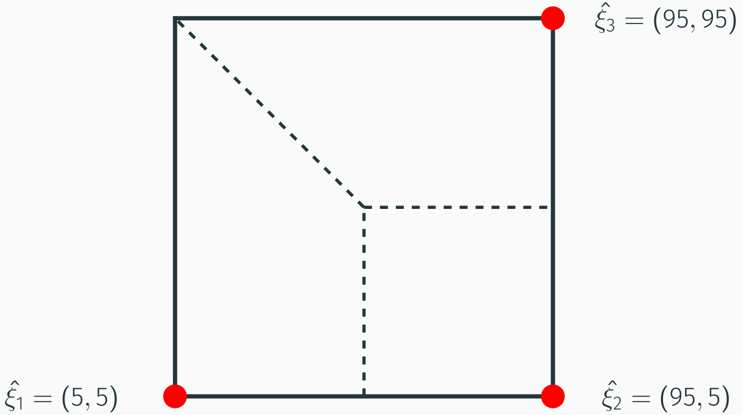
$$\Xi = \left\{5 \leq \xi^1, \xi^2 \leq 95\right\}$$

Active samples for static policy are

- $\hat{\boldsymbol{\xi}}_1 = (5, 5)$ (objective)
- $\hat{\boldsymbol{\xi}}_2 = (95, 5)$ (demand met, $t = 2$)
- $\hat{\boldsymbol{\xi}}_3 = (95, 95)$ (demand met, $t = 3$)

Construct partitions as before...

EXAMPLE OF MULTISTAGE DIFFICULTY



Had to add $x_1^2 = x_3^2$ and $x_2^2 = x_3^2$, no adaptability left at $t = 2$

Solution: modify partitioning scheme to be aware of time

Goal: balance having minimal set of anticipativity constraints while getting most useful partitions.

For each pair $\hat{\boldsymbol{\xi}}_i$ and $\hat{\boldsymbol{\xi}}_j$ as before

1. Determine which components $\hat{\boldsymbol{\xi}}^t$ shared
2. Construct hyperplane using only components up to the first time stage they differ, i.e.

$$\Xi\left(\hat{\boldsymbol{\xi}}_i\right) = \Xi \cap \left\{ \boldsymbol{\xi} \,\middle|\, \left\| \hat{\boldsymbol{\xi}}_i^{t_{i,j}} - \boldsymbol{\xi}^{t_{i,j}} \right\|_2 \leq \left\| \hat{\boldsymbol{\xi}}_j^{t_{i,j}} - \boldsymbol{\xi}^{t_{i,j}} \right\|_2 \quad \forall \hat{\boldsymbol{\xi}}_j \in \mathcal{F}^k \right\}$$

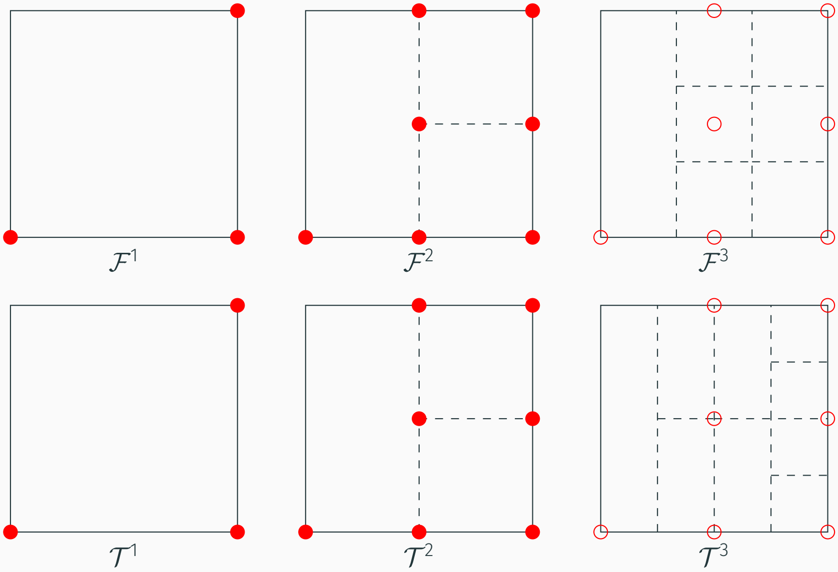
   where $t_{i,j}$ is the min t s.t. $\hat{\boldsymbol{\xi}}_i^t \neq \hat{\boldsymbol{\xi}}_j^t$
3. Proposition: sufficient to then enforce $x_i^t = x_j^t$ iff $\hat{\boldsymbol{\xi}}_i^{1,\ldots,t-1} = \hat{\boldsymbol{\xi}}_j^{1,\ldots,t-1}$ to ensure nonanticipativity

Apply same time-dependent partitioning rules, but nested

Can use similar rule to add non-anticipativity, but it is **overconservative**

Relatively cheap to check intersection of partitions, much less conservative

MULTISTAGE NESTED PARTITIONING



BOUNDS

Provide three types of bounds:

1. Lower bound on fully adaptive solution
2. Upper bound on subsequent iterations (monotonicity)
3. Lower bound on subsequent iterations

MIO branch-and-bound: have best integer UB, continuous relaxation LB

- Termination criteria e.g. $\frac{(\mathrm{UB}-\mathrm{LB})}{\mathrm{LB}}$

AMIO: UB is best approximation to fully adaptive, LB $=$?

- Termination criteria could be same e.g. $\frac{(\mathrm{UB}-\mathrm{LB})}{\mathrm{LB}}$

Proposition: the solution to

$$
\begin{aligned}
z_{\text{lower}}(\mathcal{A}) = \min_{\mathbf{x},z} \quad & z \\
\text{subject to} \quad & \sum_{t=1}^{T} \mathbf{c}^t\left(\hat{\boldsymbol{\xi}}_i\right) \cdot \mathbf{x}_i^t \leq z && \forall \hat{\boldsymbol{\xi}}_i \in \mathcal{A} \\
& \sum_{t=1}^{T} \mathbf{A}^t\left(\hat{\boldsymbol{\xi}}_i\right) \cdot \mathbf{x}_i^t \leq \mathbf{b}\left(\hat{\boldsymbol{\xi}}_i\right) && \forall \hat{\boldsymbol{\xi}}_i \in \mathcal{A} \\
& \mathbf{x}_i^t = \mathbf{x}_j^t && \forall \hat{\boldsymbol{\xi}}_i, \hat{\boldsymbol{\xi}}_j \in \mathcal{A} \text{ s.t. } \hat{\boldsymbol{\xi}}_i^{1,\ldots,t-1} = \hat{\boldsymbol{\xi}}_j^{1,\ldots,t-1} \\
& \mathbf{x} \in \mathcal{X},
\end{aligned}
$$

is a lower bound to the fully adaptive optimization problem.

Proof follows from the fact that $\mathcal{A}$ is a subset of Ξ, and we respect nonanticipativity. This is similar to the two-stage ”scenario based bound” in Hadjiyiannis et. al. 2011.

LOWER BOUND ON FULLY ADAPTIVE SOLUTION

Open question: can we do better?

This bound is very practical as we have these samples already "free"

Can improve bound by sampling more from uncertainty set, but to what end?

Getting a better bound with what we already know is key to progress

UPPER BOUND ON SUBSEQUENT ITERATIONS

The non-nested variant **does not** decrease monotonically

The nested variant **does** decreases monotonically

Utility: can use solution from one iteration to warm start next iteration

LOWER BOUND ON SUBSEQUENT ITERATIONS

Use duality to estimate improvement that could be obtained by partitioning

Not directly applicable to AMIO, but can check relaxation

Estimate usefulness of partitioning further

COMPUTATIONAL EXPERIMENTS

Observation: spend more time, get better solutions

What is better? Gap? Improvement?

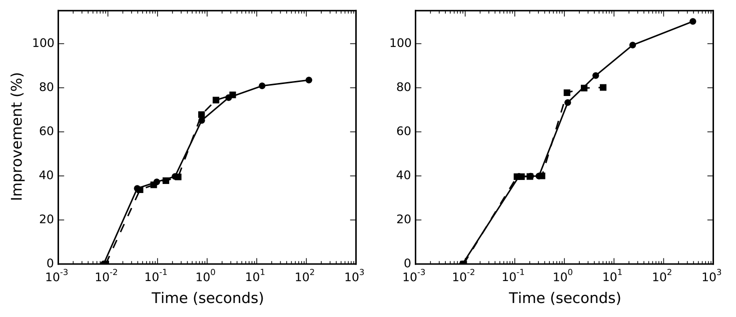
Upper and lower bounds shrink simultaneously

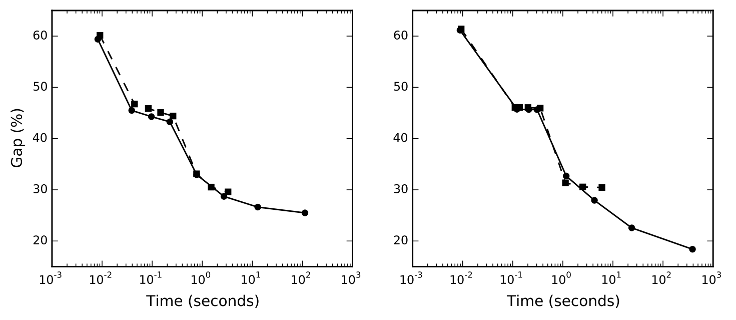
Can build projects now or later, but more expensive later.

$$
\begin{aligned}
\max_{z,\mathbf{x}} \quad & z \\
\text{subject to} \quad & \mathbf{r}(\boldsymbol{\xi}) \cdot \left(\mathbf{x}^1 + \theta \mathbf{x}^2(\boldsymbol{\xi})\right) \geq z && \forall \boldsymbol{\xi} \in \Xi \\
& \mathbf{c}(\boldsymbol{\xi}) \cdot \left(\mathbf{x}^1 + \mathbf{x}^2(\boldsymbol{\xi})\right) \leq B && \forall \boldsymbol{\xi} \in \Xi \\
& \mathbf{x}^1 \in \{0,1\}^N \\
& \mathbf{x}^2(\boldsymbol{\xi}) \in \{0,1\}^N && \forall \boldsymbol{\xi} \in \Xi,
\end{aligned}
$$

Shared uncertain factors induce structure across project revenues and costs.

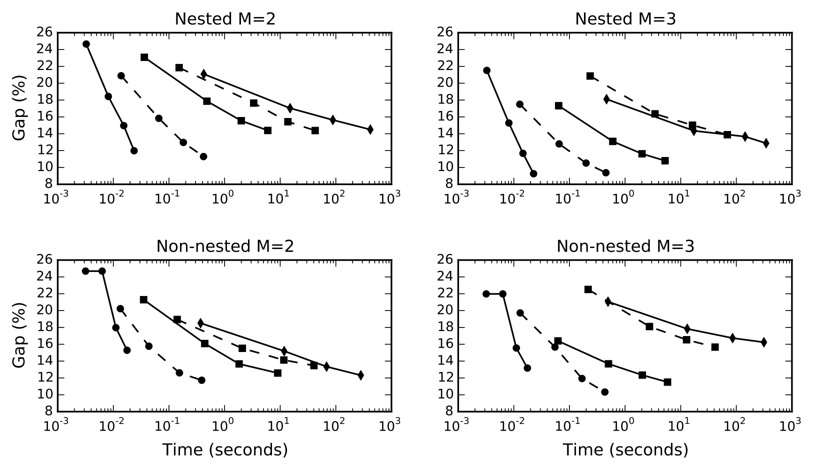
N = 10	Variant	Iteration 1	2	3	4	5	6	7	8
Total Time (s)	Non-nested	0.0	0.0	0.1	0.2	0.8	2.7	12.8	112.4
	Nested	0.0	0.0	0.1	0.1	0.2	0.7	1.5	3.2
Improvement (%)	Non-nested	0	34	37	40	65	76	81	83
	Nested	0	34	36	38	40	68	74	77
Gap (%)	Non-nested	62	48	46	45	34	29	27	34
	Nested	62	48	47	46	45	34	31	30

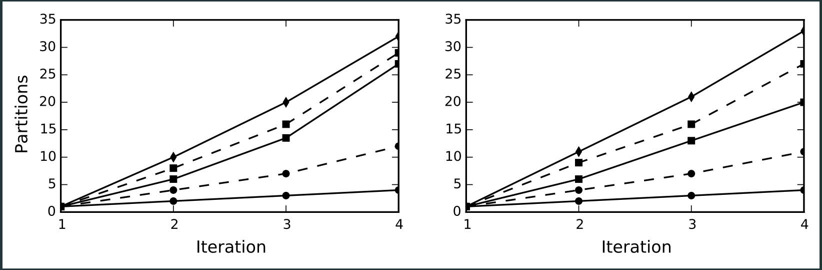

Improvement (%)
Time (seconds)
Time (seconds)


Gap (%)
Time (seconds)
Time (seconds)

Similar to working example before, but generalized to T stages:

$$
\begin{aligned}
\min_{\mathbf{x},\mathbf{y},\mathbf{I}} \quad & \sum_{t=2}^{T} \left(c_x x^{t-1} + c_h I^t + \sum_{m=1}^{M} c_m q_m y_m^t \right) \\
\text{s.t} \quad & I^{t-1} + x^{t-1} + \sum_{,m=1}^{M} q_m y_m^t - \xi^t = I^t && \forall t \in \{2, \dots, T\} \\
& \sum_{s=1}^{t-1} x^s \leq \bar{x}_{tot,t} && \forall t \in \{2, \dots, T\} \\
& I^t \geq 0 && \forall t \in \{2, \dots, T\} \\
& x^{t-1} \geq 0 && \forall t \in \{2, \dots, T\} \\
& \mathbf{y}^t \in \{0,1\}^M, && \forall t \in \{2, \dots, T\}
\end{aligned}
$$


Nested M=2
Nested M=3
Non-nested M=2
Non-nested M=3
Gap (%)
Time (seconds)
26
24
22
20
18
16
14
12
10
8
10^{-3}
10^{-2}
10^{-1}
10^{0}
10^{1}
10^{2}
10^{3}


Partitions
Iteration
Iteration
35
30
25
20
15
10
5
0
1
2
3
4

CONCLUSION

CONCLUSION

Finite adaptability with heuristically chosen partitions performs well

Have a lower bound, can trade off time for quality

Somewhat like branch-and-bound in spirit

FUTURE WORK

Smarter partitions

- Guess and improve?
- Use all active samples

Tighter integration into branch & bound

Better lower bounds

Class projects? Let's collaborate!

JuMPeR doesn't yet have this implemented "for free"

But can easily implement in JuMPeR!

Demo if time, post notebook otherwise