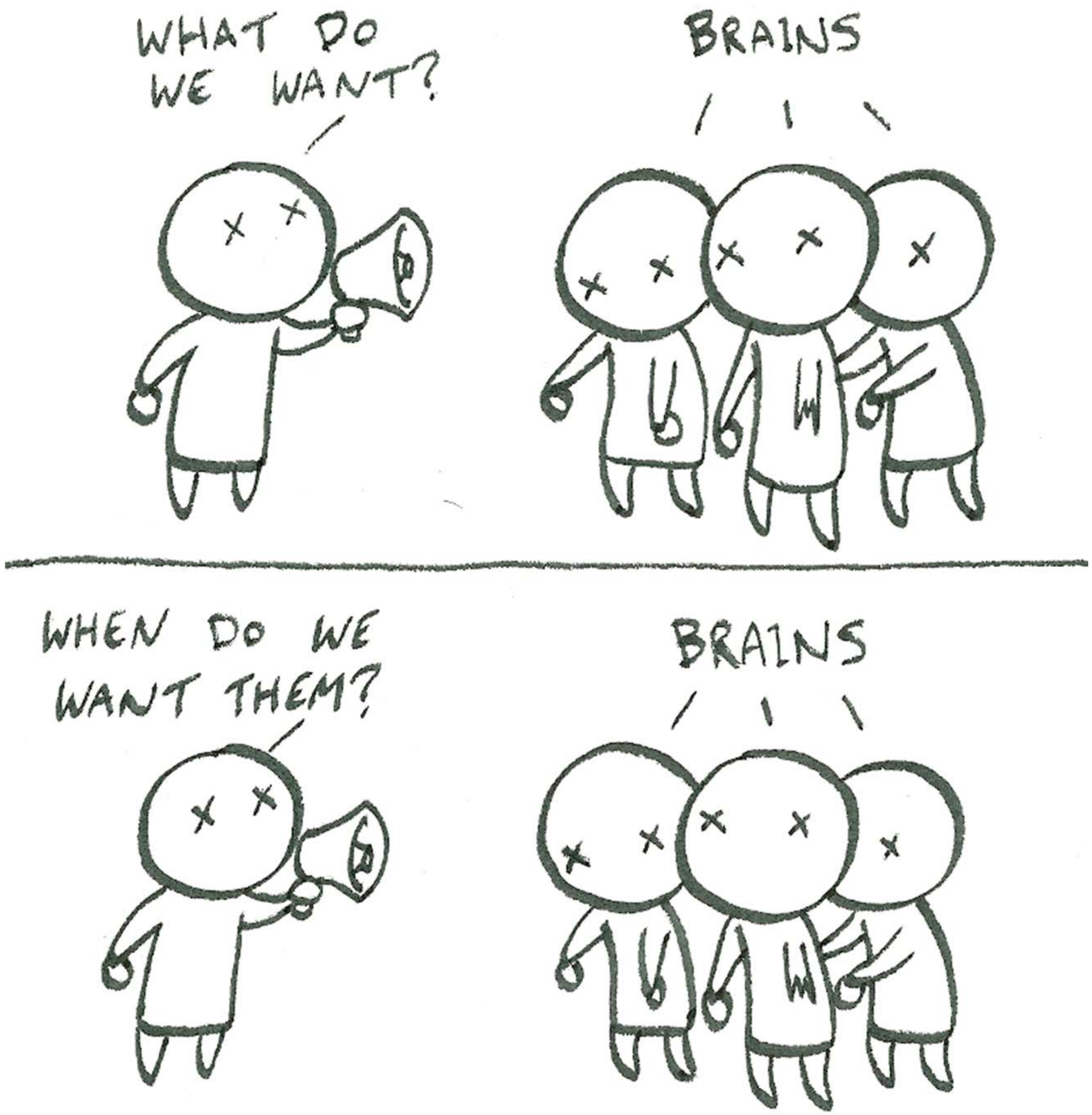
Zombie Optimization
or
How I Learned to Love Decomposition

J. Christopher Beck
Department of Mechanical & Industrial Engineering
University of Toronto
Canada
jcb@mie.utoronto.ca

in collaboration with Mohammad Fazel-Zarandi, Stefan Heinz, Wen-Yang Ku, Daria Terekhov, Jens Schulz, Tony Tran, Peter Zhang

CPAIOR 2013 Master Class
May 18, 2013

University of Toronto
Mechanical & Industrial Engineering



WHAT DO WE WANT?
BRAINS
WHEN DO WE WANT THEM?
BRAINS

Disclaimer #1

- There is really nothing (more) about zombies in this talk
  - that was just to get you in the room today

Disclaimer #1

- There is really nothing (more) about zombies in this talk
  - that was just to get you in the room today

Disclaimer #2

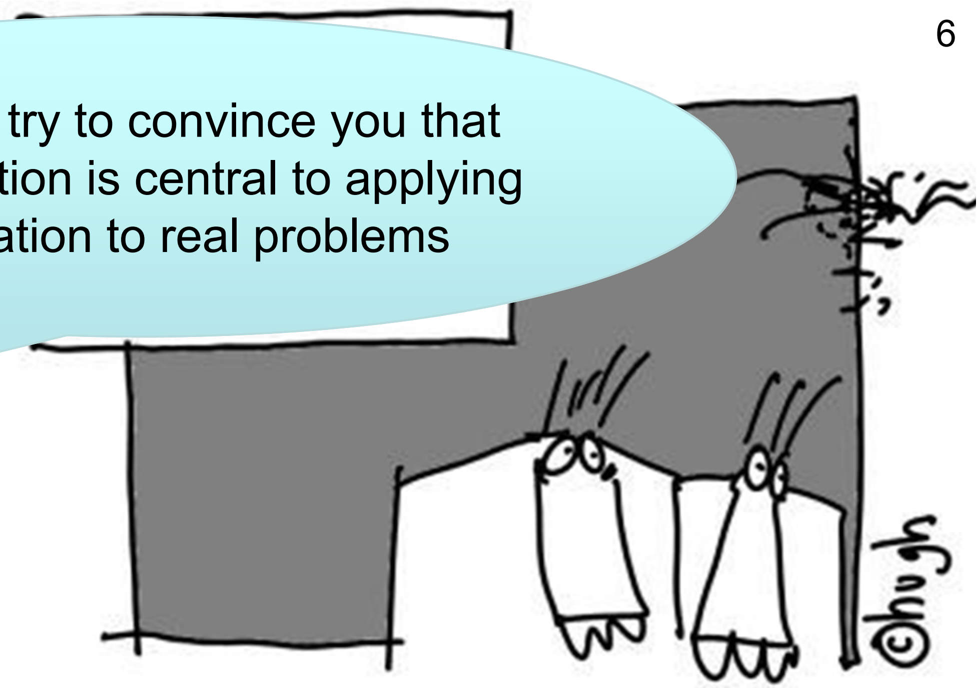
- There is not really much about computational sustainability, either
  - well, there is some
  - I will try, with varying degrees of success, to provide examples of decomposition in problems related to computational sustainability

The Plan

- Decomposition & Modeling
- Logic-based Benders Decomposition (LBBD)
- Applying LBBD to Problems Somewhat Related to Computational Sustainability
- Beyond Decomposition

The Pla

wherein, I try to convince you that
decomposition is central to applying
optimization to real problems



- Decomposition & Modeling
- Logic-based Benders Decomposition (LBBD)
- Applying LBBD to Problems Somewhat Related to Computational Sustainability
- Beyond Decomposition

Wind Farm Design

Wind Farm Design

- You want to build a commercial wind farm
  - what turbines do you buy? how many?
  - where do you build it? what do you build (e.g., turbine foundations, turbine layout, roads, electrical connections, energy storage)?
  - how do you build it (construction planning)?
  - how do you operate it?

[Zhang 2013] *MASc Thesis, University of Toronto*.

Wind Farm Design

- You want to build a commercial wind farm

Somehow you need to decide how to solve all these inter-related problems.

  - where do you build it? what do you build (e.g., turbine foundations, turbine layout, roads, electrical connections, energy storage)?
  - how do you build it (construction planning)?
  - how do you operate it?

[Zhang 2013] *MASc Thesis, University of Toronto*.

Wind Farm Design

- You want to build a commercial wind farm

Somehow you need to decide how to solve all these inter-related problems.

  – where do you build it? what do you build (e.g.,

The only reasonable way forward (as our scientific/engineering methodology has it) is to identify sub-problems we can solve (more or less) independently.

  – how do you operate it?

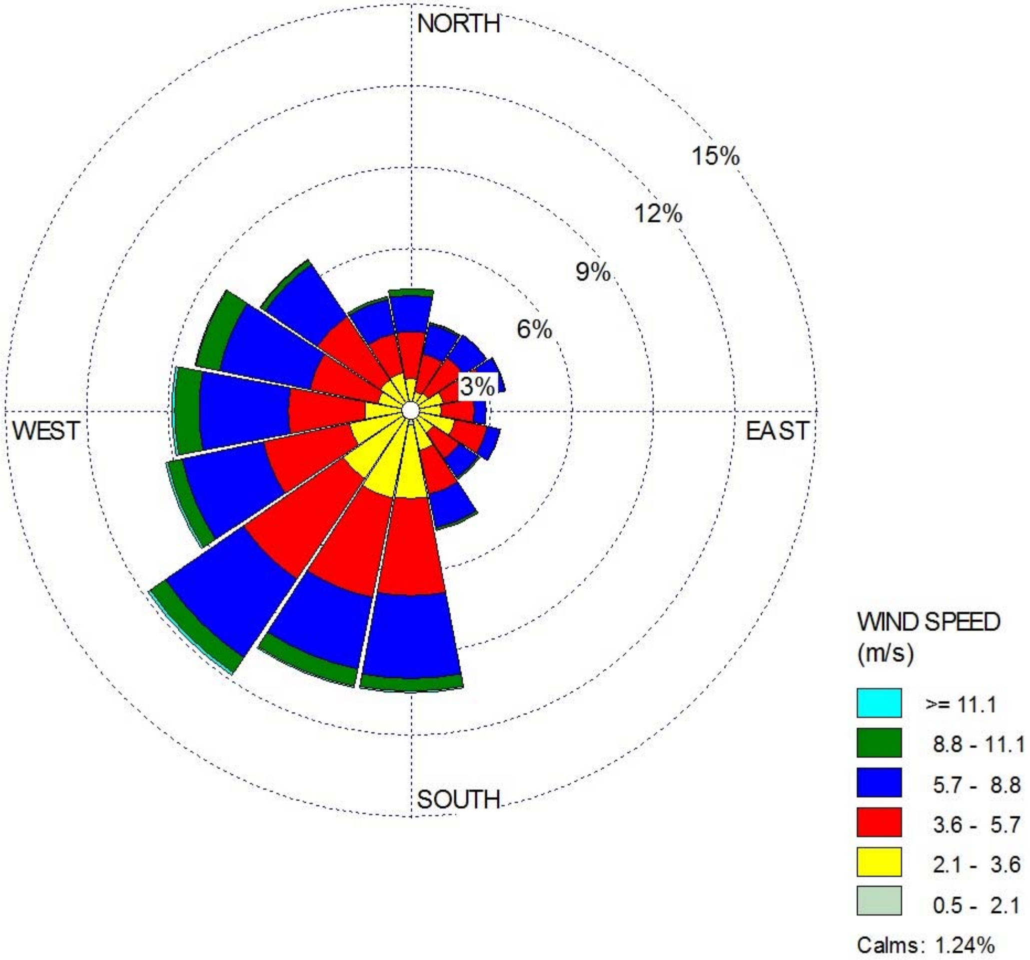
[Zhang 2013] *MASc Thesis, University of Toronto*.

Problem 1: Turbine Placement

- Objective: maximize energy production or profit
- Constraints:
  - location: min. separation, land topology, existing infrastructure
  - limit of input power to grid
  - turbine specifications
- Decisions:
  - turbine types, number, placement

Thanks to Peter Zhang.

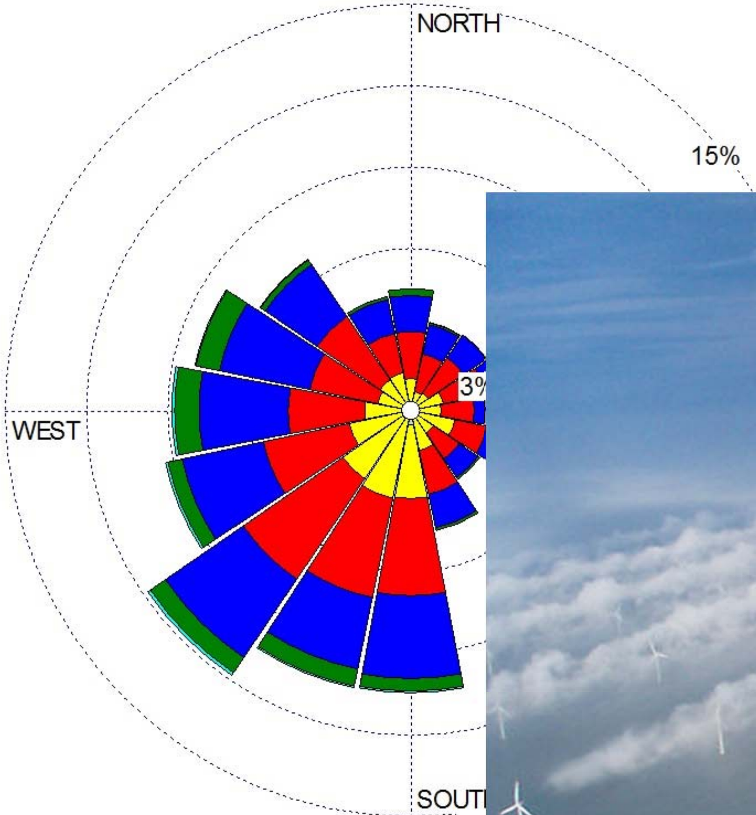
Turbine Placement Challenges



University of Toronto
Mechanical & Industrial Engineering

Thanks to Peter Zhang.

Turbine Placement Challenges



Thanks to Peter Zhang.

Problem 2: Infrastructure Layout

- Design the supporting structure
  - turbine foundations, electrical network, road network, control, monitoring and data gathering
  - reliability, maintenance, life time (stochastic!)
- Power loss via transmission scales with length
  - the turbine placement and electrical network are interdependent

Thanks to Peter Zhang.

Problem 3: Wind Energy Storage

- Smooth supply variations by storing energy (e.g., battery)
  - how big should the battery be?
- Depends on how it is used
  - connection with unit commitment problem
  - economic connection with turbine placement

Thanks to Peter Zhang.

[redacted]

- Stand[redacted]roach: decom[redacted]
  – focus on something [redacted] can solve
- Maybe particula[redacted]erous in computational s[redacted]lity
  – law of unintended [redacted]equences

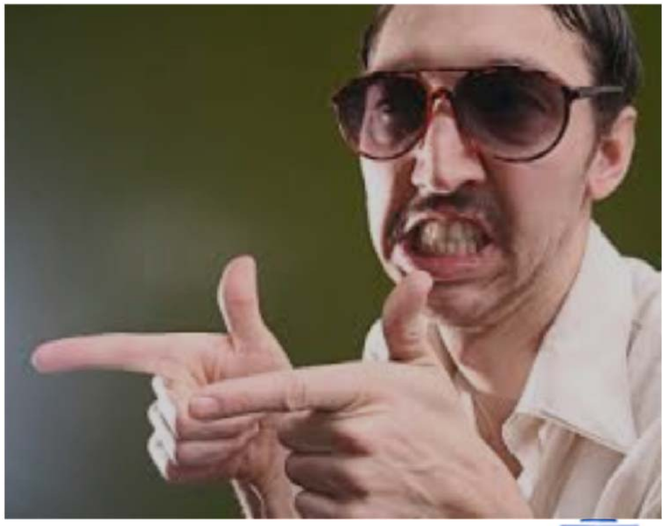
- But the [redacted] problem is ju[redacted]big!

Decomposition

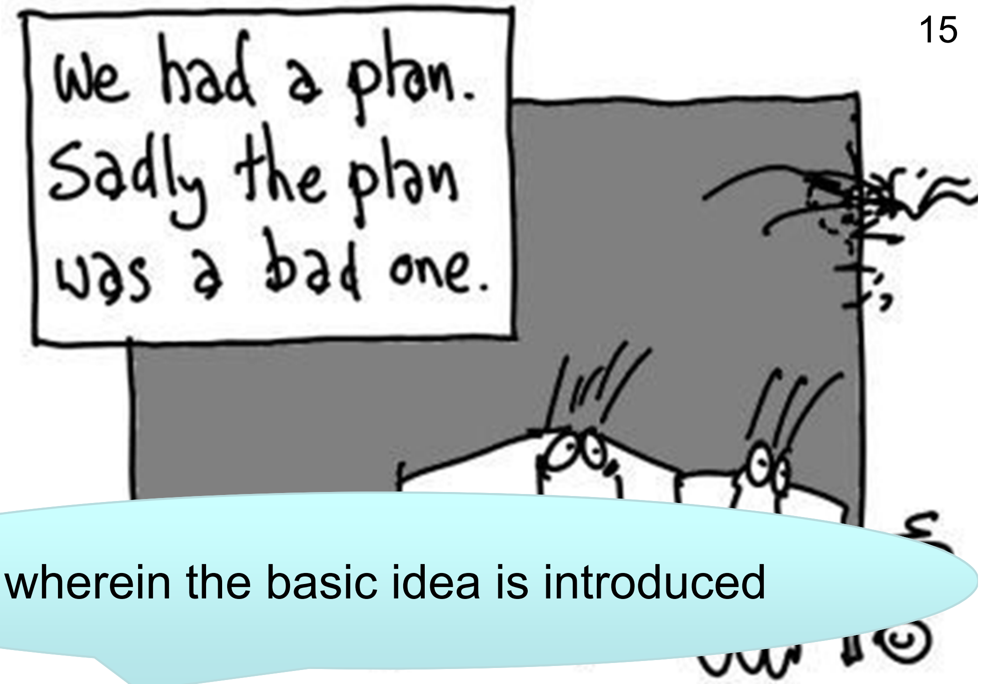
- Hierarchical (the standard way)
  - overall problem is split into sub-problems solved one at at time or independently
    - e.g., infrastructure layout after turbine placement
  - no feedback
- Integrated
  - decisions really depend on each other but problem too big to solve in one model
  - decomposition with feedback

Decomposition

- Hierarchical (the standard way)
  - overall problem is split into sub-problems solved one at at time or independently
    - e.g., infrastructure layout after turbine placement
  - no feedback
- Integrated
  - decisions really depend on each other but problem too big to solve in one model
  - decomposition with feedback



The Plan

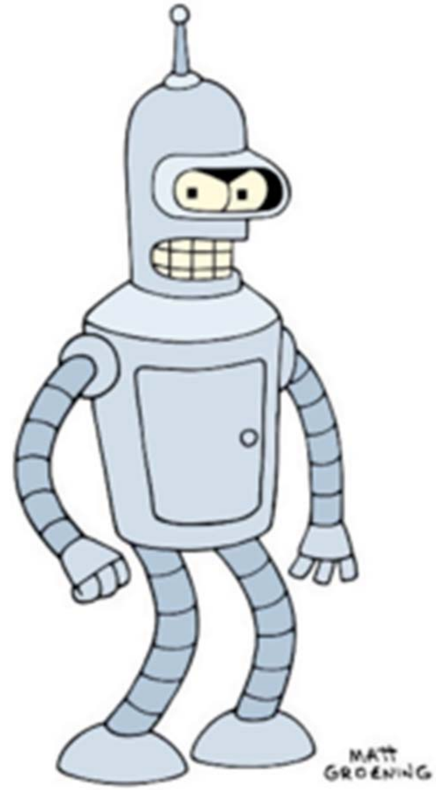


- Decomposition & Modeling
- Logic-based Benders Decomposition (LBBD)

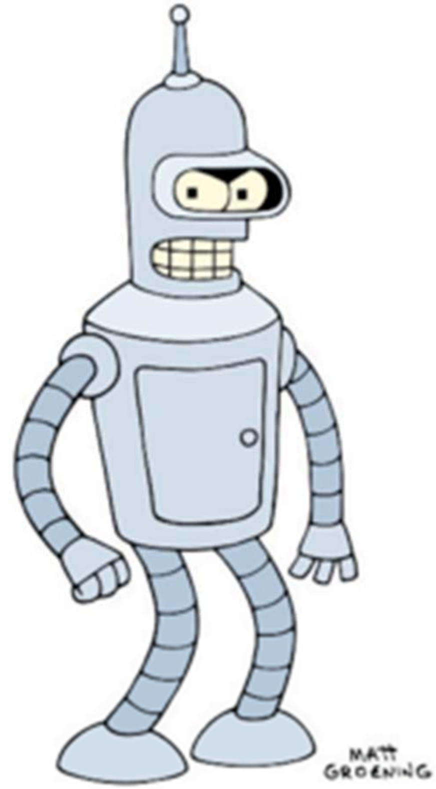
wherein the basic idea is introduced

- Applying LBBD to Problems Somewhat Related to Computational Sustainability
- Beyond Decomposition

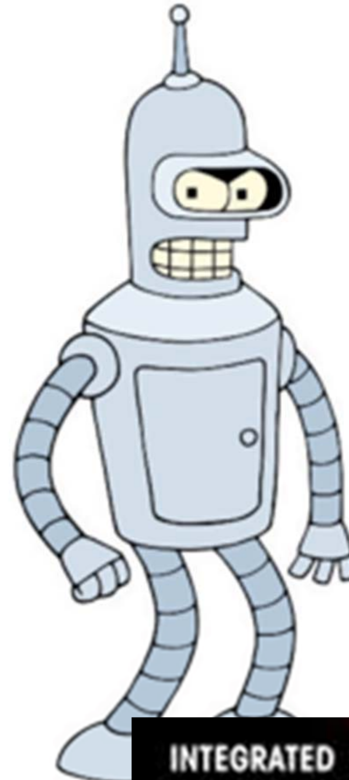
Logic-Based Benders Decomposition

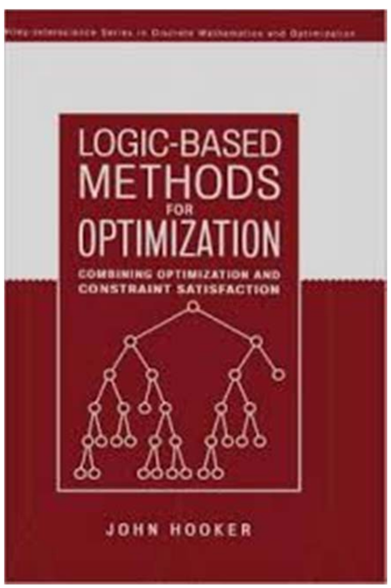


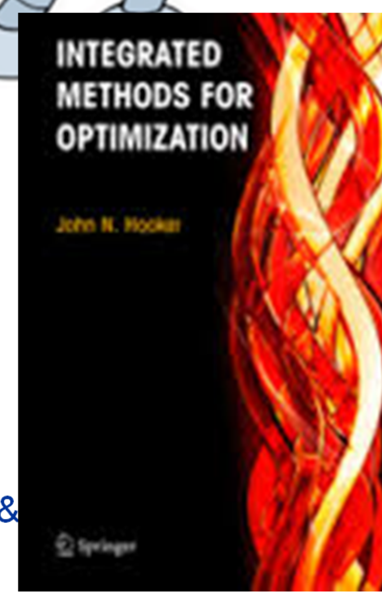
Logic-Based Benders Decomposition



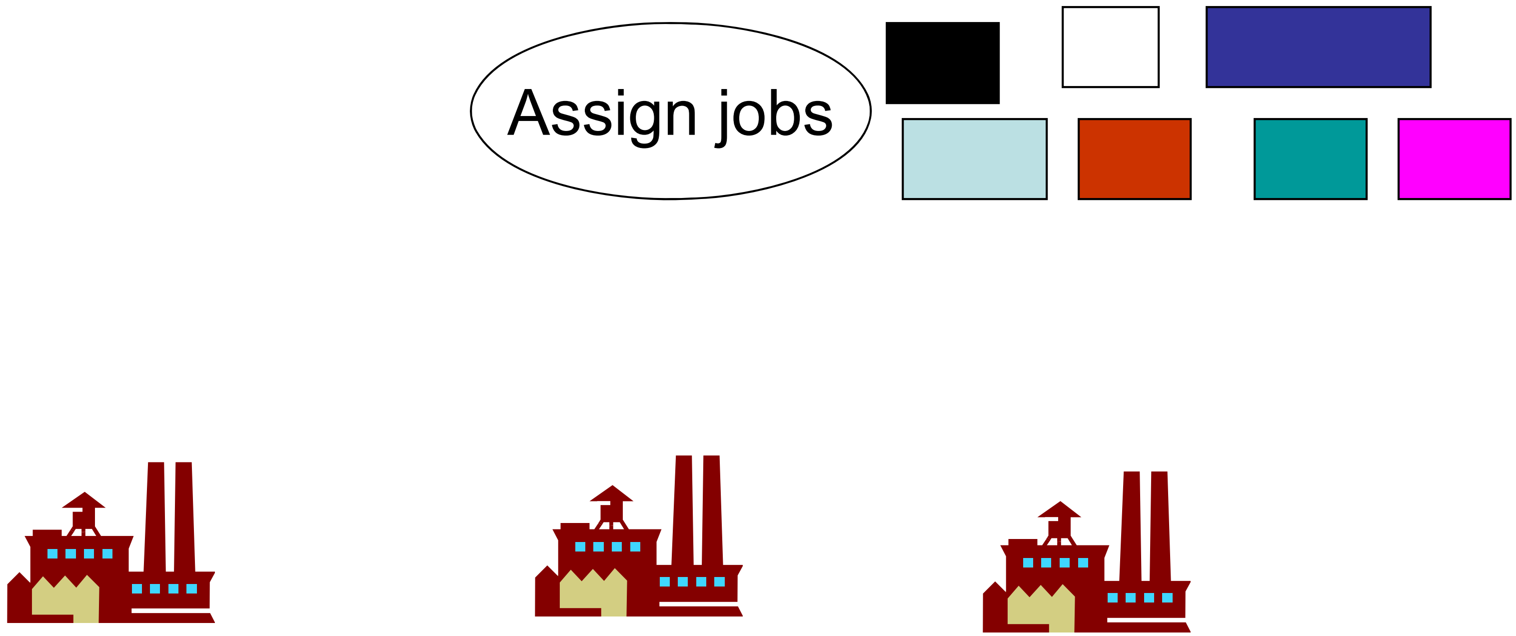
Logic-Based Benders Decomposition





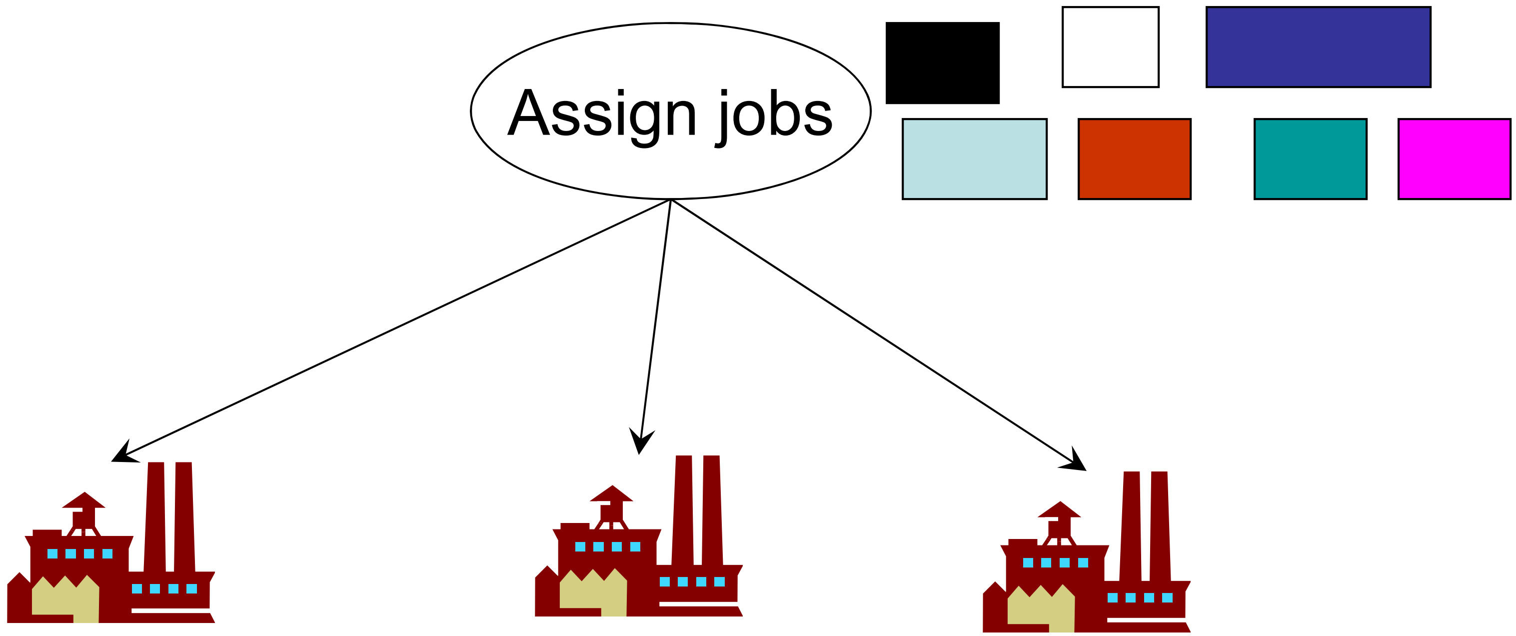


Resource Allocation & Scheduling



[Hooker 2005] *Constraints*, 10, 385-401, 2005.

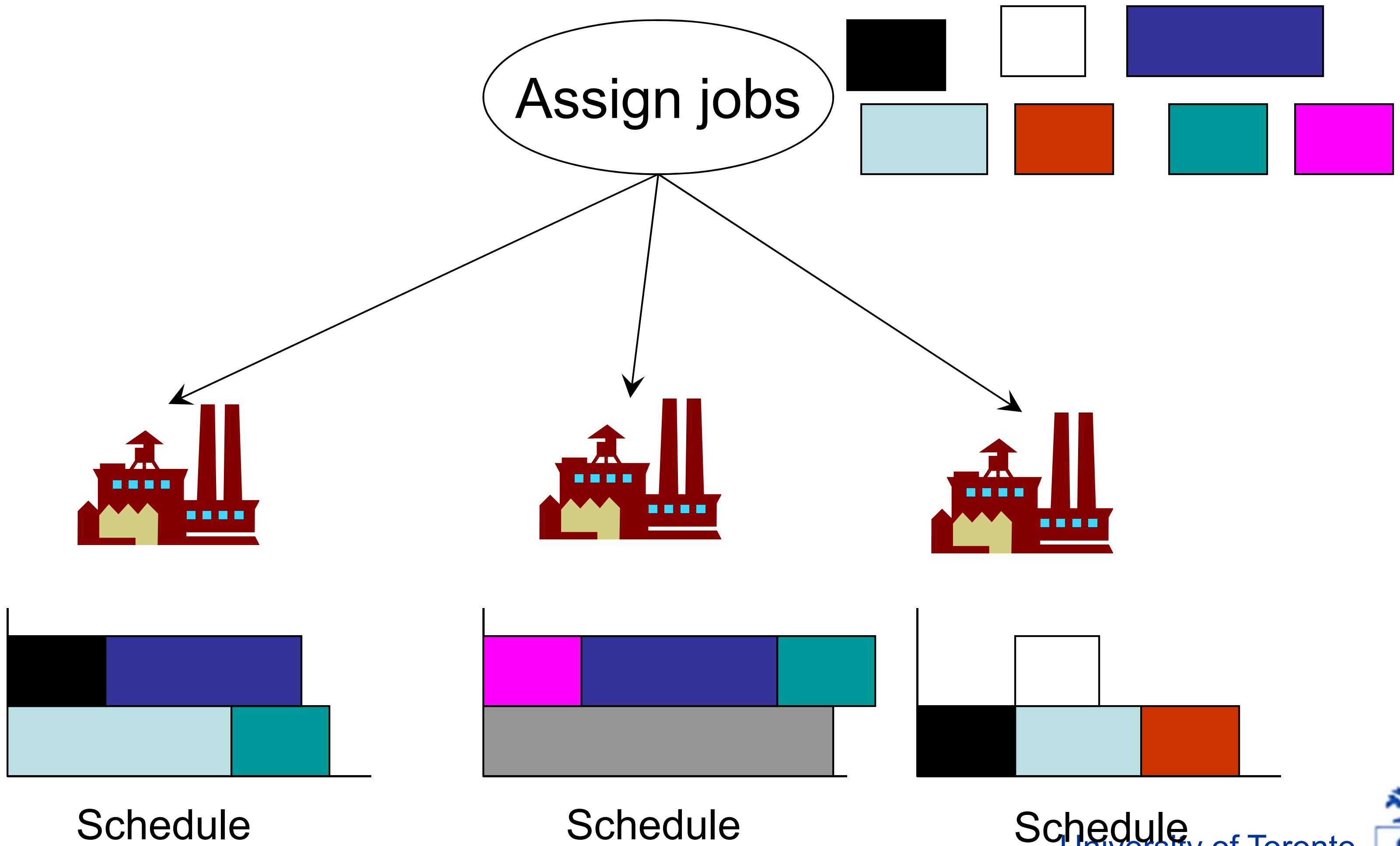
Resource Allocation & Scheduling



University of Toronto
Mechanical & Industrial Engineering

[Hooker 2005] *Constraints*, 10, 385-401, 2005.

Resource Allocation & Scheduling



[Hooker 2005] *Constraints*, 10, 385-401, 2005.

Problem Details

- Each job, *j*, has:
  - release date, R_j (earliest start time)
  - deadline, D_j (latest end time)
  - processing time, p_{jk}, on resource *k*
  - resource requirement, r_{jk}, for resource *k*
  - cost, c_{jk}, to use resource *k*
- Goal: assign and schedule jobs to minimize total assignment cost while satisfying time windows and resource capacity

CP Model

$$
\begin{aligned}
\min \quad & \sum_{k \in \mathcal{K}} \sum_{j \in \mathcal{J}} c_{jk}\, x_{jk} \\
\text{s.t.} \quad & \sum_{k \in \mathcal{K}} x_{jk} = 1 & \forall j \in \mathcal{J} \\
& \texttt{optcumulative}(\boldsymbol{S}, \boldsymbol{x}_{\cdot k}, \boldsymbol{p}_{\cdot k}, \boldsymbol{r}_{\cdot k}, C_k) & \forall k \in \mathcal{K} \\
& 0 \le \mathcal{R}_j \le S_j \le \max_{k \in \mathcal{K}} \{(\mathcal{D}_j - p_{jk})\, x_{jk}\} & \forall j \in \mathcal{J} \\
& x_{jk} \in \{0, 1\} & \forall j \in \mathcal{J},\ \forall k \in \mathcal{K} \\
& S_j \in \mathbb{Z} & \forall j \in \mathcal{J}
\end{aligned}
$$

[Heinz & B. 2012] *CPAIOR*, 211-227, 2012.

CP Model

x_{ij} = 1 if job j is assigned to resource i

$$\begin{aligned}
\min \quad & \sum_{k \in \mathcal{K}} \sum_{j \in \mathcal{J}} c_{jk}\, x_{jk} & \\
\text{s.t.} \quad & \sum_{k \in \mathcal{K}} x_{jk} = 1 & \forall j \in \mathcal{J} \\
& \texttt{optcumulative}(\boldsymbol{S}, \boldsymbol{x}_{\cdot k}, \boldsymbol{p}_{\cdot k}, \boldsymbol{r}_{\cdot k}, C_k) & \forall k \in \mathcal{K} \\
& 0 \le \mathcal{R}_j \le S_j \le \max_{k \in \mathcal{K}} \{(\mathcal{D}_j - p_{jk})\, x_{jk}\} & \forall j \in \mathcal{J} \\
& x_{jk} \in \{0, 1\} & \forall j \in \mathcal{J},\ \forall k \in \mathcal{K} \\
& S_j \in \mathbb{Z} & \forall j \in \mathcal{J}
\end{aligned}$$

[Heinz & B. 2012] *CPAIOR*, 211-227, 2012.

CP Model

$$
\begin{aligned}
\min \quad & \sum_{k \in \mathcal{K}} \sum_{j \in \mathcal{J}} c_{jk}\, x_{jk} \\
\text{s.t.} \quad & \sum_{k \in \mathcal{K}} x_{jk} = 1 \\
& \texttt{optcumulative}(\boldsymbol{S}, \boldsymbol{x}_{\cdot k}, \boldsymbol{p}_{\cdot k}, \boldsymbol{r}_{\cdot k}, C_k) && \forall k \in \mathcal{K} \\
& 0 \le \mathcal{R}_j \le S_j \le \max_{k \in \mathcal{K}} \{ (\mathcal{D}_j - p_{jk})\, x_{jk} \} && \forall j \in \mathcal{J} \\
& x_{jk} \in \{0, 1\} && \forall j \in \mathcal{J},\ \forall k \in \mathcal{K} \\
& S_j \in \mathbb{Z} && \forall j \in \mathcal{J}
\end{aligned}
$$

x_{ij} = 1 if job j is assigned to resource i

all jobs assigned to one resource

[Heinz & B. 2012] *CPAIOR*, 211-227, 2012.

CP Model

$$\min \sum_{k\in\mathcal{K}} \sum_{j\in\mathcal{J}} c_{jk}\, x_{jk}$$

$$\text{s.t.} \quad \sum_{k\in\mathcal{K}} x_{jk} = 1$$

$$\texttt{optcumulative}(\boldsymbol{S}, \boldsymbol{x}_{\cdot\boldsymbol{k}}, \boldsymbol{p}_{\cdot\boldsymbol{k}}, \boldsymbol{r}_{\cdot\boldsymbol{k}}, C_k)$$

$$0 \le \mathcal{R}_j \le S_j \le \max_{k\in\mathcal{K}}\{(\mathcal{D}_j - p_{jk})\, x_{jk}\} \qquad \forall j \in \mathcal{J}$$

$$x_{jk} \in \{0,1\} \qquad \forall j \in \mathcal{J},\ \forall k \in \mathcal{K}$$

$$S_j \in \mathbb{Z} \qquad \forall j \in \mathcal{J}$$

x_{ij} = 1 if job j is assigned to resource i

all jobs assigned to one resource

resource capacity

[Heinz & B. 2012] *CPAIOR*, 211-227, 2012.

CP Model

$$\min \quad \sum_{k \in \mathcal{K}} \sum_{j \in \mathcal{J}} c_{jk}\, x_{jk}$$

$$\text{s.t.} \quad \sum_{k \in \mathcal{K}} x_{jk} = 1$$

$$\texttt{optcumulative}(\boldsymbol{S}, \boldsymbol{x_{\cdot k}}, \boldsymbol{p_{\cdot k}}, \boldsymbol{r_{\cdot k}}, C_k)$$

$$0 \le \mathcal{R}_j \le S_j \le \max_{k \in \mathcal{K}}\{(\mathcal{D}_j - p_{jk})\, x_{jk}\}$$

$$x_{jk} \in \{0, 1\} \qquad \forall j \in \mathcal{J}, \forall k \in \mathcal{K}$$

$$S_j \in \mathbb{Z} \qquad \forall j \in \mathcal{J}$$

x_{ij} = 1 if job j is assigned to resource i

all jobs assigned to one resource

resource capacity

time windows

[Heinz & B. 2012] *CPAIOR*, 211-227, 2012.

University of Toronto
Mechanical & Industrial Engineering

CP Model

$$\min \quad \sum_{k \in \mathcal{K}} \sum_{j \in \mathcal{J}} c_{jk}\, x_{jk}$$

$$\text{s.t.} \quad \sum_{k \in \mathcal{K}} x_{jk} = 1$$

$$\texttt{optcumulative}(\boldsymbol{S}, \boldsymbol{x}_{\cdot k}, \boldsymbol{p}_{\cdot k}, \boldsymbol{r}_{\cdot k}, C_k)$$

$$0 \le \mathcal{R}_j \le S_j \le \max_{k \in \mathcal{K}} \{(\mathcal{D}_j - p_{jk})\, x_{jk}\}$$

$$x_{jk} \in \{0, 1\} \qquad \forall j \in \mathcal{J},\ \forall k \in \mathcal{K}$$

$$S_j \in \mathbb{Z} \qquad \forall j \in \mathcal{J}$$

x_{ij} = 1 if job j is assigned to resource i

all jobs assigned to one resource

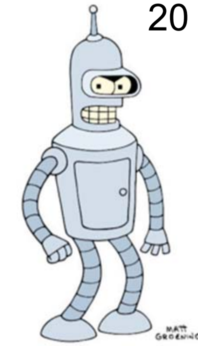
resource capacity

time windows

Tends not to work too well
(if goal is finding and proving optimality).
Why?

University of Toronto
Mechanical & Industrial Engineering

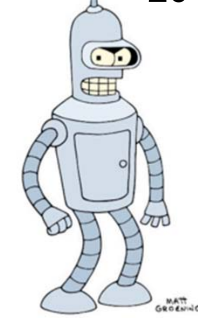
[Heinz & B. 2012] *CPAIOR*, 211-227, 2012.



LBBD

Global Model

[Hooker & Ottosson 2003] *Mathematical Programming,* **96**, 33-60, 2003.

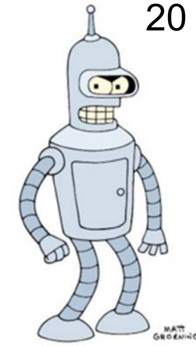


LBBD

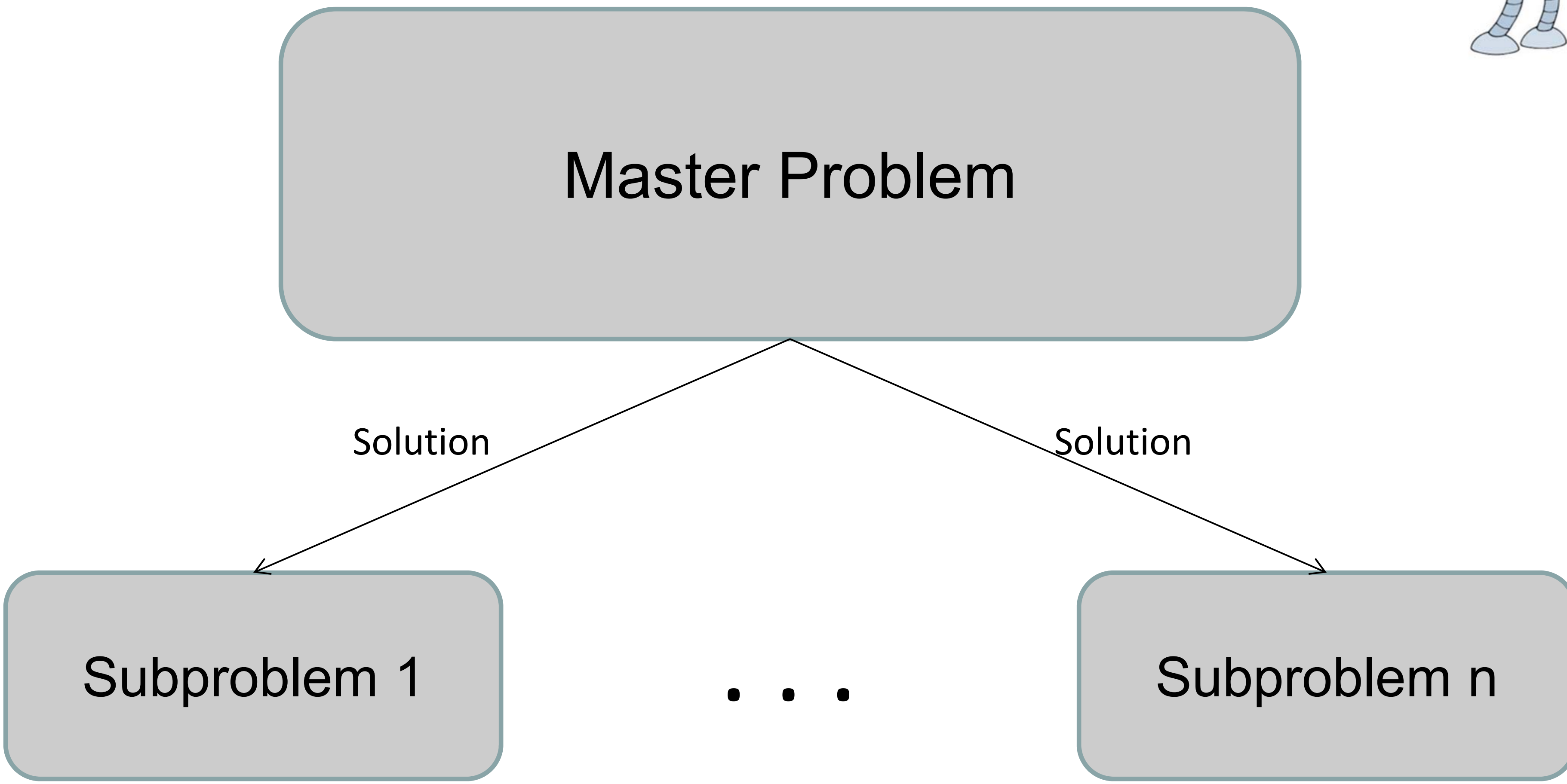
Master Problem

Subproblem 1 . . . Subproblem n

[Hooker & Ottosson 2003] *Mathematical Programming,* **96**, 33-60, 2003.

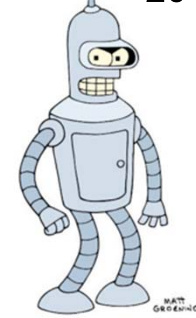


LBBD

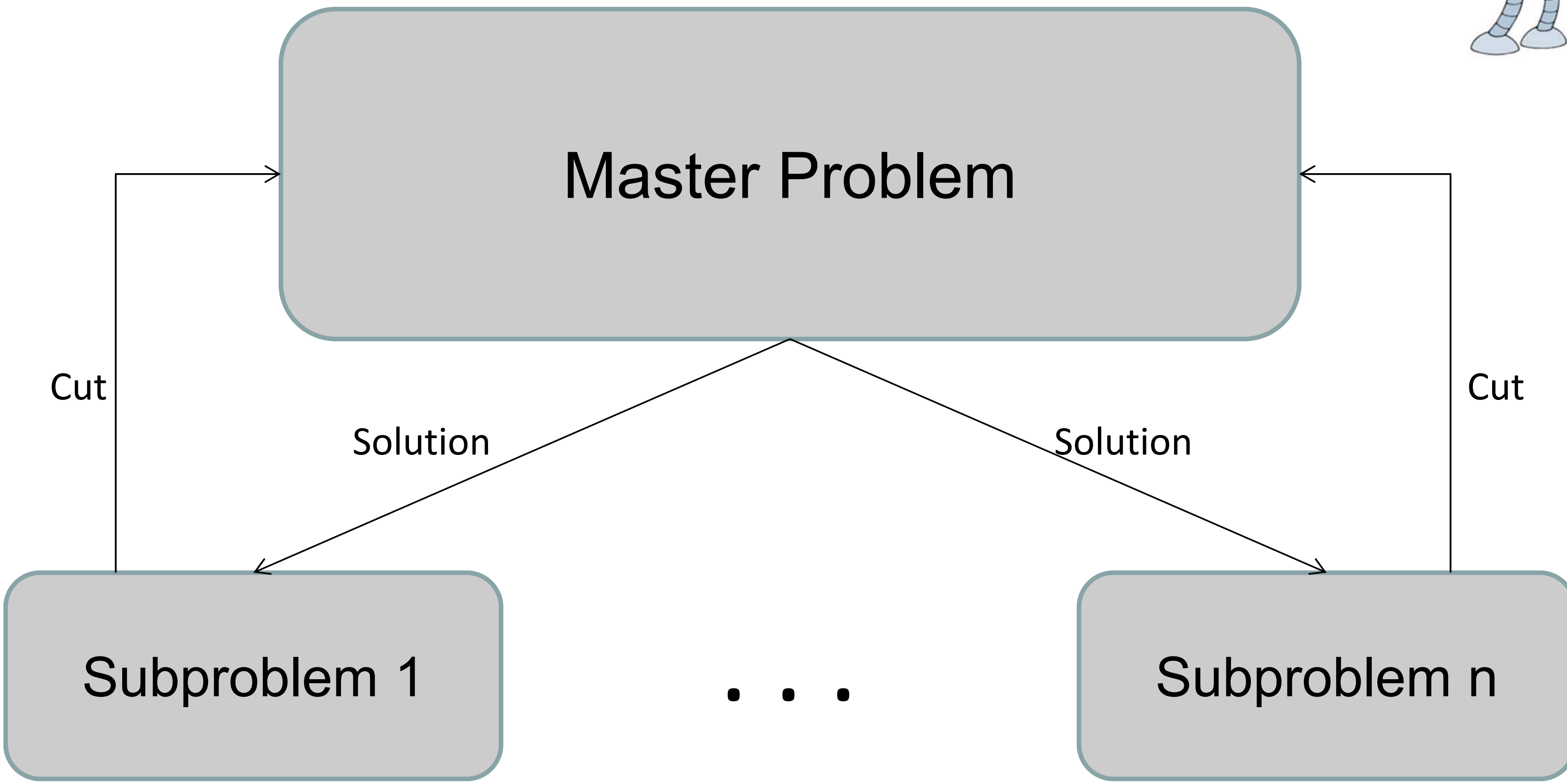


University of Toronto
Mechanical & Industrial Engineering

[Hooker & Ottosson 2003] *Mathematical Programming*, **96**, 33-60, 2003.

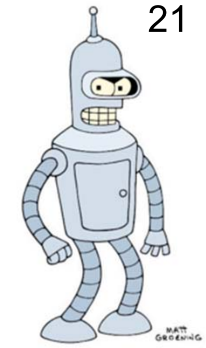


LBBD



[Hooker & Ottosson 2003] *Mathematical Programming*, **96**, 33-60, 2003.

LBBD



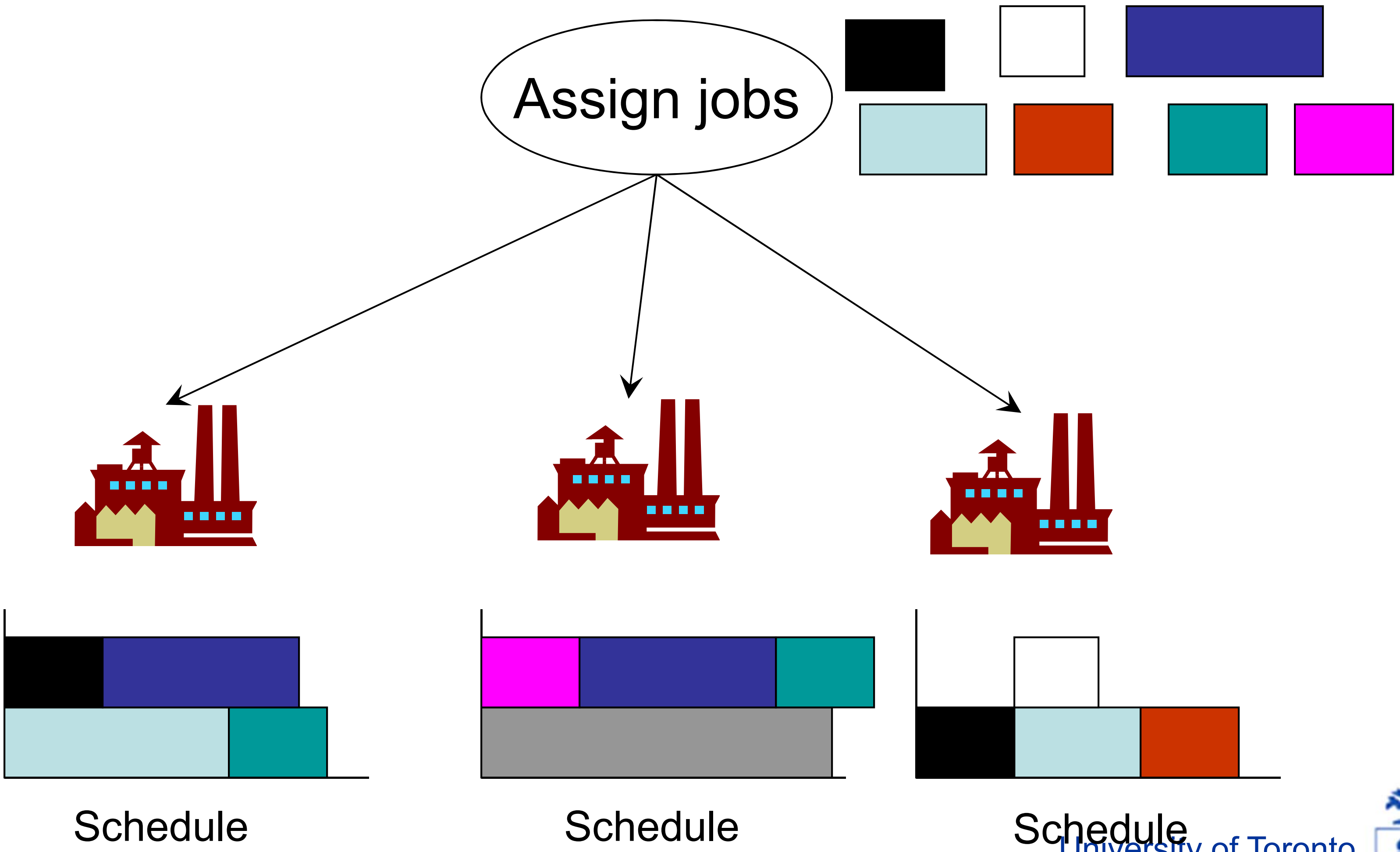
- Partition problem into
  - Master problem with decision variables, *y*
  - Sub-problem(s) with decision variables, *x*
  - When the *y*'s are fixed (to say, $\hat{y}$), sub-problems are formed
- MP & SP do not have to be any particular form (e.g., IP/LP, IP/CP)
- Each sub-problem is an inference dual
  - What is the max. LB that can be inferred assuming $y = \hat{y}$?

Making LBBD Work

- Sub-problem relaxation
  - MP solving needs to have some guidance or else it just enumerates all MP solutions
- Strong & cheap cuts
  - Cuts should remove more than just the current MP solution

Questions?

Resource Allocation & Scheduling



LBBD Master (MIP)

$$
\begin{aligned}
\min \quad & \sum_{k\in\mathcal{K}} \sum_{j\in\mathcal{J}} c_{jk}\, x_{jk} & \\
\text{s.t.} \quad & \sum_{k\in\mathcal{K}} x_{jk} = 1 & \forall j \in \mathcal{J} \\
& \sum_{j\in\mathcal{J}} x_{jk}\, p_{jk} r_{jk} \le \hat{C}_k & \forall k \in \mathcal{K} \\
& \sum_{j\in\mathcal{J}_{hk}} (1 - x_{jk}) \ge 1 & \forall k \in \mathcal{K},\ h \in [H-1] \\
& x_{kj} \in \{0,1\} & \forall k \in \mathcal{K},\ \forall j \in \mathcal{J},
\end{aligned}
$$

with $\hat{C}_k = C_k \cdot (\max_{j\in\mathcal{J}}\{\mathcal{D}_j\} - \min_{j\in\mathcal{J}}\{\mathcal{R}_j\})$.

LBBD Master (MIP)

Minimize resource assignment cost

$$\min \sum_{k\in\mathcal{K}} \sum_{j\in\mathcal{J}} c_{jk}\, x_{jk}$$

$$\text{s.t.} \quad \sum_{k\in\mathcal{K}} x_{jk} = 1 \qquad \forall j \in \mathcal{J}$$

$$\sum_{j\in\mathcal{J}} x_{jk}\, p_{jk} r_{jk} \leq \hat{C}_k \qquad \forall k \in \mathcal{K}$$

$$\sum_{j\in\mathcal{J}_{hk}} (1 - x_{jk}) \geq 1 \qquad \forall k \in \mathcal{K},\ h \in [H-1]$$

$$x_{kj} \in \{0, 1\} \qquad \forall k \in \mathcal{K},\ \forall j \in \mathcal{J},$$

with $\hat{C}_k = C_k \cdot (\max_{j\in\mathcal{J}}\{\mathcal{D}_j\} - \min_{j\in\mathcal{J}}\{\mathcal{R}_j\})$.

nto
ing

LBBD Master (MIP)

Minimize resource assignment cost

$$\min \quad \sum_{k \in \mathcal{K}} \sum_{j \in \mathcal{J}} c_{jk} x_{jk}$$

Each activity is assigned to one resource

$$\text{s.t.} \quad \sum_{k \in \mathcal{K}} x_{jk} = 1$$

$$\sum_{j \in \mathcal{J}} x_{jk} p_{jk} r_{jk} \leq \hat{C}_k \quad \forall k \in \mathcal{K}$$

$$\sum_{j \in \mathcal{J}_{hk}} (1 - x_{jk}) \geq 1 \quad \forall k \in \mathcal{K},\ h \in [H-1]$$

$$x_{kj} \in \{0, 1\} \quad \forall k \in \mathcal{K},\ \forall j \in \mathcal{J},$$

with $\hat{C}_k = C_k \cdot (\max_{j \in \mathcal{J}}\{\mathcal{D}_j\} - \min_{j \in \mathcal{J}}\{\mathcal{R}_j\})$.

nto
ing

LBBD Master (MIP)

$$\min \quad \sum_{k \in \mathcal{K}} \sum_{j \in \mathcal{J}} c_{jk} \, x_{jk}$$

Minimize resource assignment cost

$$\text{s.t.} \quad \sum_{k \in \mathcal{K}} x_{jk} = 1$$

Each activity is assigned to one resource

$$\sum_{j \in \mathcal{J}} x_{jk} \, p_{jk} r_{jk} \le \hat{C}_k \qquad \forall k$$

Sub-problem relaxation
(Can we do better?)

$$\sum_{j \in \mathcal{J}_{hk}} (1 - x_{jk}) \ge 1 \qquad \forall k \in \mathcal{K}, \; h \in [H-1]$$

$$x_{kj} \in \{0, 1\} \qquad \forall k \in \mathcal{K}, \; \forall j \in \mathcal{J},$$

with $\hat{C}_k = C_k \cdot (\max_{j \in \mathcal{J}}\{\mathcal{D}_j\} - \min_{j \in \mathcal{J}}\{\mathcal{R}_j\})$.

LBBD Master (MIP)

$$\min \quad \sum_{k \in \mathcal{K}} \sum_{j \in \mathcal{J}} c_{jk} x_{jk}$$

Minimize resource assignment cost

$$\text{s. t.} \quad \sum_{k \in \mathcal{K}} x_{jk} = 1$$

Each activity is assigned to one resource

$$\sum_{j \in \mathcal{J}} x_{jk} p_{jk} r_{jk} \leq \hat{C}_k \qquad \forall k$$

Sub-problem relaxation
(Can we do better?)

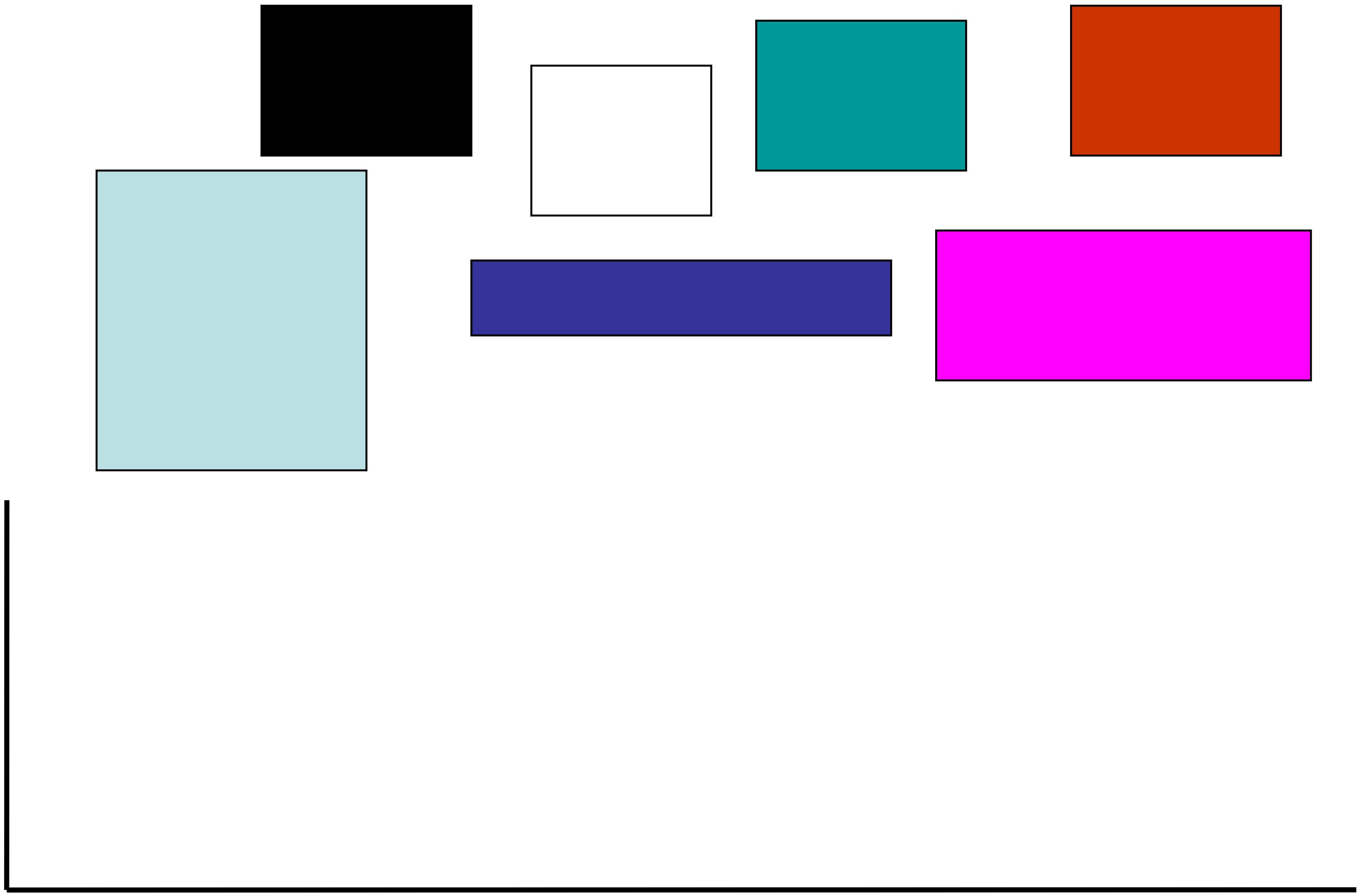
$$\sum_{j \in \mathcal{J}_{hk}} (1 - x_{jk}) \geq 1 \qquad \forall k \in \mathcal{K},\ h \in [H-1]$$

Benders cut
(Can we do better?)

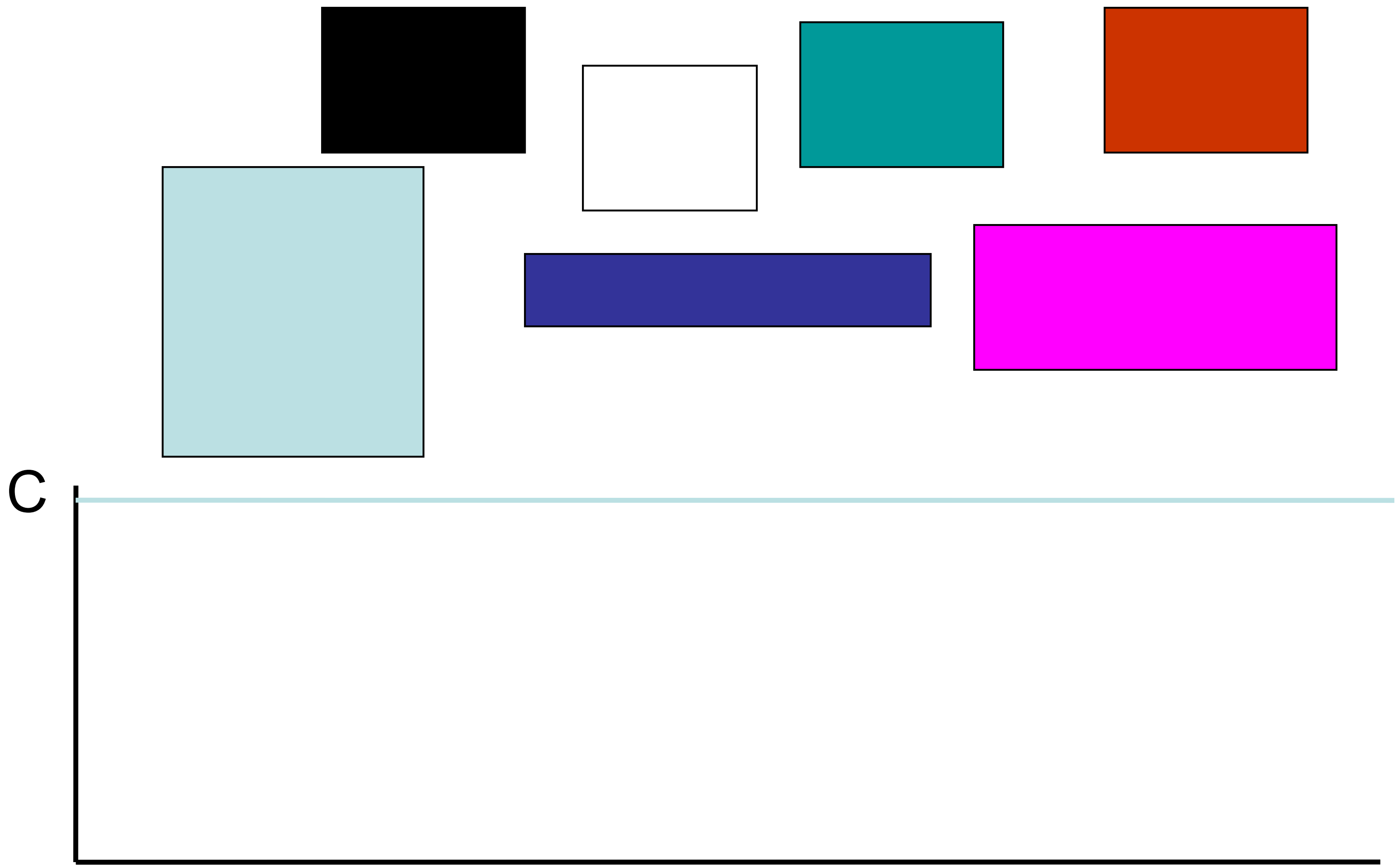
$$x_{kj} \in \{0, 1\} \qquad \forall k \in \mathcal{K},\ \forall j \in \mathcal{J},$$

with $\hat{C}_k = C_k \cdot (\max_{j \in \mathcal{J}}\{\mathcal{D}_j\} - \min_{j \in \mathcal{J}}\{\mathcal{R}_j\})$.

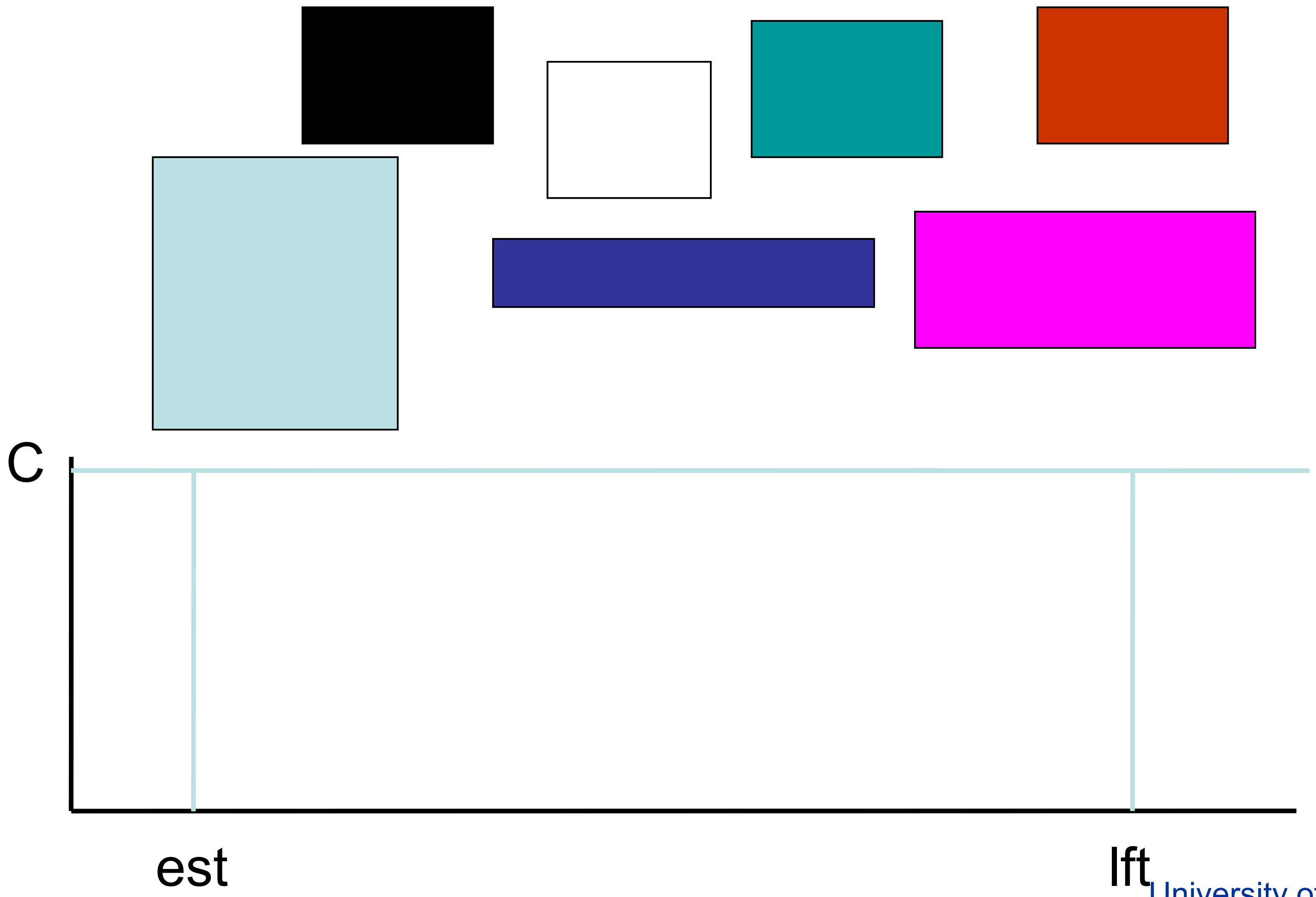
Sub-problem Relaxation



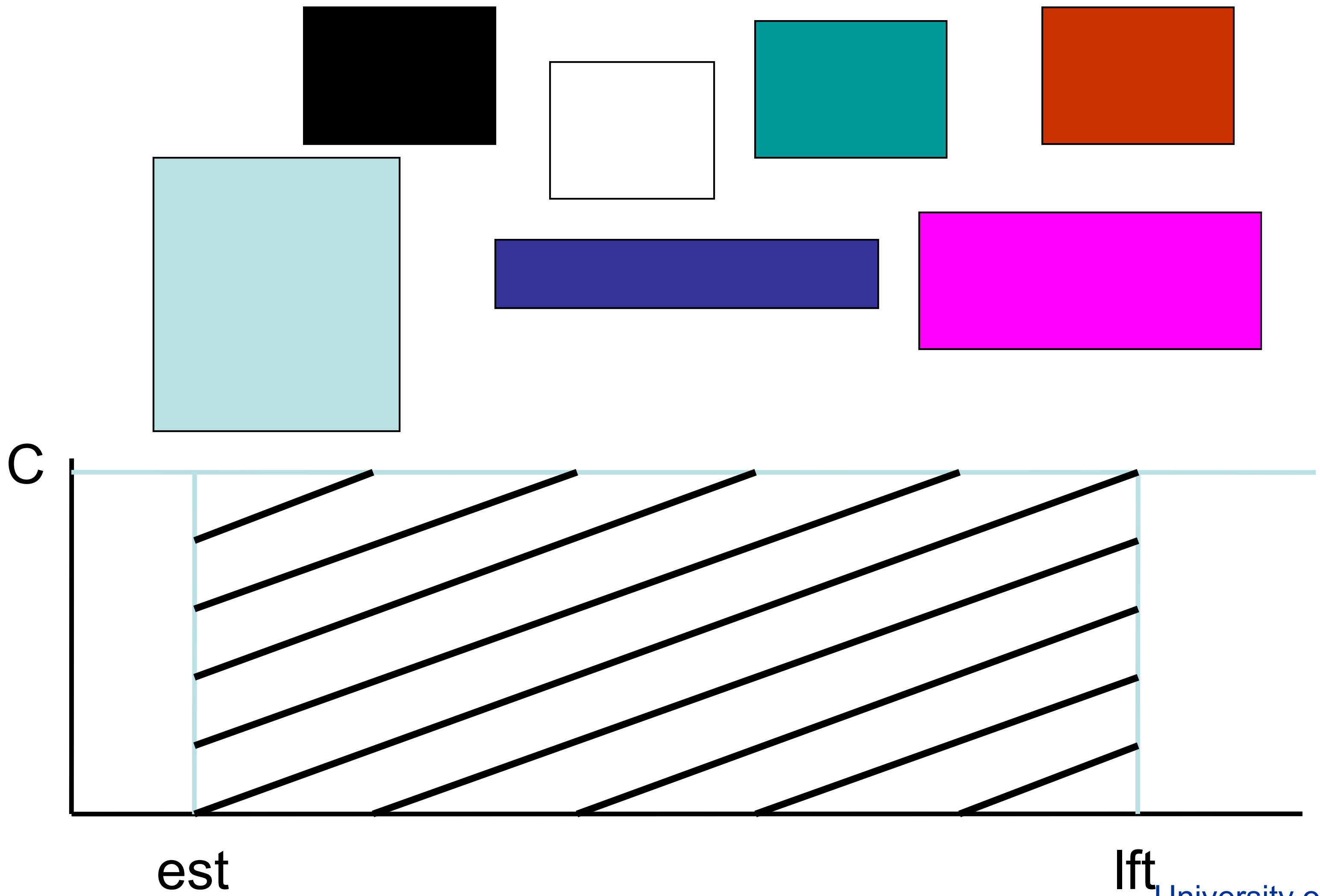
Sub-problem Relaxation



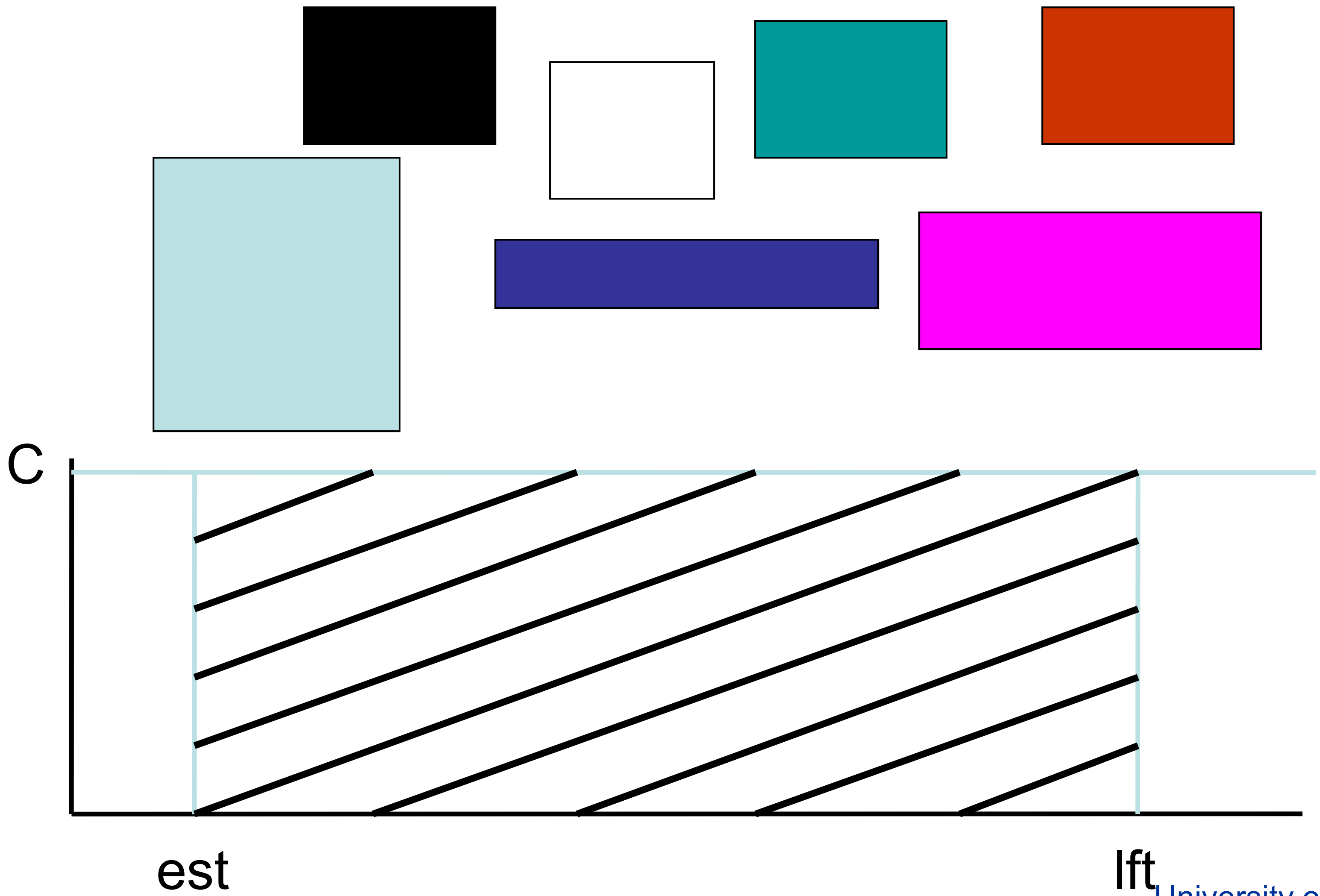
Sub-problem Relaxation



Sub-problem Relaxation



Sub-problem Relaxation



Benders Cut

$$\sum_{j \in \mathcal{J}_{hk}} (1 - x_{jk}) \geq 1 \qquad \forall k \in \mathcal{K},\ h \in [H-1]$$

- Do not allow same assignment of activities (or a superset) to be assigned to the same resource
- Gets inserted into the master problem!

Benders Cut

Counter for the iterations

$$\sum_{j \in \mathcal{J}_{hk}} (1 - x_{jk}) \geq 1 \qquad \forall k \in \mathcal{K},\ h \in [H-1]$$

- Do not allow same assignment of activities (or a superset) to be assigned to the same resource
- Gets inserted into the master problem!

Benders Cut

Counter for the iterations

$$\sum_{j \in \mathcal{J}_{hk}} (1 - x_{jk}) \geq 1 \qquad \forall k \in \mathcal{K}, h \in [H-1]$$

The set of jobs assigned to resource k in iteration h.

- Do not allow same assignment of activities (or a superset) to be assigned to the same resource
- Gets inserted into the master problem!

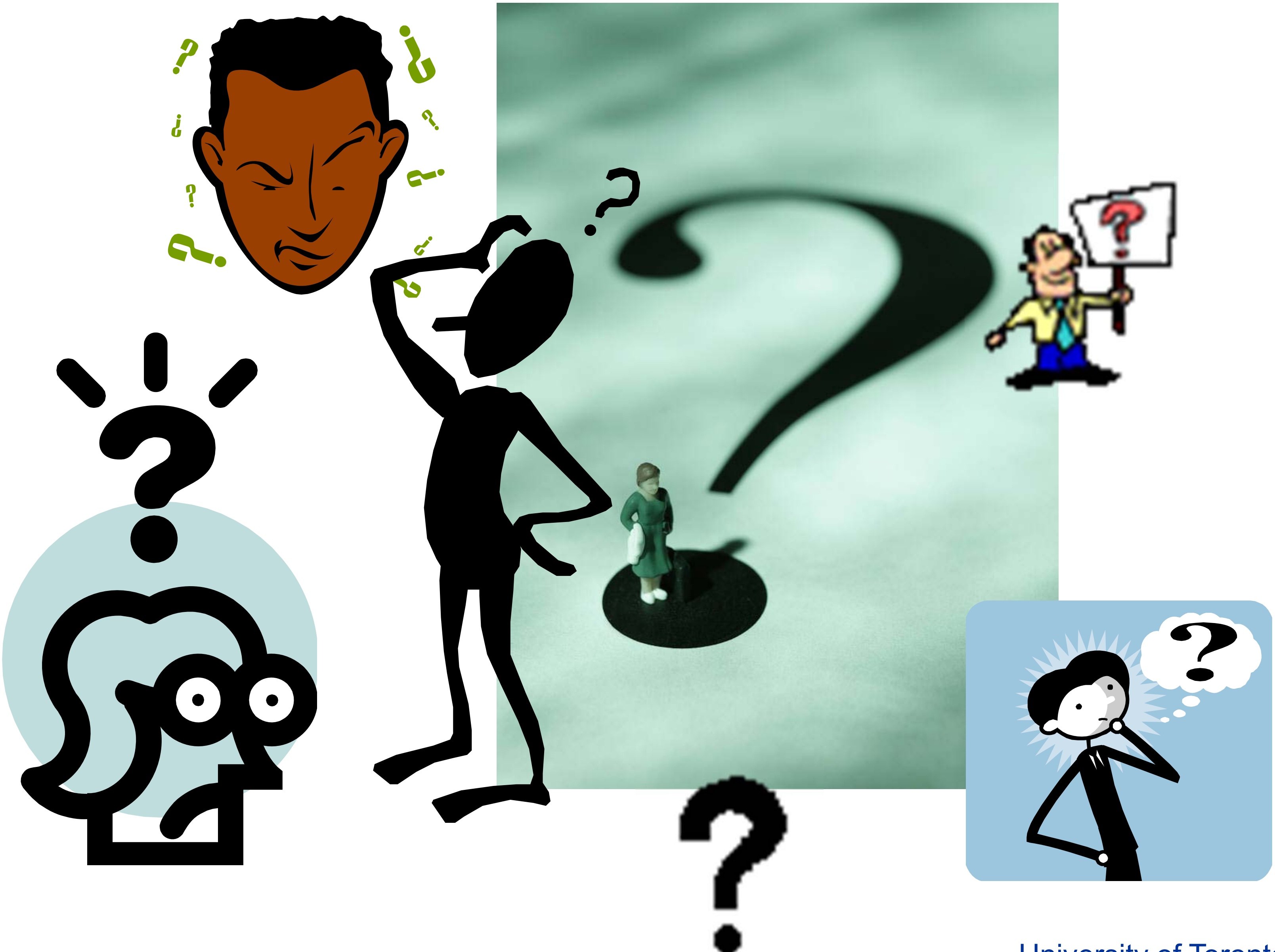
LBBD Subproblem (CP)

$$\texttt{cumulative}(\boldsymbol{S}, \boldsymbol{p}_{\cdot\boldsymbol{k}}, \boldsymbol{r}_{\cdot\boldsymbol{k}}, C_k)$$

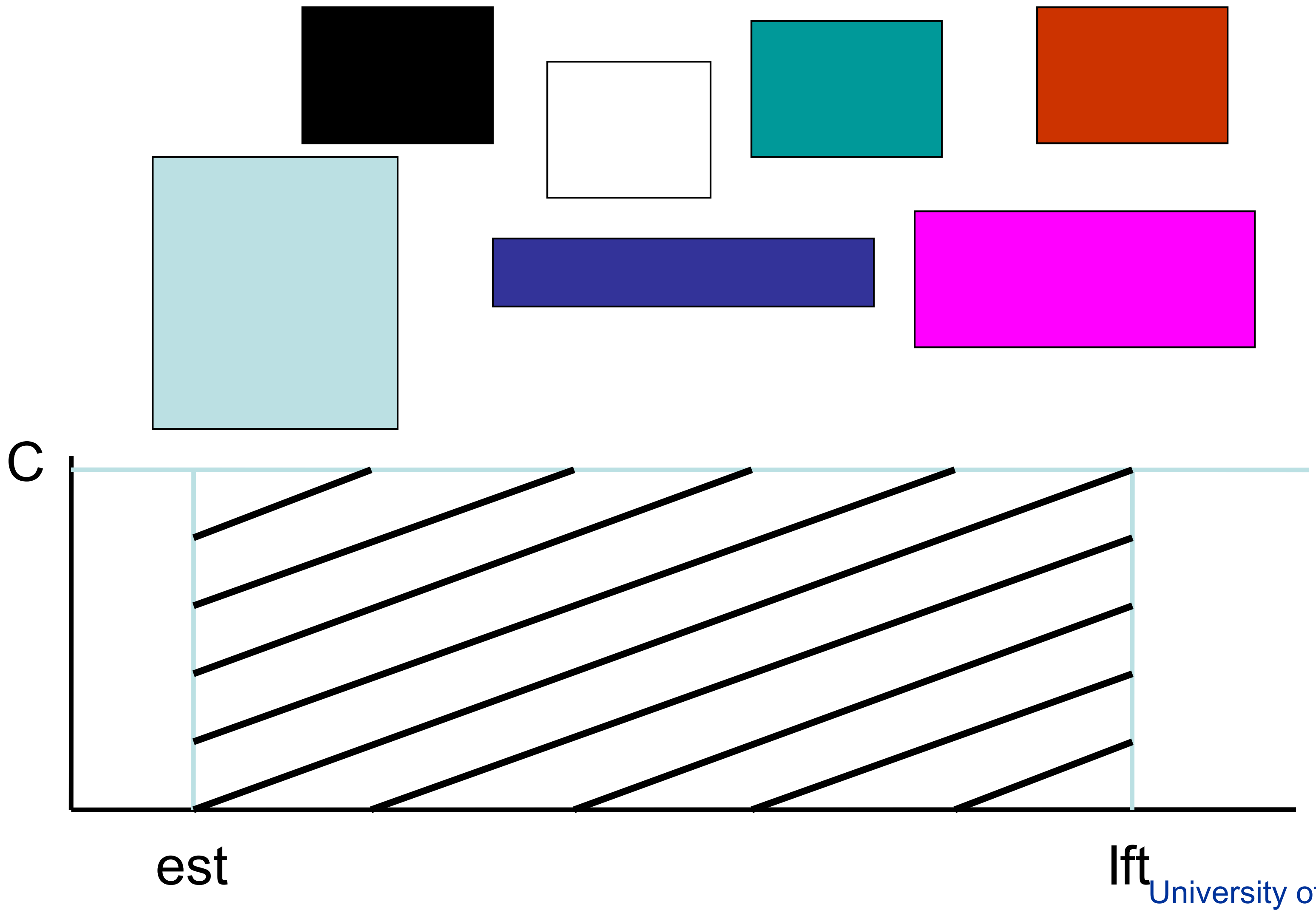
$$\mathcal{R}_j \leq S_j \leq \mathcal{D}_j - p_{jk} \qquad \forall j \in \mathcal{J}_k$$

$$S_j \in \mathbb{Z} \qquad \forall j \in \mathcal{J}_k$$

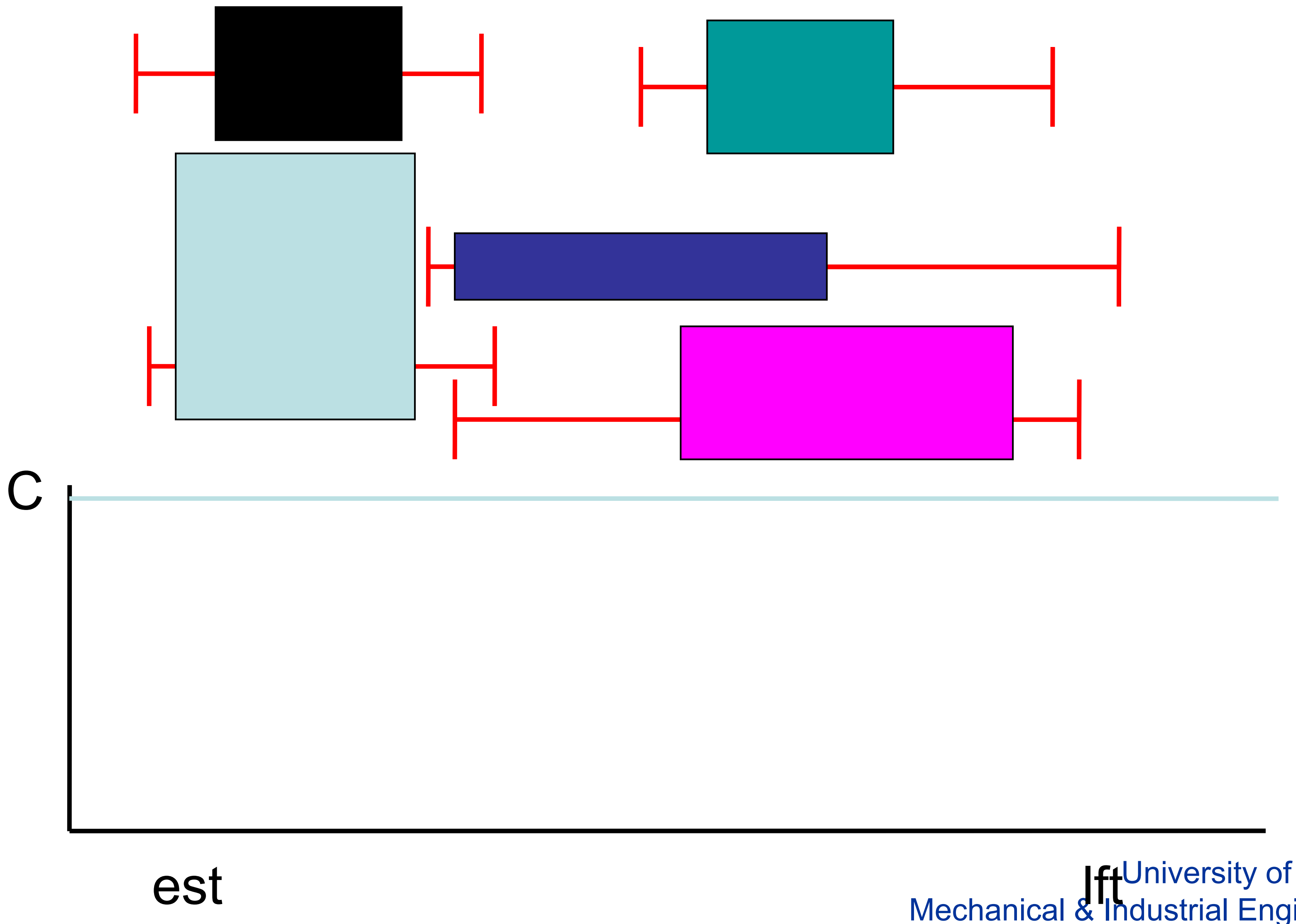
- Single-machine, feasibility problem



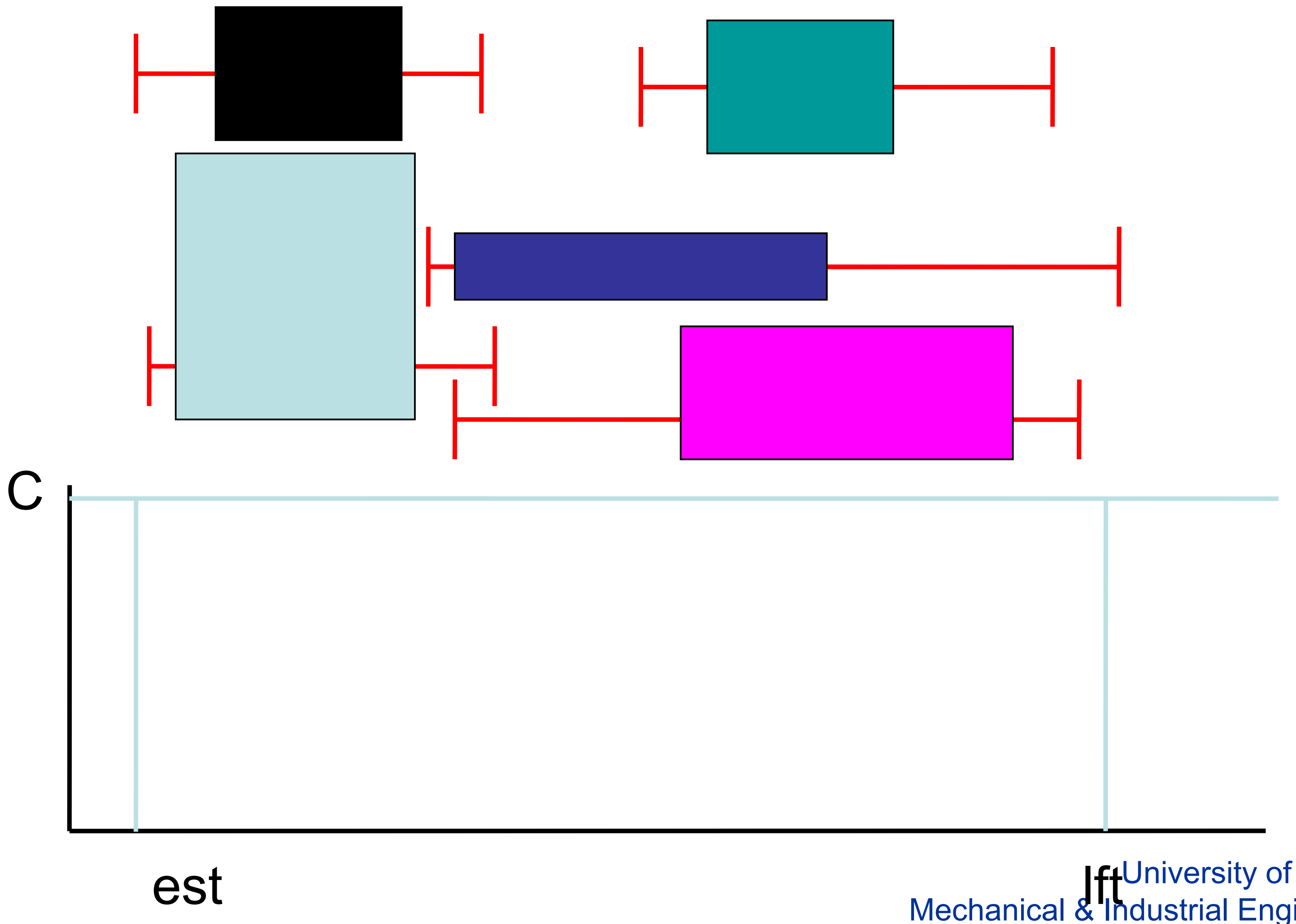
A Tighter Relaxation



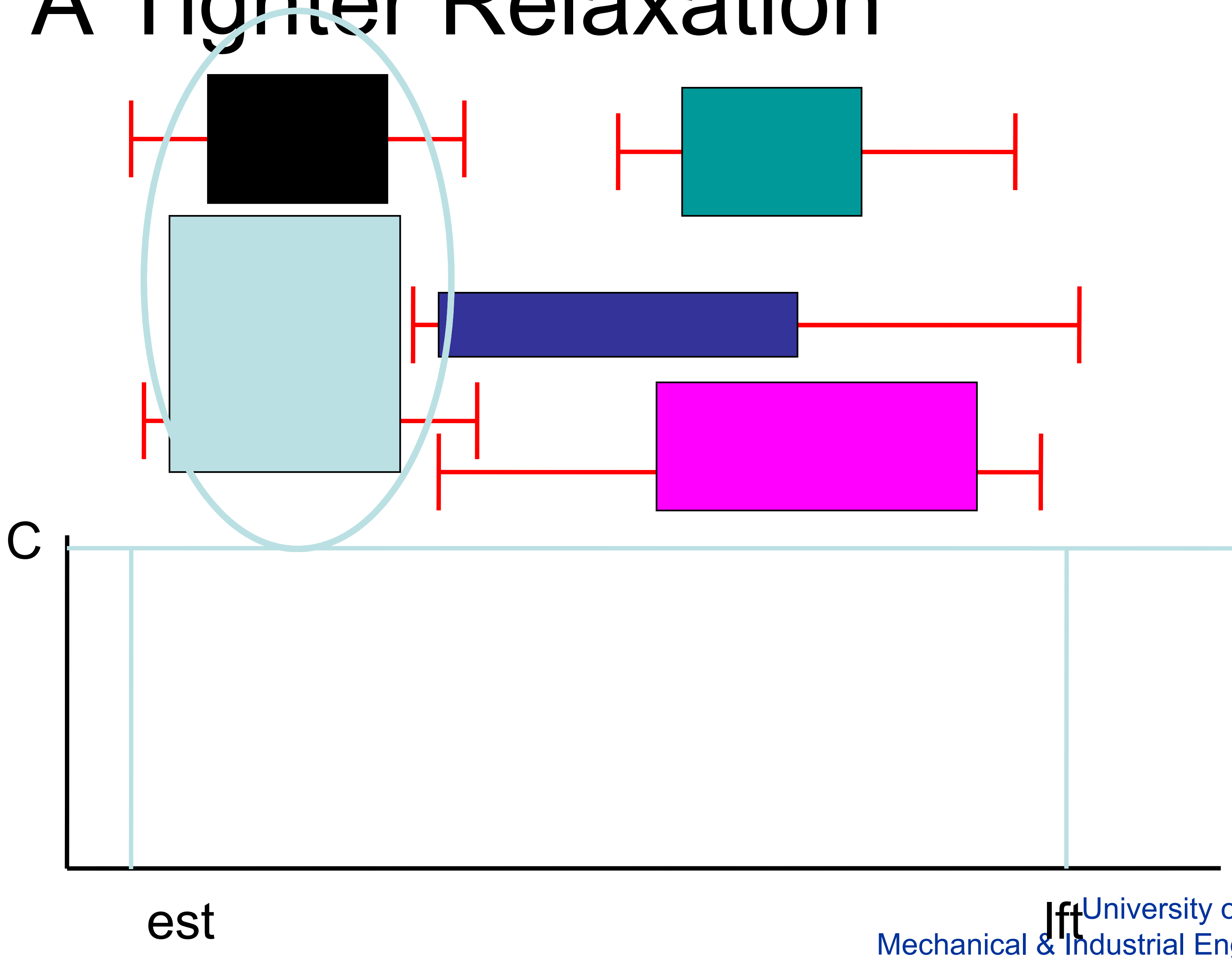
A Tighter Relaxation



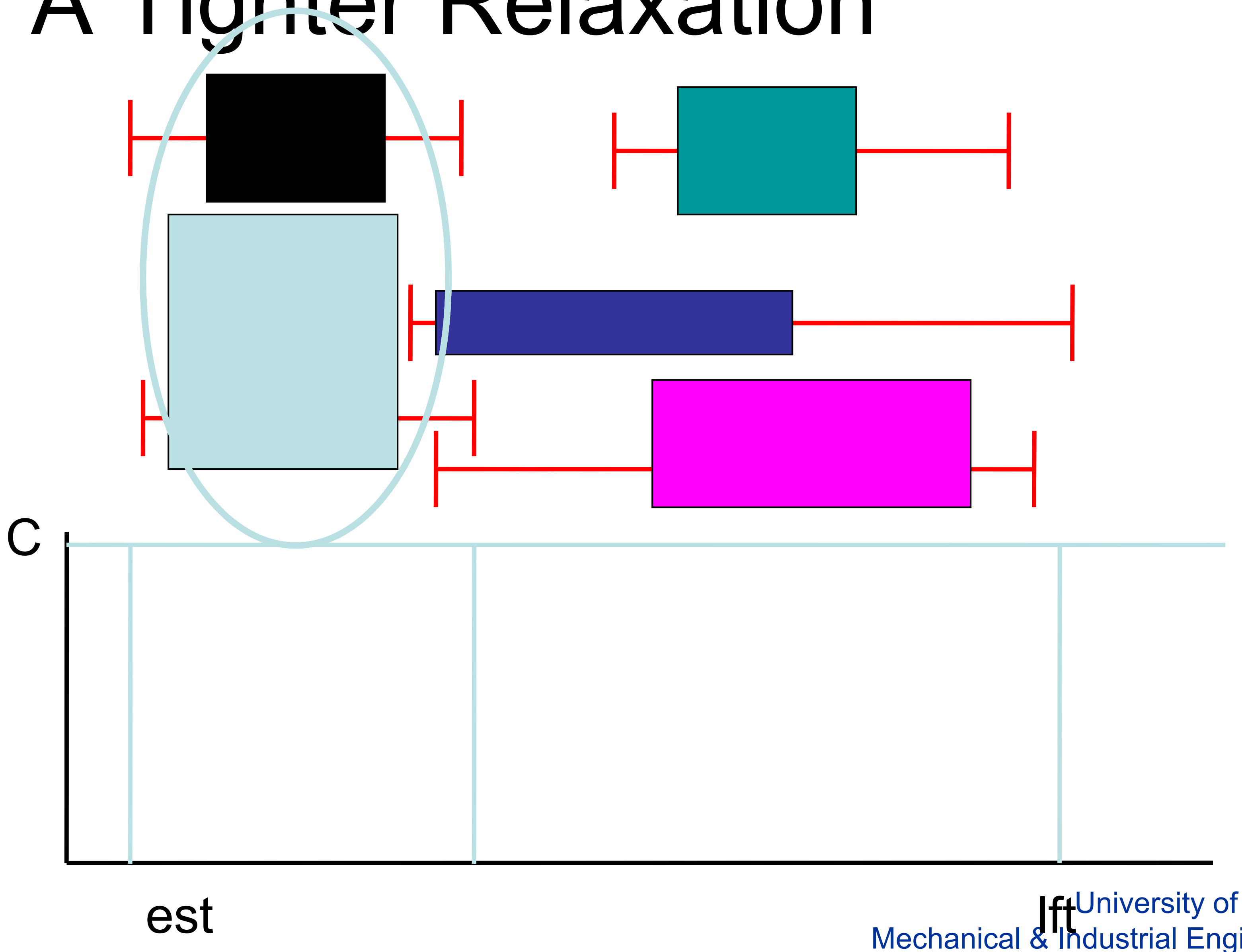
A Tighter Relaxation



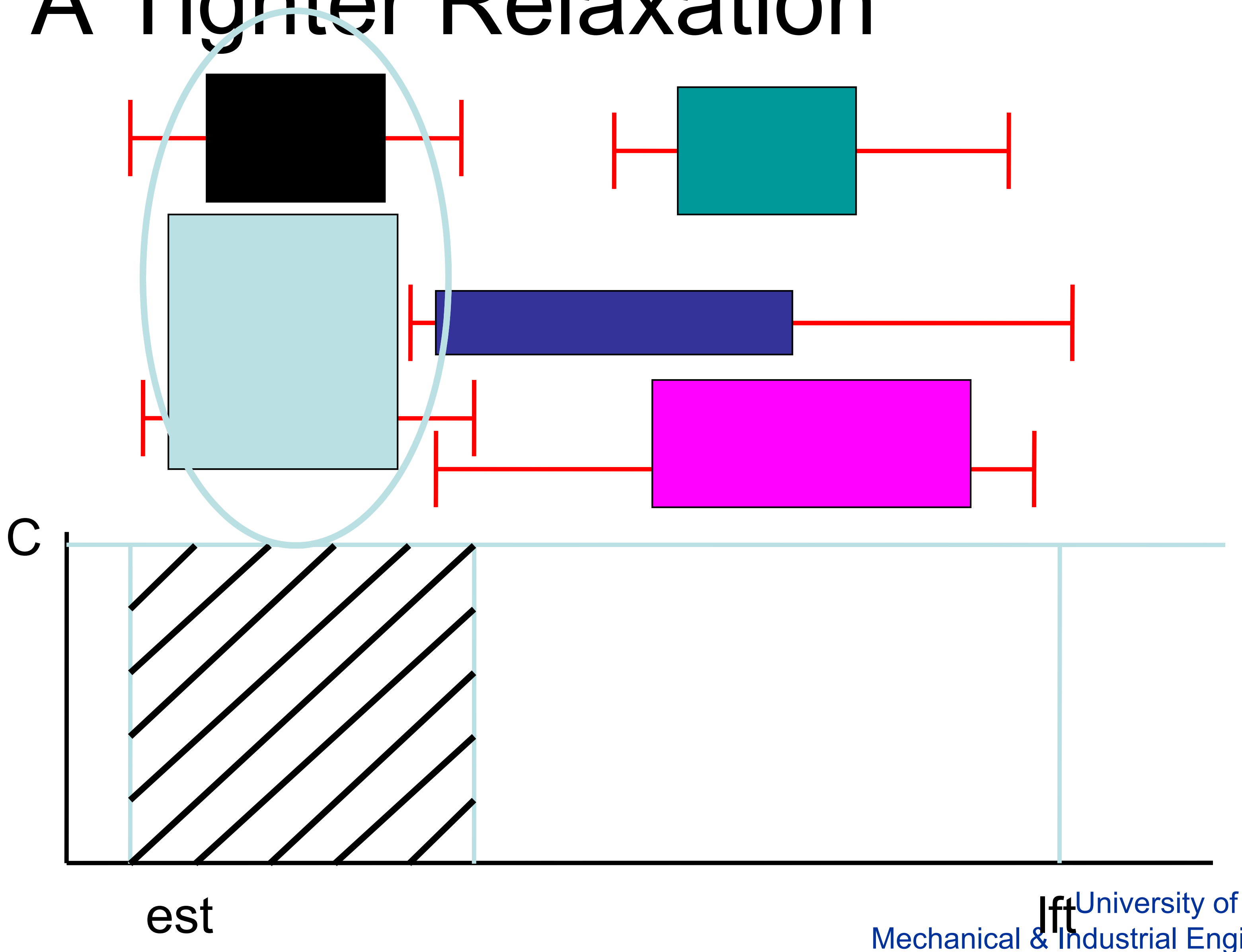
A Tighter Relaxation



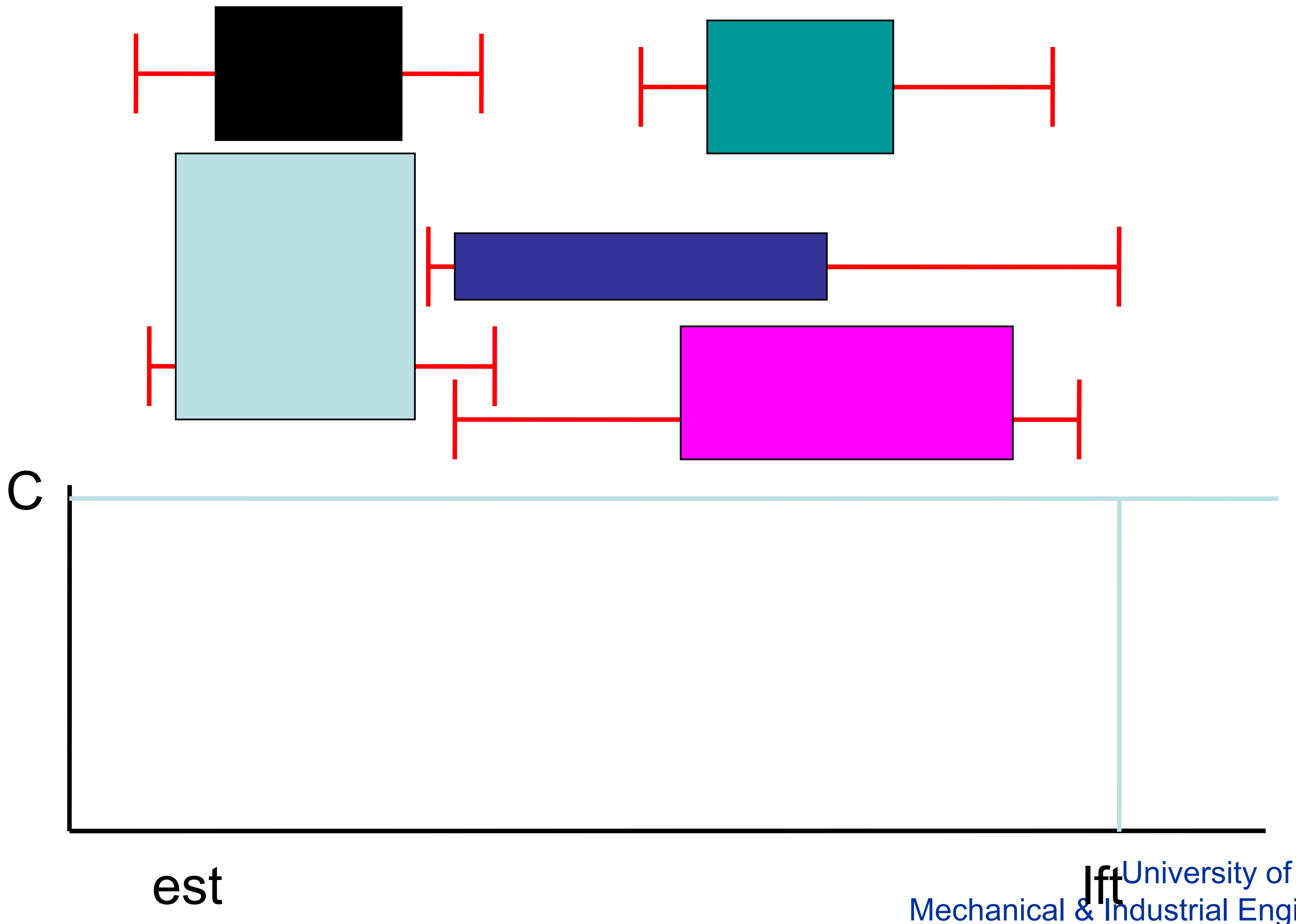
A Tighter Relaxation



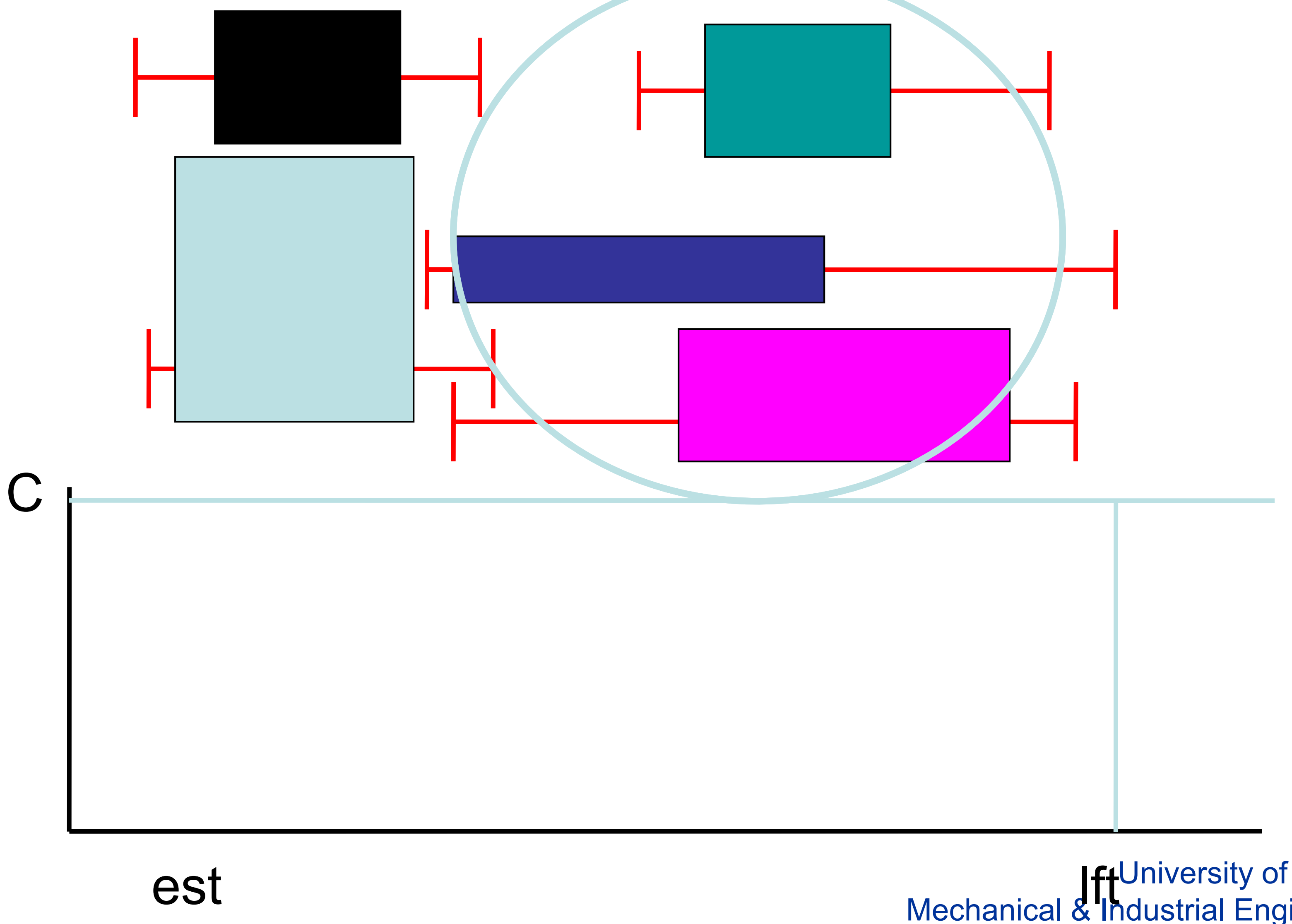
A Tighter Relaxation



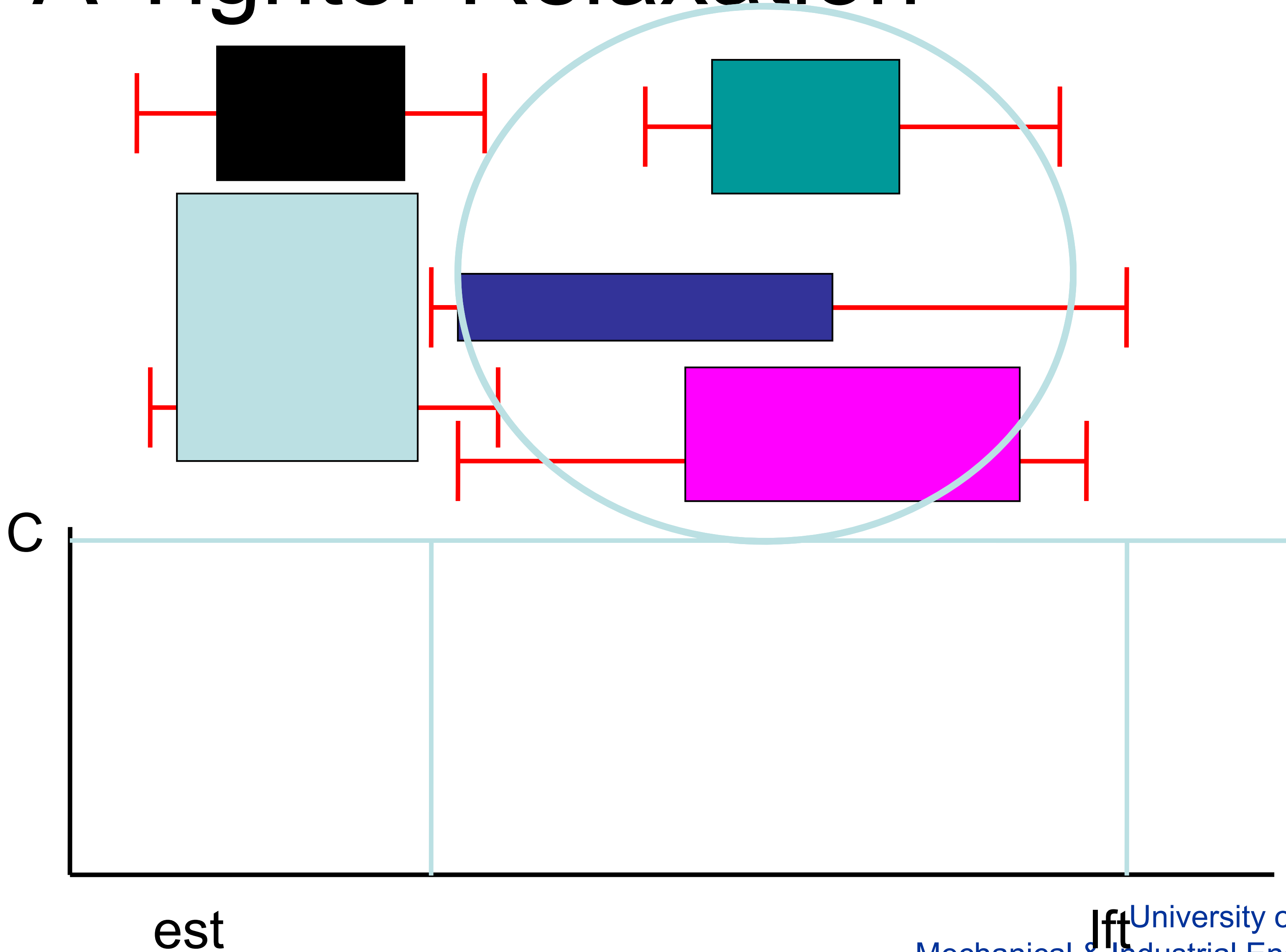
A Tighter Relaxation



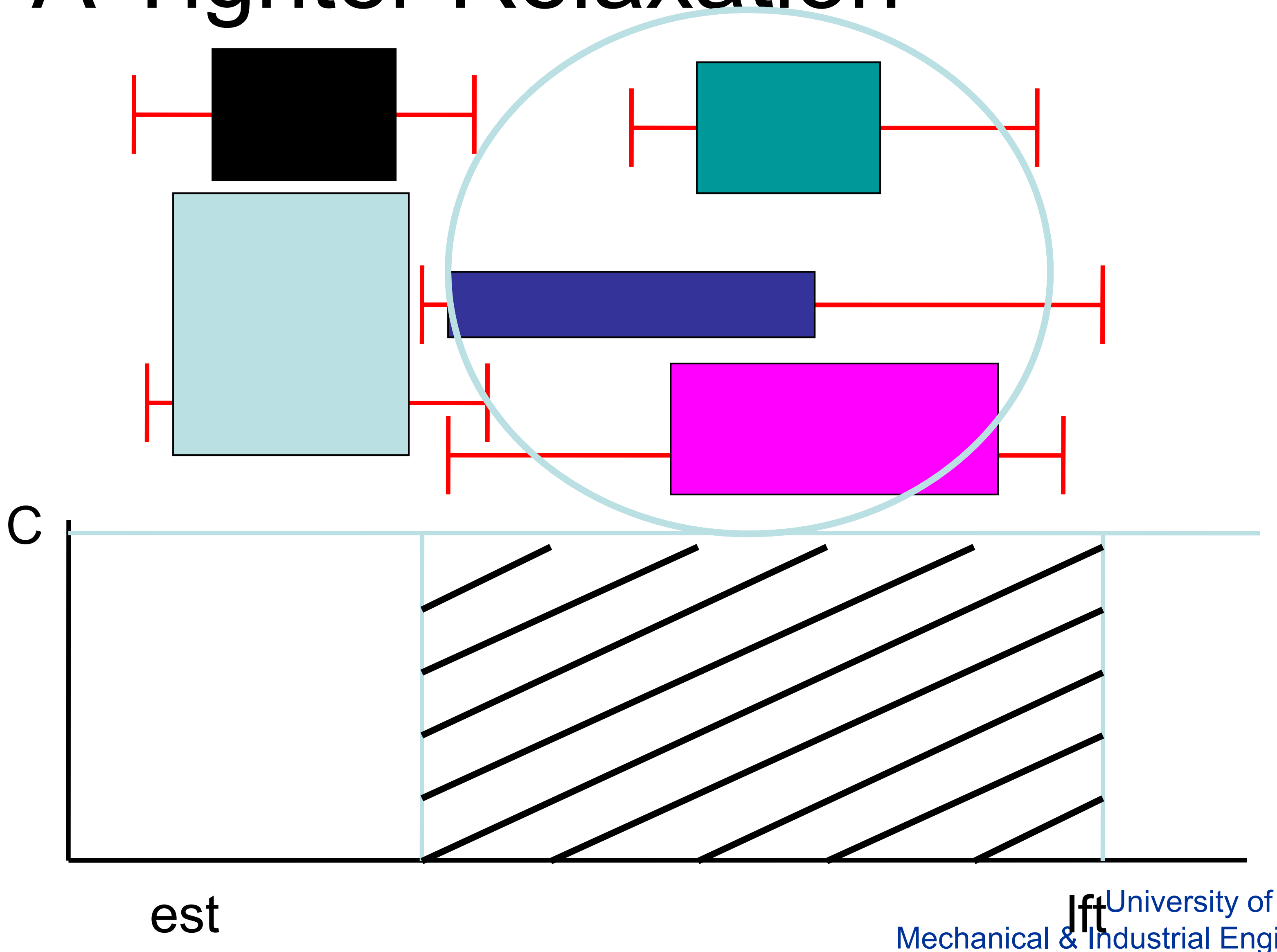
A Tighter Relaxation



A Tighter Relaxation



A Tighter Relaxation



A Tighter Relaxation

[Hooker 2007] *Integrated Methods for Optimization*, 2007.

A Tighter Relaxation

"Single" relaxation

$$\sum_{j \in \mathcal{J}} p_{jk} r_{jk} x_{jk} \leq C_k \cdot (\max_{j \in \mathcal{J}}\{\mathcal{D}_j\} - \min_{j \in \mathcal{J}}\{\mathcal{R}_j\})$$

[Hooker 2007] *Integrated Methods for Optimization*, 2007.

A Tighter Relaxation

“Single” relaxation

$$\sum_{j\in\mathcal{J}} p_{jk} r_{jk}\, x_{jk} \le C_k \cdot (\max_{j\in\mathcal{J}}\{\mathcal{D}_j\} - \min_{j\in\mathcal{J}}\{\mathcal{R}_j\})$$

“Interval” relaxation

$$\sum_{j\in\mathcal{J}(t_1,t_2)} p_{jk} r_{jk}\, x_{jk} \le C_k \cdot (t_2 - t_1) \qquad \forall k \in \mathcal{K},\ \forall (t_1,t_2) \in \mathcal{E}$$

$$\mathcal{E} = \{(t_1,t_2) \mid t_1 \in \mathcal{R}, t_2 \in \mathcal{D}, t_1 < t_2\}$$

$$\mathcal{J}(t_1,t_2) = \{j \in \mathcal{J} \mid t_1 \le \mathcal{R}_j, t_2 \ge \mathcal{D}_j\}$$

[Hooker 2007] *Integrated Methods for Optimization, 2007.*

A Stronger Benders Cut?

$$\sum_{j \in \mathcal{J}_{hk}} (1 - x_{jk}) \geq 1 \qquad \forall k \in \mathcal{K},\ h \in [H-1]$$

A Stronger Benders Cut?

$$\sum_{j \in \mathcal{J}_{hk}} (1 - x_{jk}) \geq 1 \qquad \forall k \in \mathcal{K},\ h \in [H-1]$$

- Repeatedly resolve infeasible sub-problem, removing activities to identify a minimal infeasible subset of J_{hk}

Results?

- Well it is a bit controversial
  - LBBD best for finding and proving optimality
  - MIP best for finding high-quality feasible solutions
  - CIP competitive
  - CP good for finding high-quality feasible, bad for proving optimality

Results?

- Well it is a bit controversial
  - LBBD best for finding and proving optimality
  - MIP best for finding high-quality feasible solutions
  - CIP competitive
  - CP good for finding high-quality feasible, bad for proving optimality

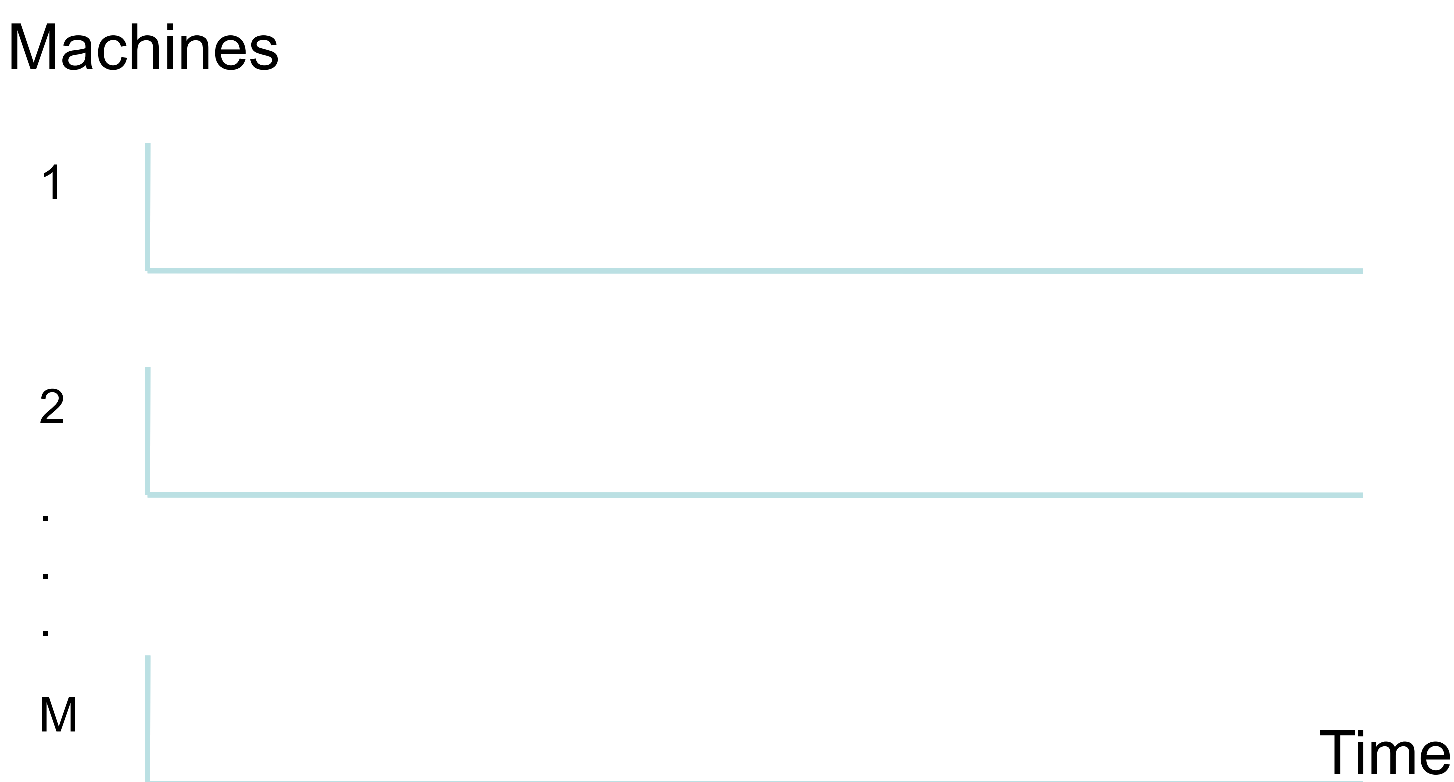
Results?

- Well it is a bit controversial
  - LBBD best for finding and proving optimality
  - MIP best for finding high-quality feasible solutions

Stefan Heinz, Wen-Yang Ku and Chris Beck
Recent improvements using constraint integer programming for resource allocation and scheduling problems

Andre Cire, Elvin Coban and John Hooker
Mixed integer programming vs logic-based Benders decomposition for planning and scheduling

Parallel Machine Scheduling



University of Toronto
Mechanical & Industrial Engineering

[Tran & B. 2012] *ECAI*, 774-779, 2012.

Parallel Machine Scheduling

University of Toronto
Mechanical & Industrial Engineering

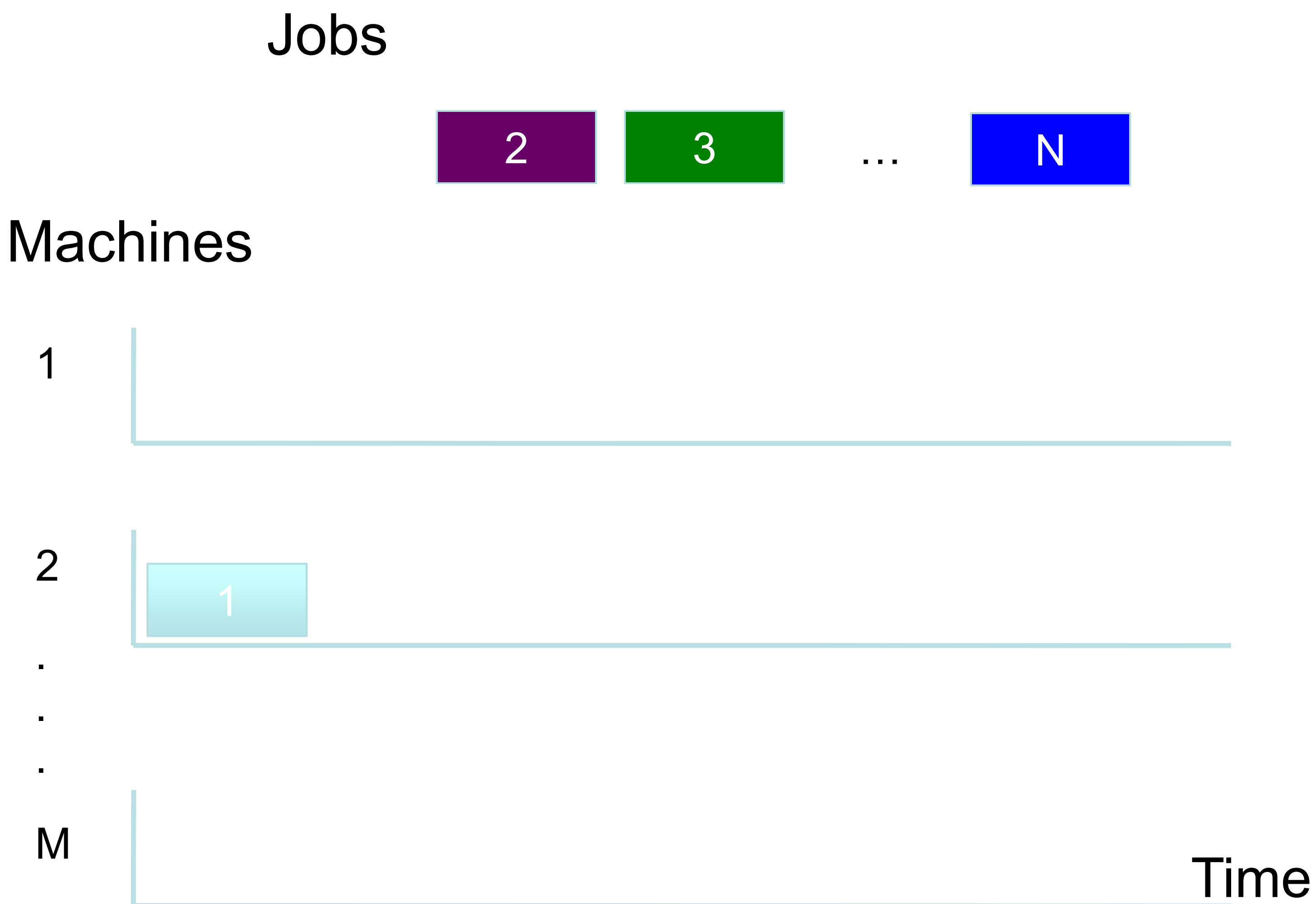
[Tran & B. 2012] *ECAI*, 774-779, 2012.

Parallel Machine Scheduling

University of Toronto
Mechanical & Industrial Engineering

[Tran & B. 2012] *ECAI*, 774-779, 2012.

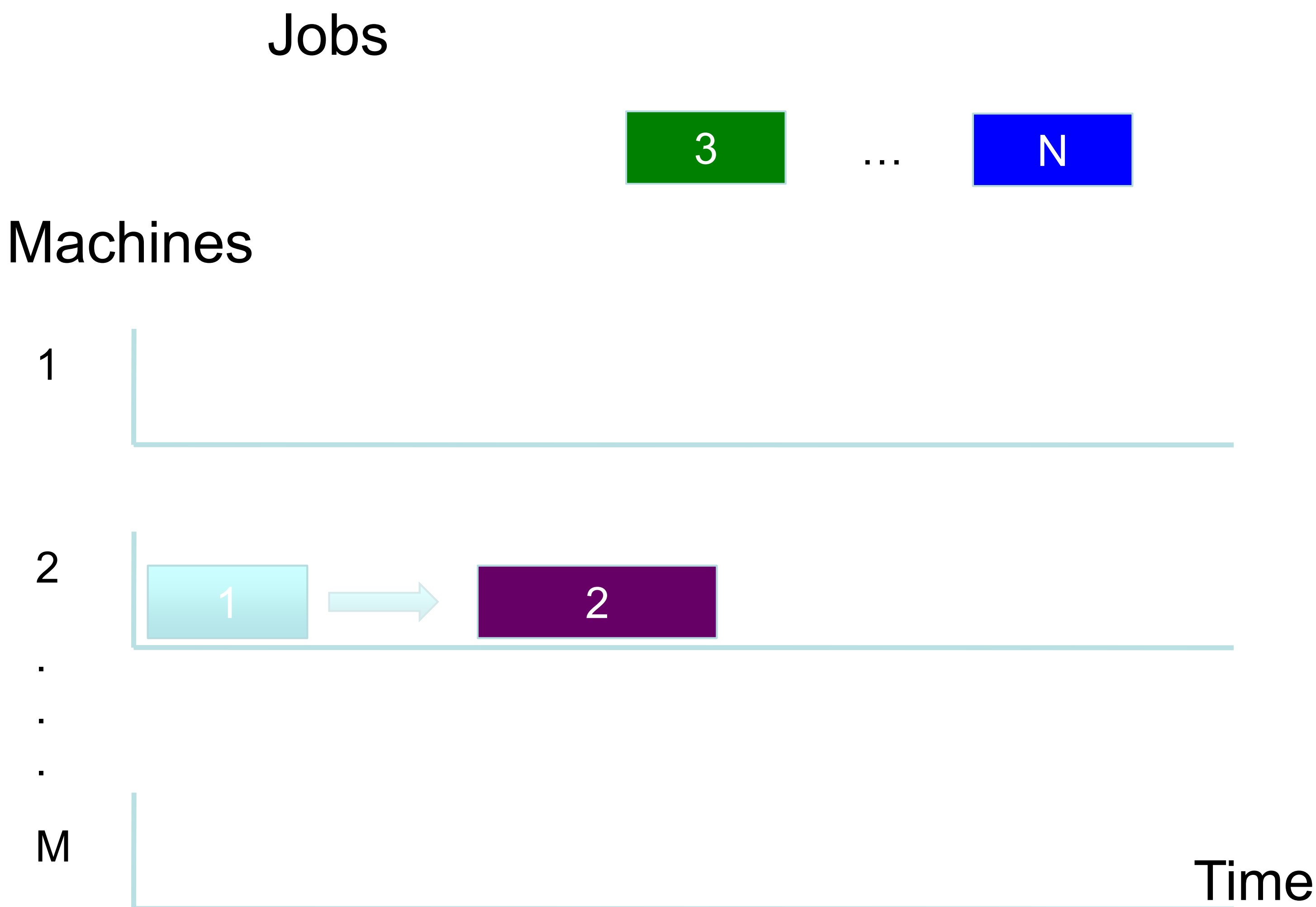
Parallel Machine Scheduling



University of Toronto
Mechanical & Industrial Engineering

[Tran & B. 2012] *ECAI*, 774-779, 2012.

Parallel Machine Scheduling



University of Toronto
Mechanical & Industrial Engineering

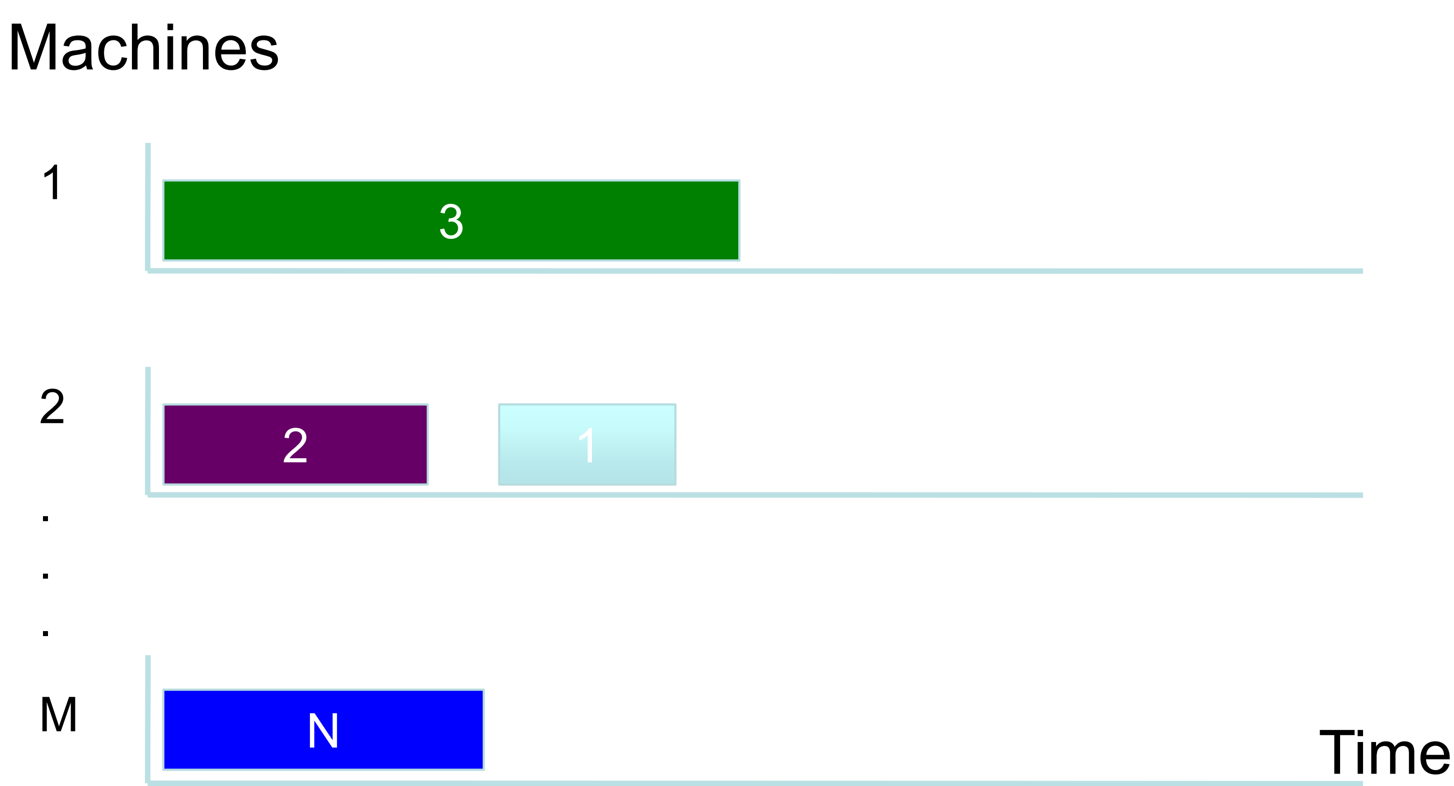
[Tran & B. 2012] *ECAI*, 774-779, 2012.

Parallel Machine Scheduling

University of Toronto
Mechanical & Industrial Engineering

[Tran & B. 2012] *ECAI*, 774-779, 2012.

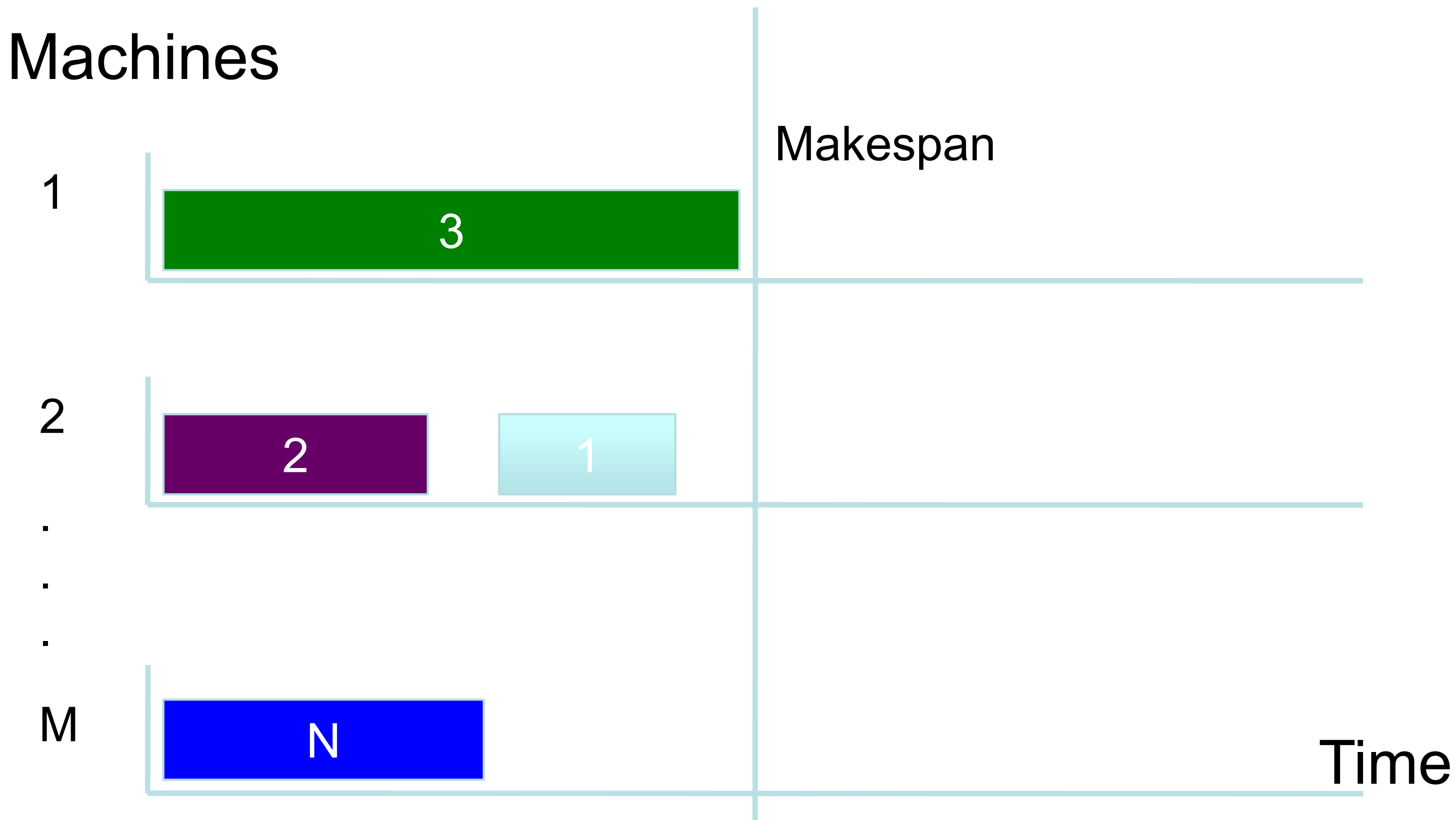
Parallel Machine Scheduling



University of Toronto
Mechanical & Industrial Engineering

[Tran & B. 2012] *ECAI*, 774-779, 2012.

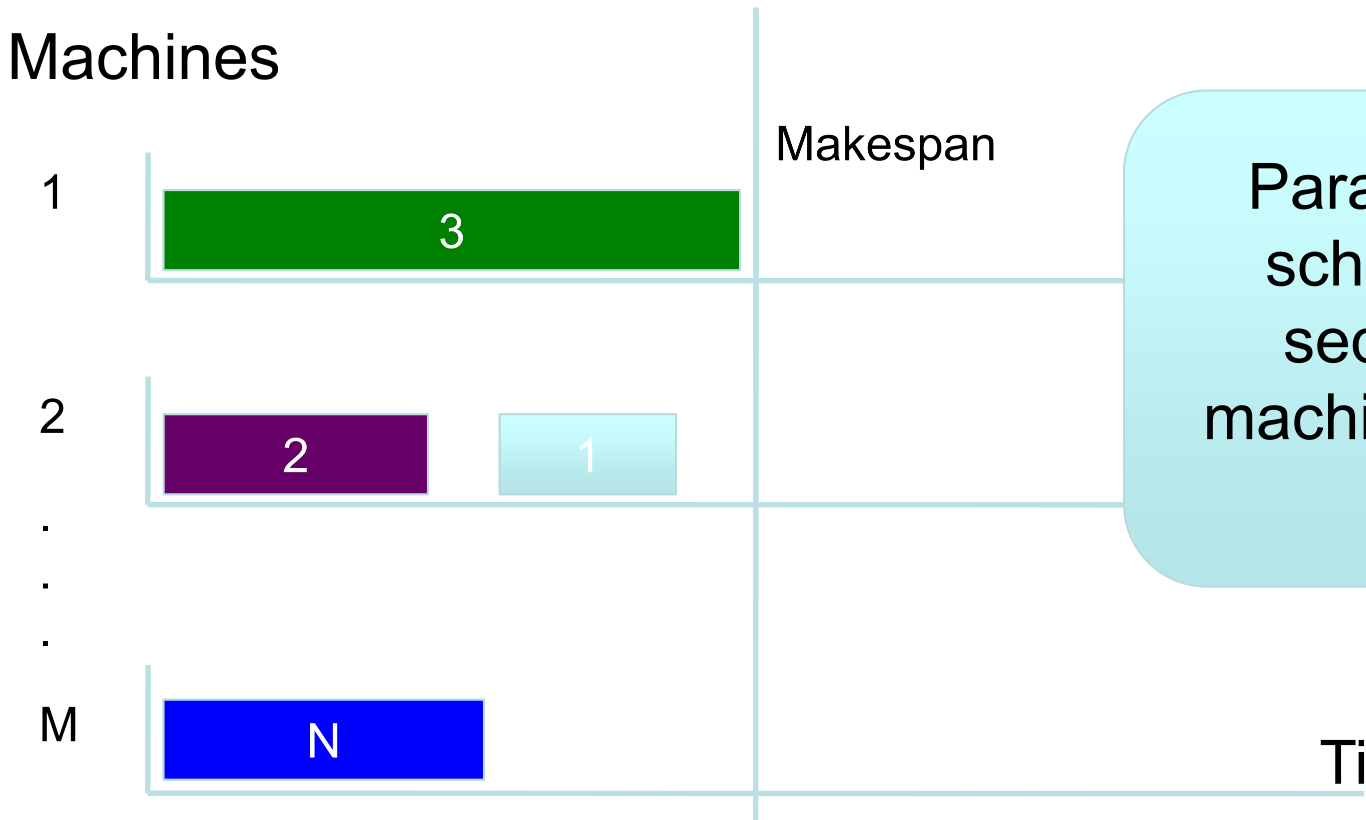
Parallel Machine Scheduling



University of Toronto
Mechanical & Industrial Engineering

[Tran & B. 2012] *ECAI*, 774-779, 2012.

Parallel Machine Scheduling



Parallel machine scheduling with sequence and machine dependent set-ups

[Tran & B. 2012] *ECAI*, 774-779, 2012.

$$\min \quad C_{max}$$

$$\text{s.t.} \quad \sum_{j=0,j\neq k}^{N} \sum_{i=1}^{M} x_{ijk} = 1 \qquad k \in N \qquad (1)$$

$$\sum_{j=0,j\neq h}^{N} x_{ijh} = \sum_{k=0,k\neq h}^{N} x_{ihk} \qquad h \in N;\ i \in M \qquad (2)$$

$$C_k \geq C_j + \sum_{i=1}^{M} x_{ijk}(s_{ijk} + p_{ik}) + V(\sum_{i=1}^{M} x_{ijk} - 1) \qquad j \in N;\ k \in N \qquad (3)$$

$$\sum_{j=0}^{N} x_{i0j} = 1 \qquad i \in M \qquad (4)$$

$$C_j \leq C_{max} \qquad j \in N \qquad (5)$$

$$C_0 = 0 \qquad (6)$$

$$C_j \geq 0 \qquad j \in N \qquad (7)$$

$$x_{ijk} \in \{0,1\} \qquad j,k \in N; i \in M \qquad (8)$$

nto
ing

$$\min \quad C_{max}$$

$$\text{s.t.} \quad \sum_{j=0, j\neq k}^{N} \sum_{i=1}^{M} x_{ijk} = 1$$

x_{ijk} = 1 if k is processed directly after j on machine i

$$\sum_{j=0, j\neq h}^{N} x_{ijh} = \sum_{k=0, k\neq h}^{N} x_{ihk} \qquad h \in N;\ i \in M \qquad (2)$$

$$C_k \geq C_j + \sum_{i=1}^{M} x_{ijk}(s_{ijk} + p_{ik}) + V(\sum_{i=1}^{M} x_{ijk} - 1) \qquad j \in N;\ k \in N \qquad (3)$$

$$\sum_{j=0}^{N} x_{i0j} = 1 \qquad i \in M \qquad (4)$$

$$C_j \leq C_{max} \qquad j \in N \qquad (5)$$

$$C_0 = 0 \qquad (6)$$

$$C_j \geq 0 \qquad j \in N \qquad (7)$$

$$x_{ijk} \in \{0, 1\} \qquad j, k \in N; i \in M \qquad (8)$$

nto
ing

$$\min \quad C_{max}$$

$$\text{s.t.} \quad \sum_{j=0, j \neq k}^{N} \sum_{i=1}^{M} x_{ijk} = 1$$

x_{ijk} = 1 if k is processed directly after j on machine i

$$\sum_{j=0, j \neq h}^{N} x_{ijh} = \sum_{k=0, k \neq h}^{N} x_{ihk}$$

each job preceded and succeeded by at most one other job

$$C_k \geq C_j + \sum_{i=1}^{M} x_{ijk}(s_{ijk} + p_{ik}) + V(\sum_{i=1}^{M} x_{ijk} - 1)$$

$$j \in N;\ k \in N \qquad (3)$$

$$\sum_{j=0}^{N} x_{i0j} = 1 \qquad i \in M \qquad (4)$$

$$C_j \leq C_{max} \qquad j \in N \qquad (5)$$

$$C_0 = 0 \qquad (6)$$

$$C_j \geq 0 \qquad j \in N \qquad (7)$$

$$x_{ijk} \in \{0, 1\} \qquad j, k \in N; i \in M \qquad (8)$$

$$\min \quad C_{max}$$

$$\text{s.t.} \quad \sum_{j=0, j\neq k}^{N} \sum_{i=1}^{M} x_{ijk} = 1$$

x_{ijk} = 1 if k is processed directly after j on machine i

$$\sum_{j=0, j\neq h}^{N} x_{ijh} = \sum_{k=0, k\neq h}^{N} x_{ihk}$$

each job preceded and succeeded by at most one other job

$$C_k \geq C_j + \sum_{i=1}^{M} x_{ijk}(s_{ijk} + p_{ik}) + V(\sum^{M} x_{ijk} - 1)$$

sets completion time of jobs based on sequence

$$\sum_{j=0}^{N} x_{i0j} = 1 \qquad i \in M \qquad (4)$$

$$C_j \leq C_{max} \qquad j \in N \qquad (5)$$

$$C_0 = 0 \qquad (6)$$

$$C_j \geq 0 \qquad j \in N \qquad (7)$$

$$x_{ijk} \in \{0, 1\} \qquad j, k \in N; i \in M \qquad (8)$$

nto
ing

$$\min \quad C_{max}$$

$$\text{s.t.} \quad \sum_{j=0, j \neq k}^{N} \sum_{i=1}^{M} x_{ijk} = 1$$

x_{ijk} = 1 if k is processed directly after j on machine i

$$\sum_{j=0, j \neq h}^{N} x_{ijh} = \sum_{k=0, k \neq h}^{N} x_{ihk}$$

each job preceded and succeeded by at most one other job

$$C_k \geq C_j + \sum_{i=1}^{M} x_{ijk}(s_{ijk} + p_{ik}) + V(\sum^{M} x_{ijk} - 1)$$

sets completion time of jobs based on sequence

$$\sum_{j=0}^{N} x_{i0j} = 1$$

only one job first on each machine

$$C_j \leq C_{max} \qquad j \in N \qquad (5)$$

$$C_0 = 0 \qquad (6)$$

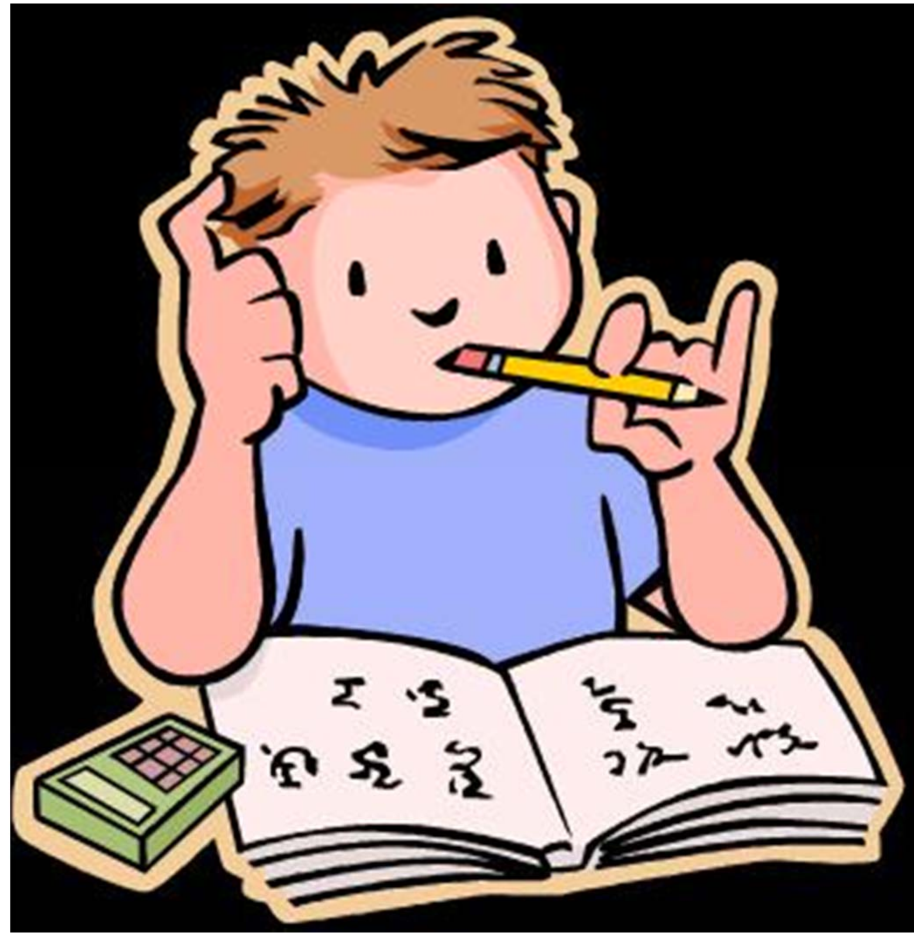
$$C_j \geq 0 \qquad j \in N \qquad (7)$$

$$x_{ijk} \in \{0, 1\} \qquad j, k \in N; i \in M \qquad (8)$$

nto
ing

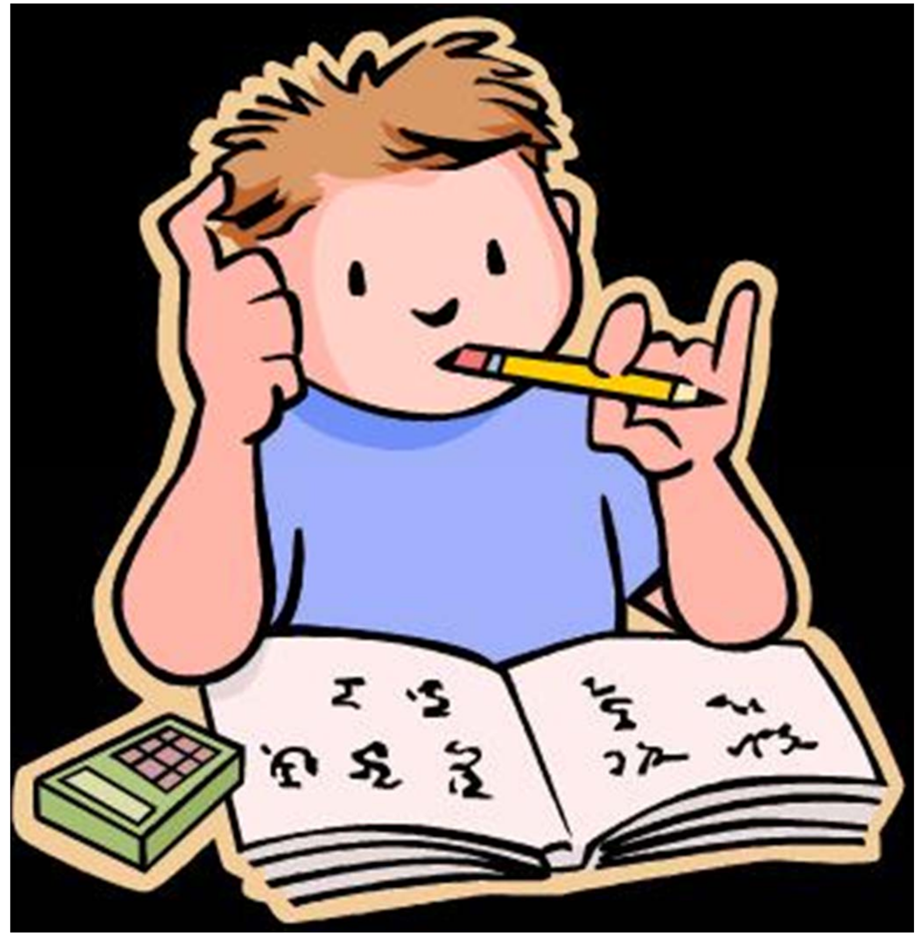
Your Turn

- Develop an LBBD model
  - master problem?
  - sub-problem?
  - sub-problem relaxation?
  - cut?



Your Turn

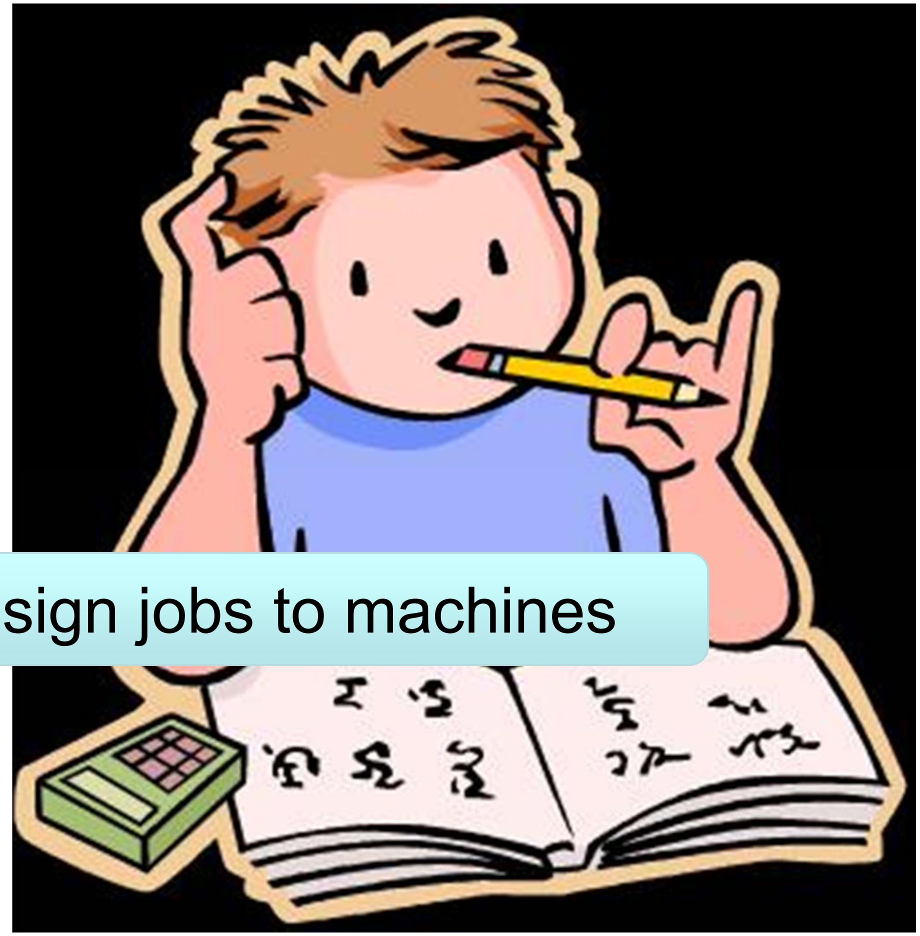
- Develop an LBBD model
  - master problem?
  - sub-problem?
  - sub-problem relaxation?
  - cut?



Remember: jobs needs to be assigned to machines and the jobs on a machine need to be sequenced.

Your Turn

- Develop an LBBD model
  - master problem?
  - sub-problem?
  - sub-problem relaxation?
  - cut?

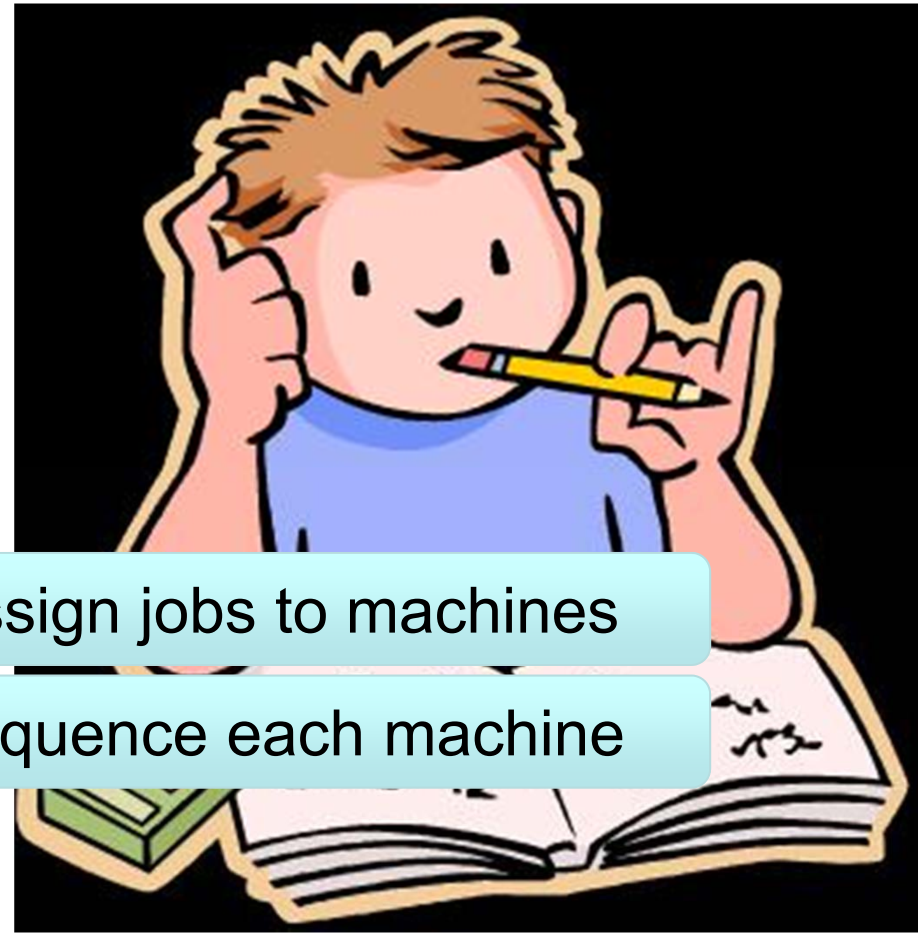


assign jobs to machines

Remember: jobs needs to be assigned to machines and the jobs on a machine need to be sequenced.

Your Turn

- Develop an LBBD model
  - master problem?
  - sub-problem?
  - sub-problem relaxation?
  - cut?



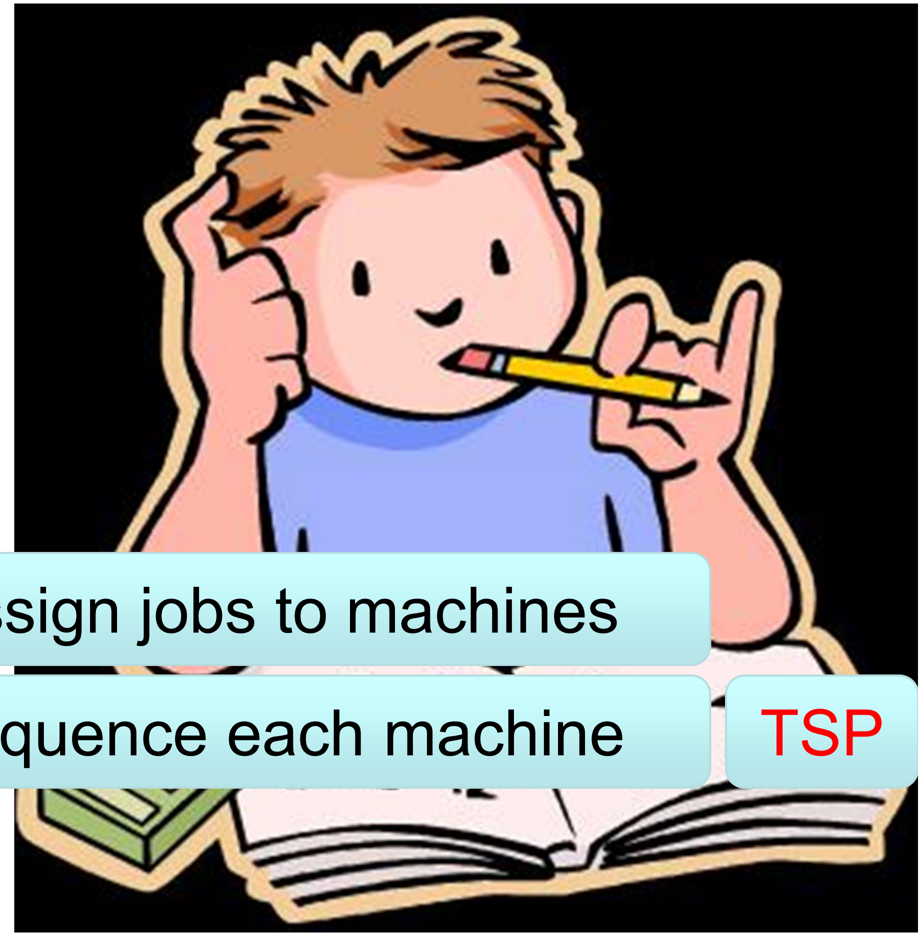
assign jobs to machines

sequence each machine

Remember: jobs needs to be assigned to machines and the jobs on a machine need to be sequenced.

Your Turn

- Develop an LBBD model
  - master problem?
  - sub-problem?
  - sub-problem relaxation?
  - cut?



assign jobs to machines

sequence each machine

TSP

Remember: jobs needs to be assigned to machines and the jobs on a machine need to be sequenced.

Master Problem

$$
\begin{aligned}
\min \quad & C_{max} \\
\text{s.t.} \quad & \sum_{j \in N} x_{ij} p_{ij} + \xi_i \leq C_{max}, && i \in M \\
& \sum_{i \in M} x_{ij} = 1, && j \in N \\
& \xi_i = \sum_{j \in N} \sum_{k \in N, k \neq j} y_{ijk} s_{ijk}, && i \in M \\
& x_{ik} = \sum_{j \in N} y_{ijk}, && k \in N; i \in M \\
& x_{ij} = \sum_{k \in N} y_{ijk}, && j \in N; i \in M \\
& cuts \\
& x_{ij} \in \{0; 1\}, && j \in N; i \in M \\
& 0 \leq y_{ijk} \leq 1, && j, k \in N; i \in M
\end{aligned}
$$

x_{ij} = 1 iff job j is assigned to machine i

y_{ijk} = 1 iff job j is immediately before job k on machine i

Master Problem

$$
\begin{aligned}
\min \quad & C_{max} \\
\text{s.t.} \quad & \sum_{j \in N} x_{ij} p_{ij} + \xi_i \leq C_{max}, && i \in M \\
& \sum_{i \in M} x_{ij} = 1, && j \in N \\
& \xi_i = \sum_{j \in N} \sum_{k \in N, k \neq j} y_{ijk} s_{ijk}, && i \in M \\
& x_{ik} = \sum_{j \in N} y_{ijk}, && k \in N; i \in M \\
& x_{ij} = \sum_{k \in N} y_{ijk}, && j \in N; i \in M \\
& cuts \\
& x_{ij} \in \{0; 1\}, && j \in N; i \in M \\
& 0 \leq y_{ijk} \leq 1, && j, k \in N; i \in M
\end{aligned}
$$

x_{ij} = 1 iff job j is assigned to machine i
y_{ijk} = 1 iff job j is immediately before job k on machine i

Sub-problem relaxation

Master Problem

x_{ij} = 1 iff job j is assigned to machine i
y_{ijk} = 1 iff job j is immediately before job k on machine i

$$\min \quad C_{max}$$

$$\text{s.t.} \quad \sum_{j \in N} x_{ij} p_{ij} + \xi_i \le C_{max}, \qquad i \in M$$

$$\sum_{i \in M} x_{ij} = 1, \qquad j \in N$$

$$\xi_i = \sum_{j \in N} \sum_{k \in N, k \neq j} y_{ijk} s_{ijk}, \qquad i \in M$$

$$x_{ik} = \sum_{j \in N} y_{ijk}, \qquad k \in N; i \in M$$

$$x_{ij} = \sum_{k \in N} y_{ijk}, \qquad j \in N; i \in M$$

Sub-problem relaxation

$$cuts$$

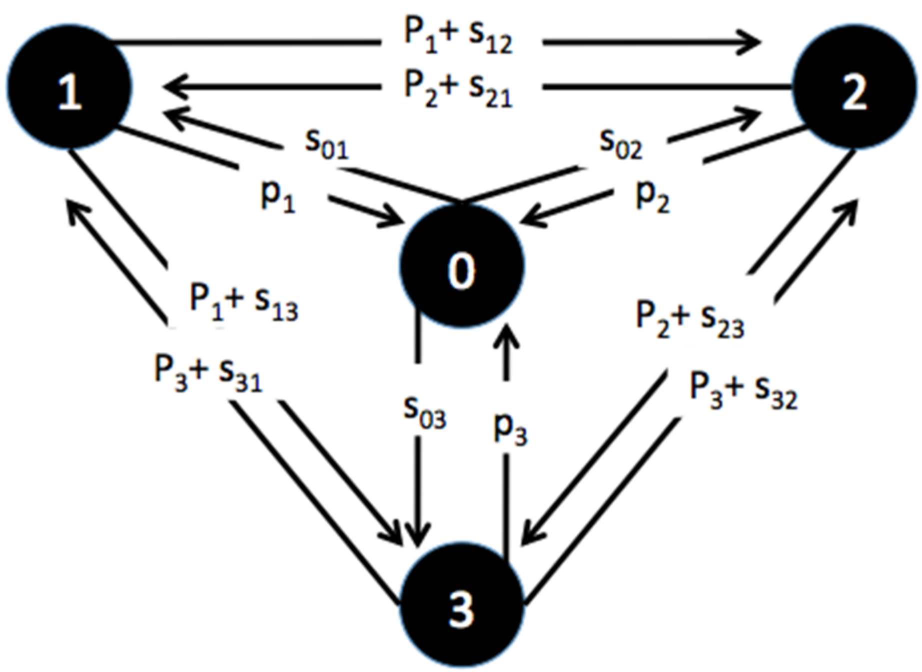
Generated by sub-problem

$$x_{ij} \in \{0; 1\}, \qquad j \in N; i \in M$$

$$0 \le y_{ijk} \le 1, \qquad j, k \in N; i \in M$$

Sub-problem

- Assymetric TSP
  - nodes = jobs
  - distance = set-up time



iversity of Toronto
ıstrial Engineering

Cut

$$C_{max} \geq C_{max}^{hi*} - \sum_{j \in N_i^h} (1 - x_{ij})\theta_{hij}.$$

Cut

$$C_{max} \geq C_{max}^{hi*} - \sum_{j \in N_i^h} (1 - x_{ij})\theta_{hij}.$$

Optimal makespan on machine i in iteration h

Cut

$$C_{max} \geq C_{max}^{hi*} - \sum_{j \in N_i^h} (1 - x_{ij})\theta_{hij}.$$

Optimal makespan on machine i in iteration h

$$\theta_{hij} = p_{ij} + maxPre_{hj},$$

Cut

$$C_{max} \geq C_{max}^{hi*} - \sum_{j \in N_i^h} (1 - x_{ij})\theta_{hij}.$$

Optimal makespan on machine i in iteration h

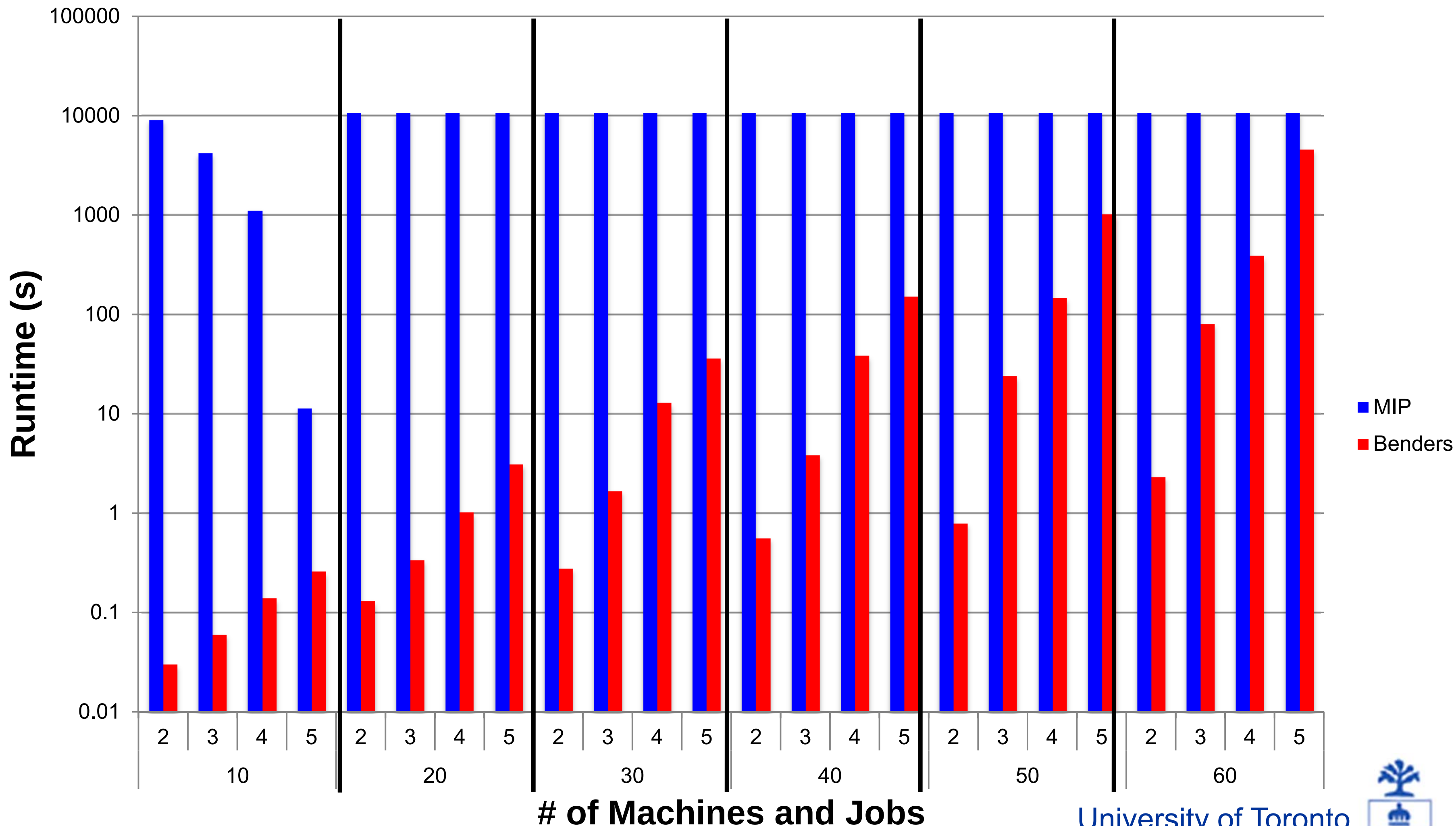
$$\theta_{hij} = p_{ij} + maxPre_{hj},$$

Lower bound on job j's contribution to the makespan on machine i in iteration h

Stopping Conditions

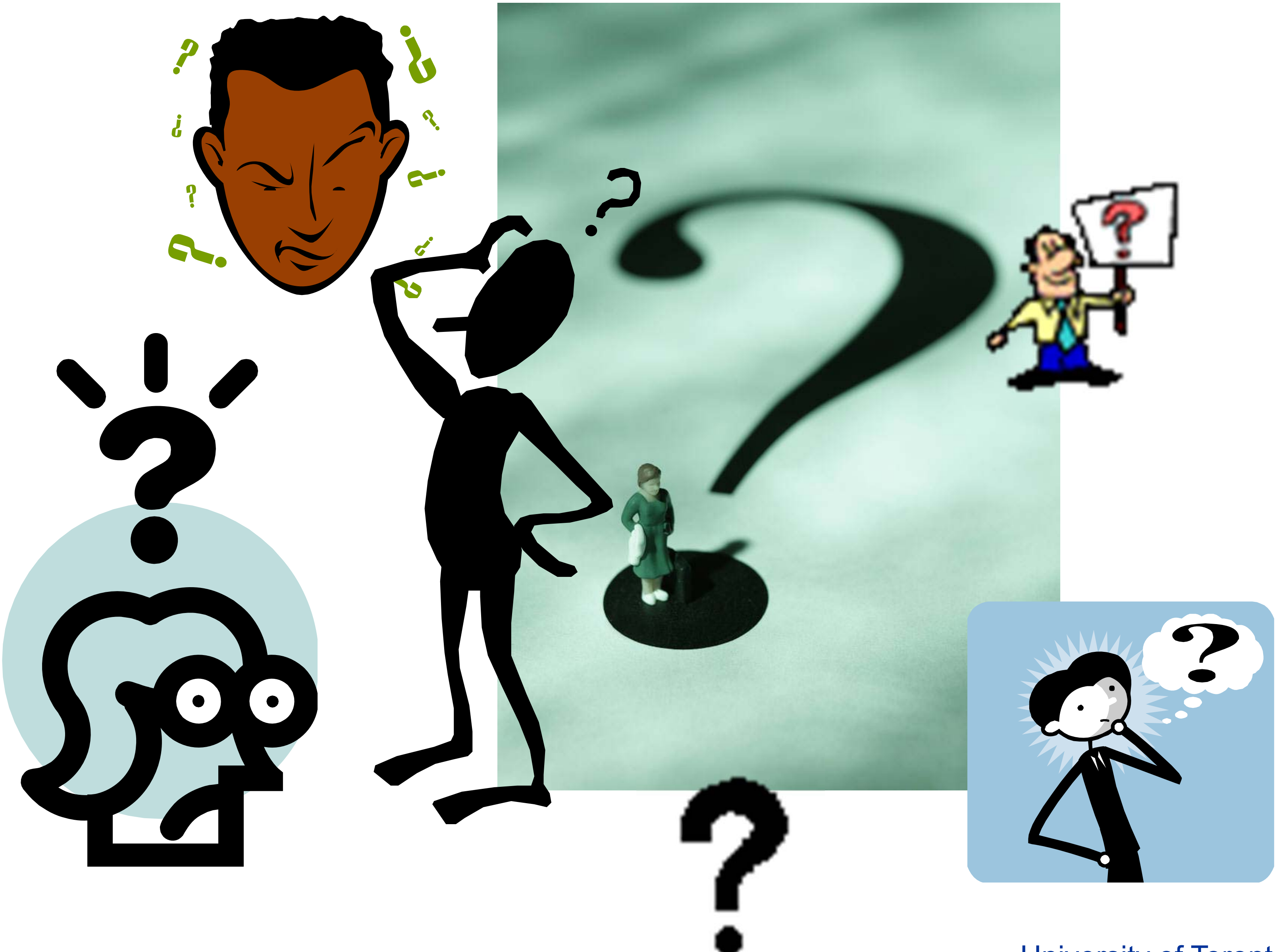
- All SPs find schedule with makespan ≤ makespan of MP, or
- MP finds solution with makespan equal to best feasible solution found so far
  – each iteration provides a feasible (but not necessarily improving) solution

Results



University of Toronto
Mechanical & Industrial Engineering

[Tran & B. 2012] *ECAI*, 774-779, 2012.



The Plan

- Decomposition & Modeling
- Logic-based Benders Decomposition (LBBD)
- Applying LBBD to Problems Somewhat Related to Computational Sustainability
- Beyond Decomposition

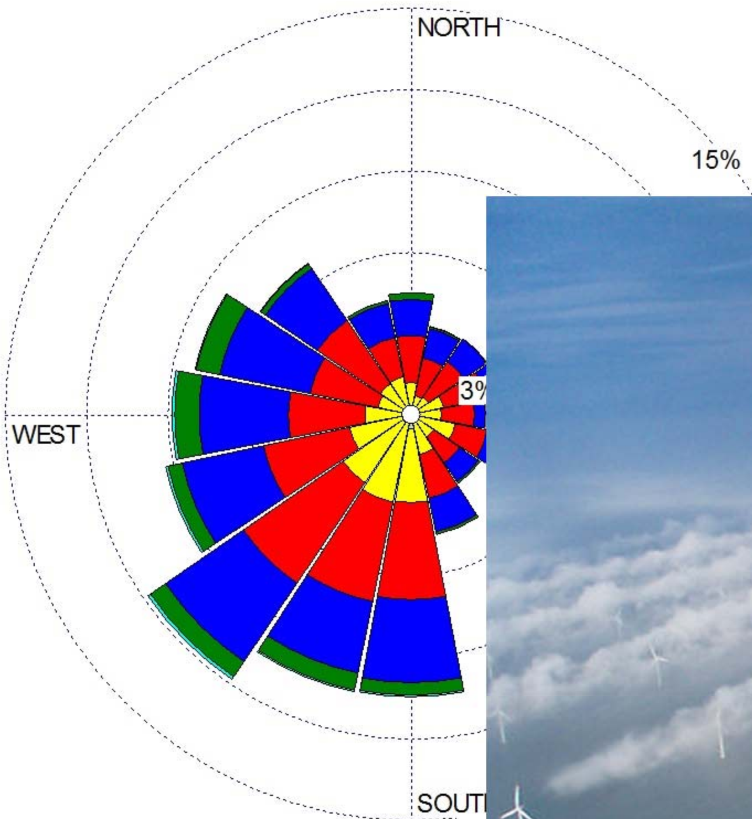
wherein we try to get back to the topic of the Master Class

Problem 1: Turbine Placement

- Objective: maximize energy production or profit
- Constraints:
  - location: min. separation, land topology, existing infrastructure
  - limit of input power to grid
  - turbine specifications
- Decisions:
  - turbine types, number, placement

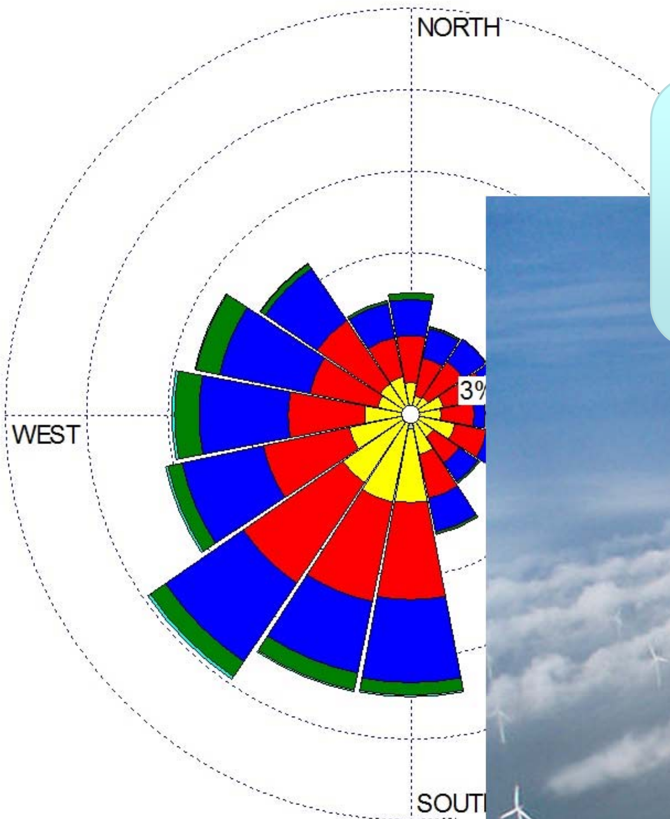
Thanks to Peter Zhang.

Turbine Placement Challenges



Thanks to Peter Zhang.

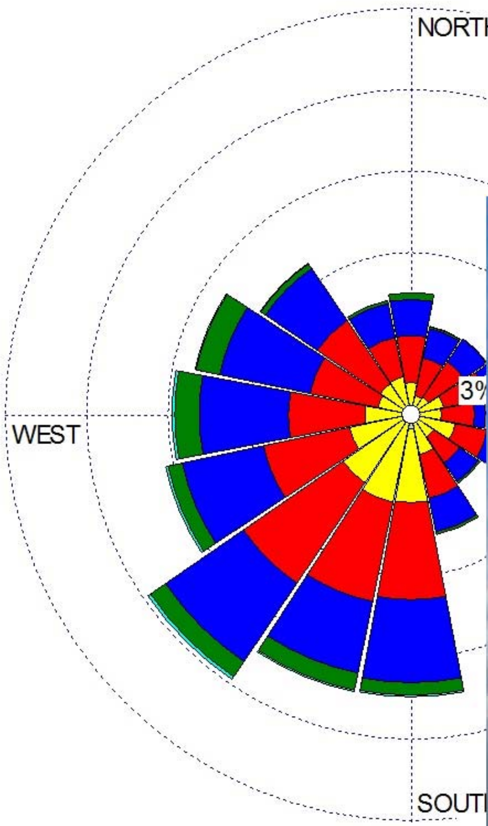
Turbine Placement Challenges



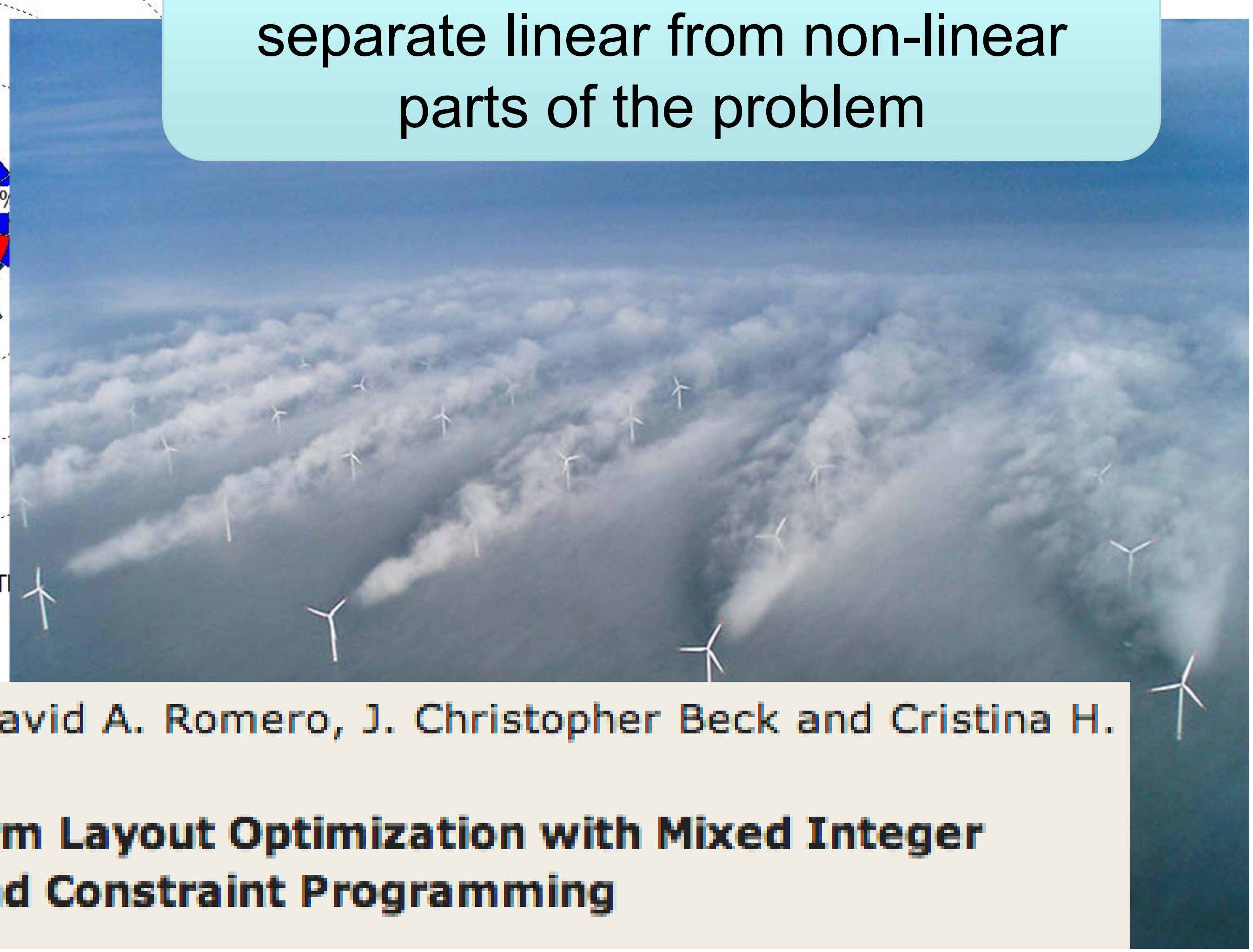
Idea: use decomposition to separate linear from non-linear parts of the problem

Thanks to Peter Zhang.

Turbine Placement Challenges



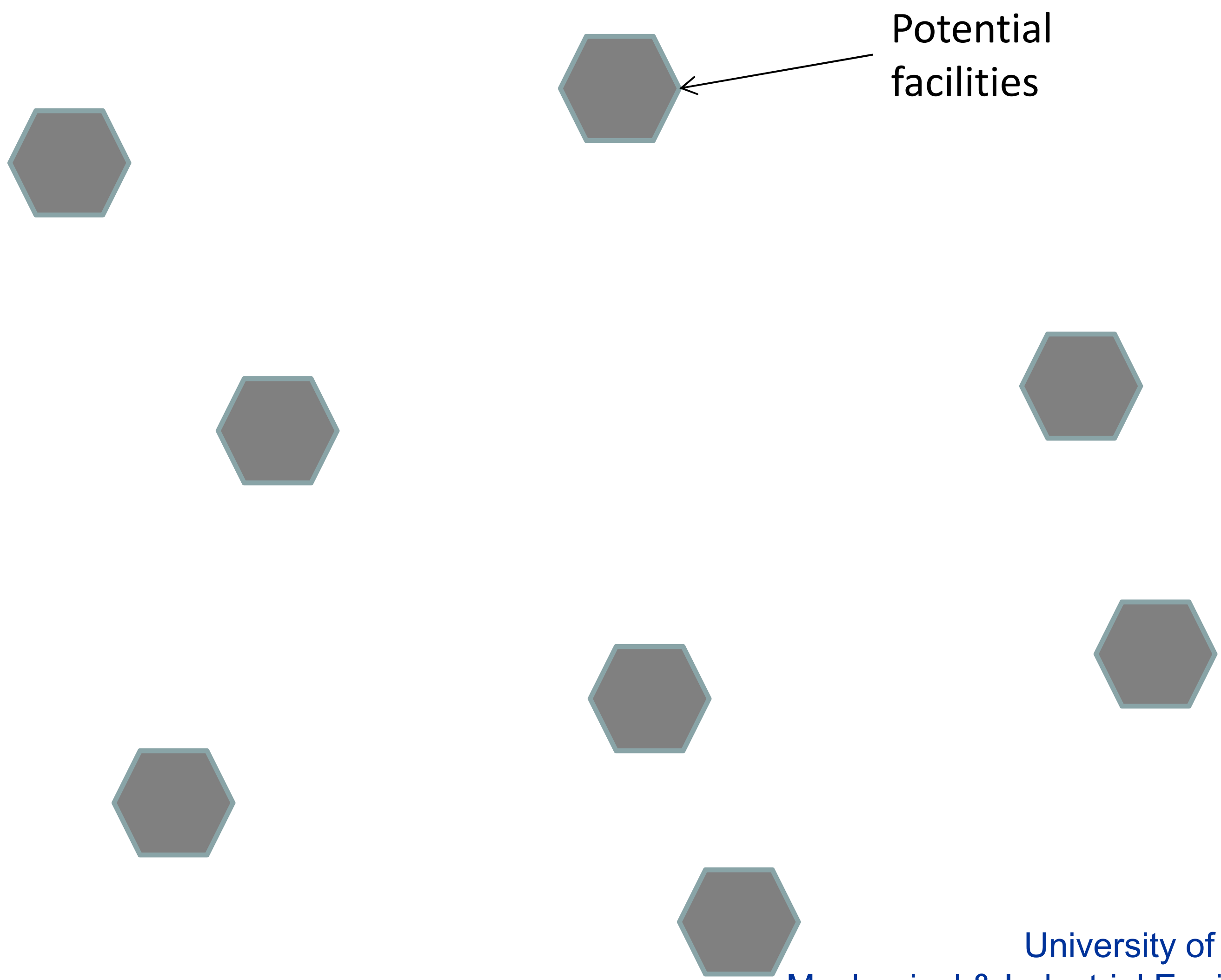
Idea: use decomposition to separate linear from non-linear parts of the problem



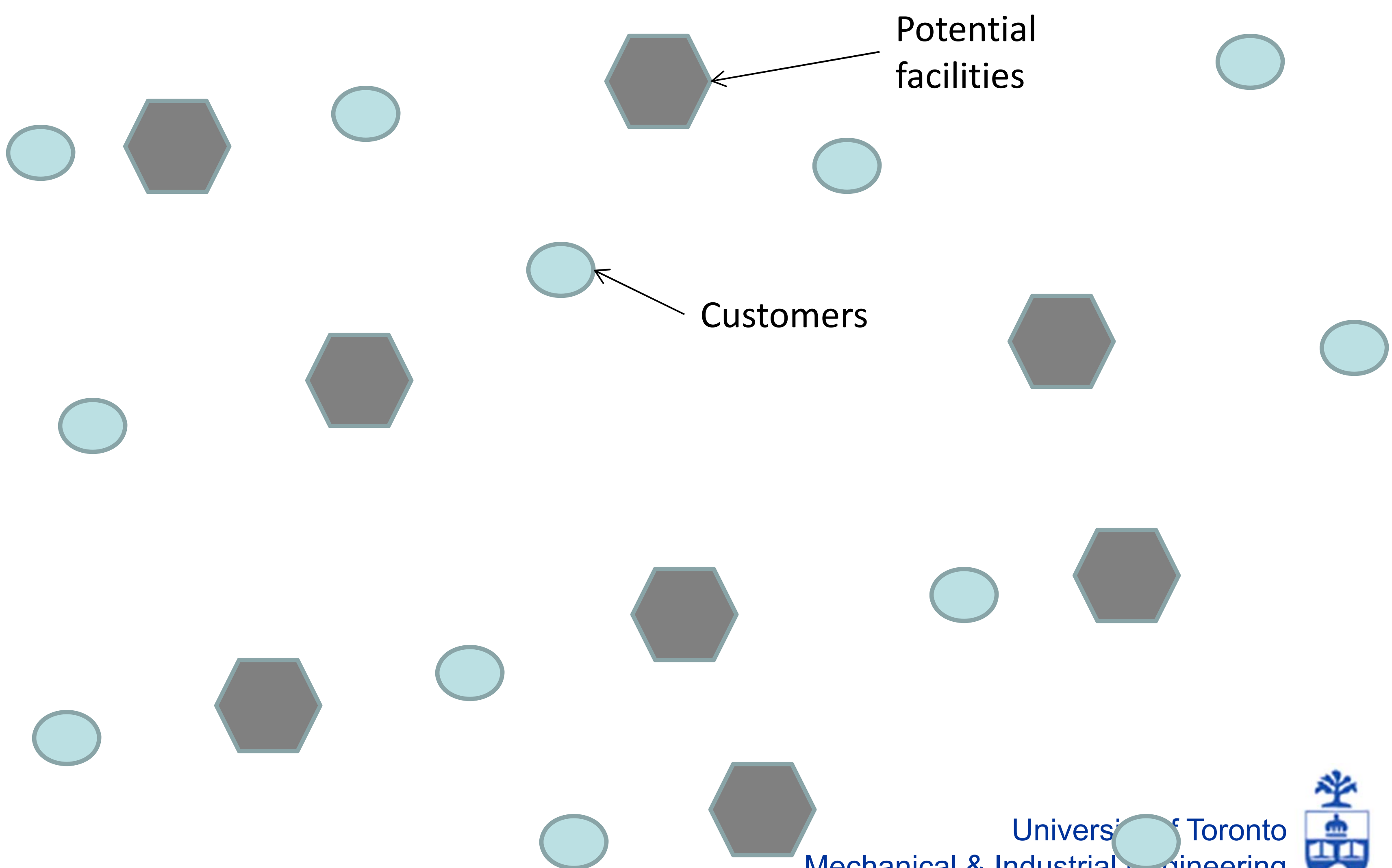
Peter Y. Zhang, David A. Romero, J. Christopher Beck and Cristina H. Amon
Solving Wind Farm Layout Optimization with Mixed Integer Programming and Constraint Programming

Than

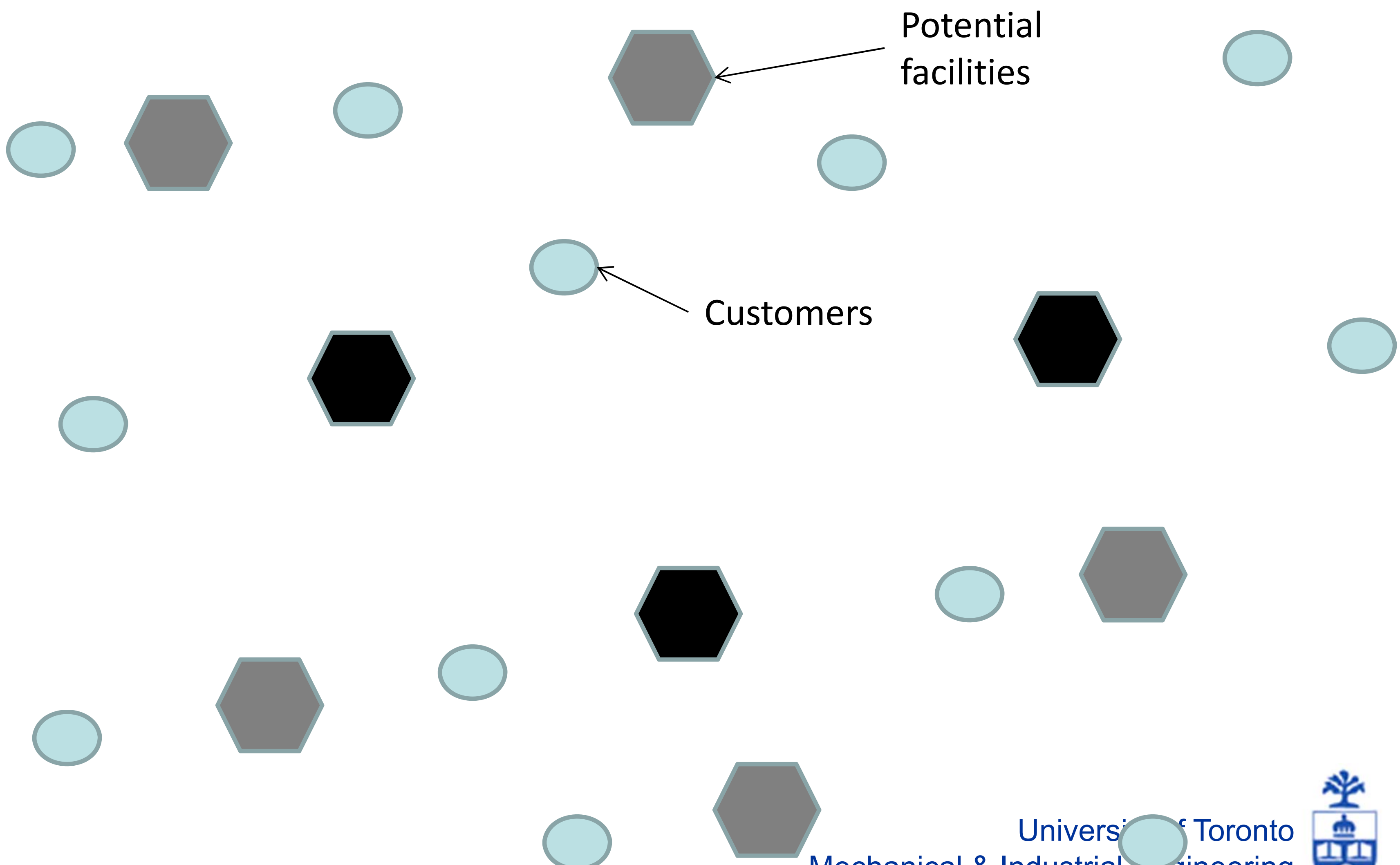
A Location-Allocation Problem



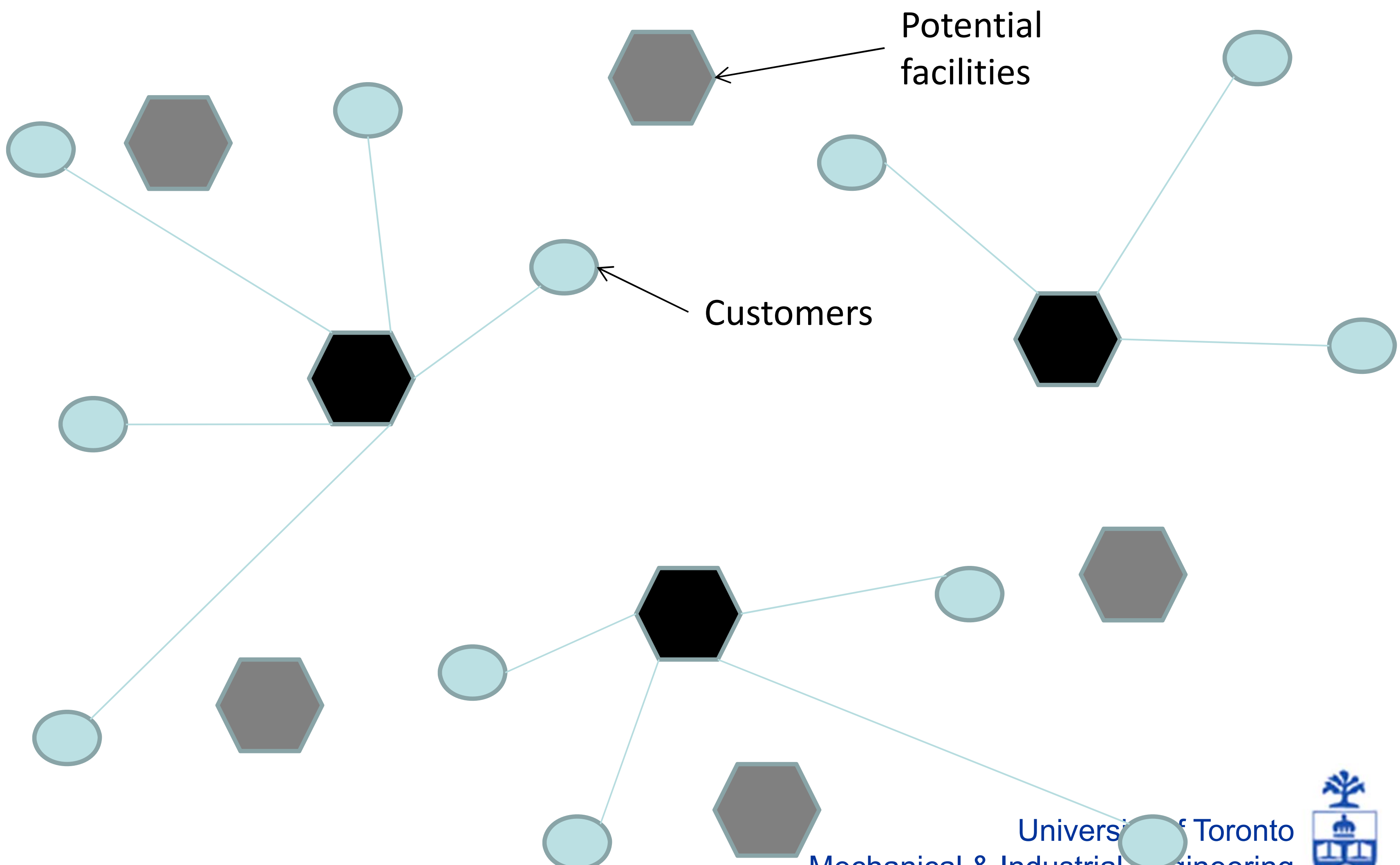
A Location-Allocation Problem



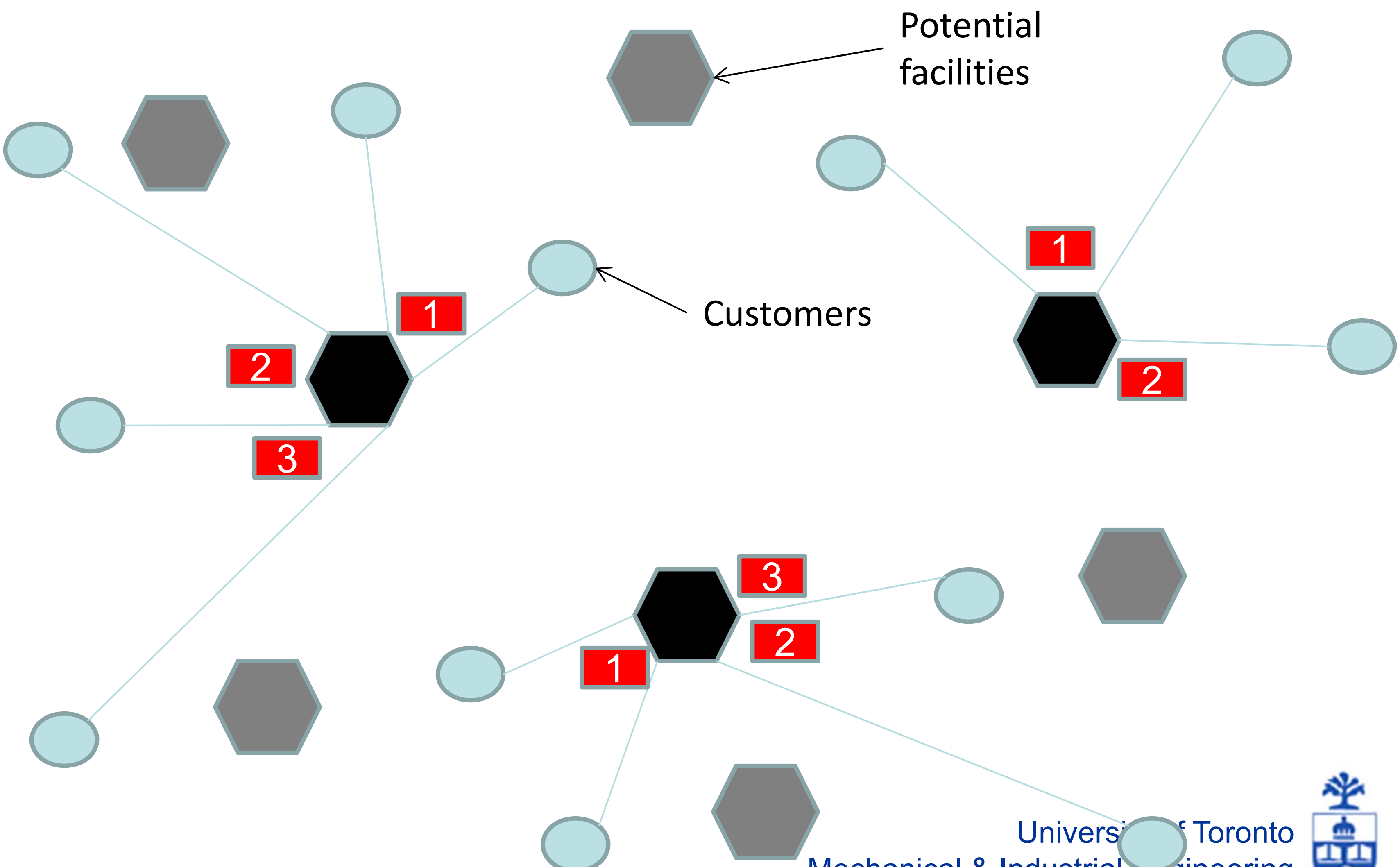
A Location-Allocation Problem



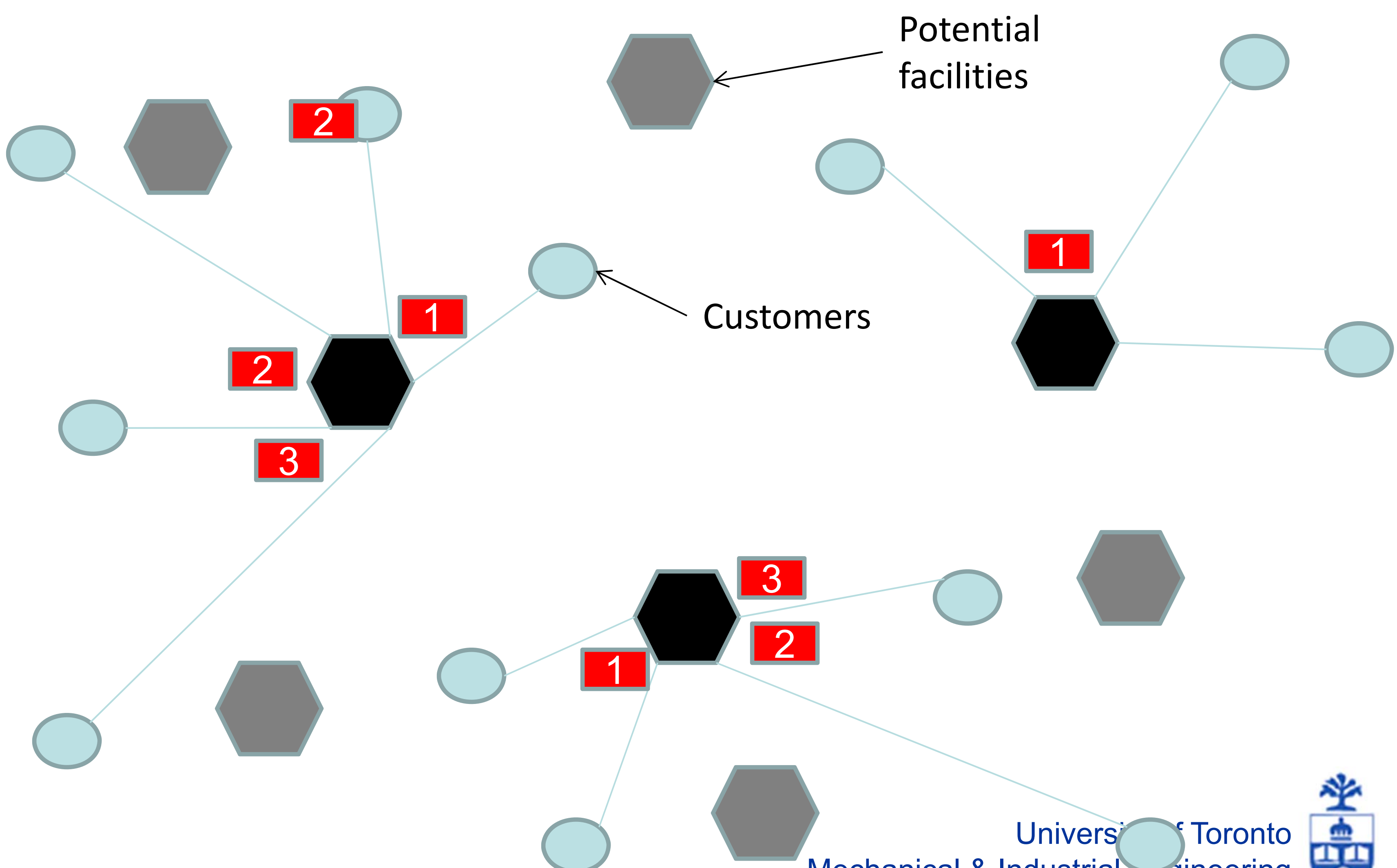
A Location-Allocation Problem



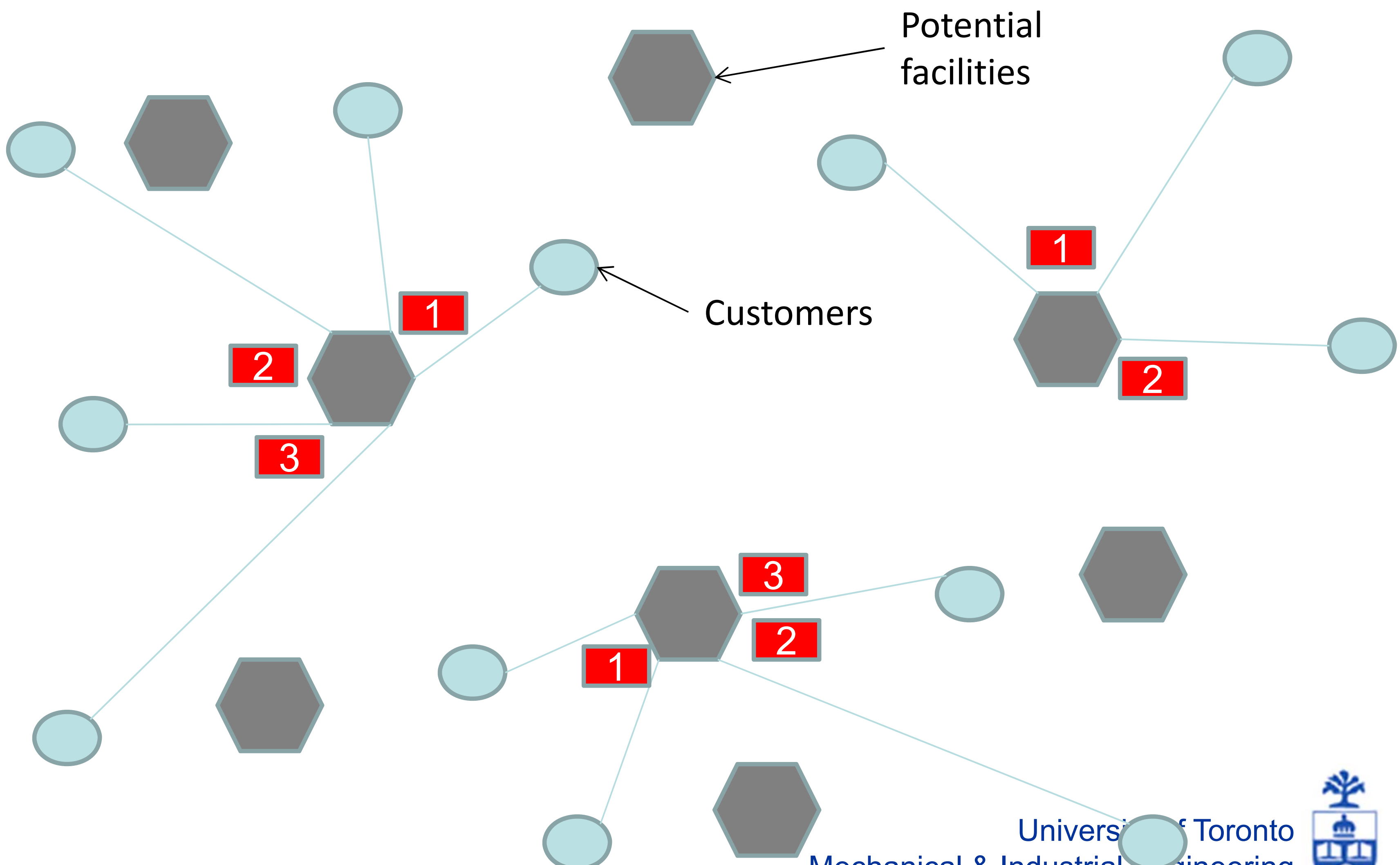
A Location-Allocation Problem



A Location-Allocation Problem



A Location-Allocation Problem



Computational Sustainability?

- Originally the facilities were to be recycling centres in the city of Tehran

A Mixed Integer Model

$$p_j = \begin{cases} 1: & \text{if site j is open} \\ 0: & \text{otherwise} \end{cases}$$

$$x_{ijk} = \begin{cases} 1: & \text{if client i is served by the kth vehicle of site j} \\ 0: & \text{otherwise} \end{cases}$$

$$z_{jk} = \begin{cases} 1: & \text{if a kth vehicle of site j} \\ 0: & \text{otherwise} \end{cases}$$

[Alberada-Sambola et al. 2009], *Computers & OR*, 36(2): 597-611, 2009.

$$\min \sum_{j \in J} f_j p_j + u \sum_{j \in J} \sum_{k \in K} z_{jk} + \sum_{i \in I} \sum_{j \in J} c_{ij} \sum_{k \in k} x_{ijk}$$

$$\begin{aligned}
\text{s.t.} \quad & \sum_{j \in J} \sum_{k \in K} x_{ijk} = 1 && i \in I && (1)\\
& \sum_{i \in I} t_{ij} x_{ijk} \leq l \cdot z_{jk} && j \in J, k \in K && (2)\\
& \sum_{i \in I} \sum_{k \in K} d_i x_{ijk} \leq b_j p_j && j \in J && (3)\\
& z_{jk} \leq p_j && j \in J, k \in K && (4)\\
& x_{ijk} \leq z_{jk} && i \in I, j \in J, k \in K && (5)\\
& z_{jk} \leq z_{jk-1} && j \in J, k \in K \backslash \{1\} && (6)\\
& x_{ijk}, p_j, z_{jk} \in \{0, 1\} && i \in I, j \in J, k \in K && (7)
\end{aligned}$$

[Alberada-Sambola et al. 2009], *Computers & OR*, 36(2): 597-611, 2009.

Fixed facility cost

$$\min \sum_{j\in J} f_j p_j + u \sum_{j\in J}\sum_{k\in K} z_{jk} + \sum_{i\in I}\sum_{j\in J} c_{ij} \sum_{k\in k} x_{ijk}$$

$$
\begin{aligned}
\text{s.t.} \quad & \sum_{j\in J}\sum_{k\in K} x_{ijk} = 1 && i \in I && (1)\\
& \sum_{i\in I} t_{ij} x_{ijk} \le l \cdot z_{jk} && j \in J, k \in K && (2)\\
& \sum_{i\in I}\sum_{k\in K} d_i x_{ijk} \le b_j p_j && j \in J && (3)\\
& z_{jk} \le p_j && j \in J, k \in K && (4)\\
& x_{ijk} \le z_{jk} && i \in I, j \in J, k \in K && (5)\\
& z_{jk} \le z_{jk-1} && j \in J, k \in K \backslash \{1\} && (6)\\
& x_{ijk}, p_j, z_{jk} \in \{0,1\} && i \in I, j \in J, k \in K && (7)
\end{aligned}
$$

[Alberada-Sambola et al. 2009], *Computers & OR*, 36(2): 597-611, 2009.

$$\min \sum_{j\in J} f_j p_j + u \sum_{j\in J}\sum_{k\in K} z_{jk} + \sum_{i\in I}\sum_{j\in J} c_{ij} \sum_{k\in k} x_{ijk}$$

Fixed facility cost

Vehicle cost

$$\text{s.t.} \quad \sum_{j\in J}\sum_{k\in K} x_{ijk} = 1 \qquad i \in I \qquad (1)$$

$$\sum_{i\in I} t_{ij} x_{ijk} \le l \cdot z_{jk} \qquad j \in J, k \in K \qquad (2)$$

$$\sum_{i\in I}\sum_{k\in K} d_i x_{ijk} \le b_j p_j \qquad j \in J \qquad (3)$$

$$z_{jk} \le p_j \qquad j \in J, k \in K \qquad (4)$$

$$x_{ijk} \le z_{jk} \qquad i \in I, j \in J, k \in K \qquad (5)$$

$$z_{jk} \le z_{jk-1} \qquad j \in J, k \in K\backslash\{1\} \qquad (6)$$

$$x_{ijk}, p_j, z_{jk} \in \{0,1\} \qquad i \in I, j \in J, k \in K \qquad (7)$$

[Alberada-Sambola et al. 2009], *Computers & OR*, 36(2): 597-611, 2009.

$$\min \sum_{j \in J} f_j p_j + u \sum_{j \in J} \sum_{k \in K} z_{jk} + \sum_{i \in I} \sum_{j \in J} c_{ij} \sum_{k \in k} x_{ijk}$$

Fixed facility cost

Assignment cost

Vehicle cost

$$\text{s.t.} \quad \sum_{j \in J} \sum_{k \in K} x_{ijk} = 1 \qquad i \in I \qquad (1)$$

$$\sum_{i \in I} t_{ij} x_{ijk} \leq l \cdot z_{jk} \qquad j \in J, k \in K \qquad (2)$$

$$\sum_{i \in I} \sum_{k \in K} d_i x_{ijk} \leq b_j p_j \qquad j \in J \qquad (3)$$

$$z_{jk} \leq p_j \qquad j \in J, k \in K \qquad (4)$$

$$x_{ijk} \leq z_{jk} \qquad i \in I, j \in J, k \in K \qquad (5)$$

$$z_{jk} \leq z_{jk-1} \qquad j \in J, k \in K \backslash \{1\} \qquad (6)$$

$$x_{ijk}, p_j, z_{jk} \in \{0, 1\} \qquad i \in I, j \in J, k \in K \qquad (7)$$

[Alberada-Sambola et al. 2009], *Computers & OR*, 36(2): 597-611, 2009.

$$\min \sum_{j \in J} f_j p_j + u \sum_{j \in J} \sum_{k \in K} z_{jk} + \sum_{i \in I} \sum_{j \in J} c_{ij} \sum_{k \in k} x_{ijk}$$

Fixed facility cost

Assignment cost

Vehicle cost

$$\text{s.t.} \quad \sum_{j \in J} \sum_{k \in K} x_{ijk} = 1 \qquad i \in I$$

Each client is served by one truck at one site

$$\sum_{i \in I} t_{ij} x_{ijk} \leq l \cdot z_{jk} \qquad j \in J, k \in K \qquad (2)$$

$$\sum_{i \in I} \sum_{k \in K} d_i x_{ijk} \leq b_j p_j \qquad j \in J \qquad (3)$$

$$z_{jk} \leq p_j \qquad j \in J, k \in K \qquad (4)$$

$$x_{ijk} \leq z_{jk} \qquad i \in I, j \in J, k \in K \qquad (5)$$

$$z_{jk} \leq z_{jk-1} \qquad j \in J, k \in K \setminus \{1\} \qquad (6)$$

$$x_{ijk}, p_j, z_{jk} \in \{0, 1\} \qquad i \in I, j \in J, k \in K \qquad (7)$$

[Alberada-Sambola et al. 2009], *Computers & OR*, 36(2): 597-611, 2009.

$$\min \sum_{j \in J} f_j p_j + u \sum_{j \in J} \sum_{k \in K} z_{jk} + \sum_{i \in I} \sum_{j \in J} c_{ij} \sum_{k \in k} x_{ijk}$$

Fixed facility cost

Assignment cost

Vehicle cost

$$\text{s.t.} \quad \sum_{j \in J} \sum_{k \in K} x_{ijk} = 1 \qquad i \in I$$

$$\sum_{i \in I} t_{ij} x_{ijk} \le l \cdot z_{jk} \qquad j \in J, k \in K$$

$$\sum_{i \in I} \sum_{k \in K} d_i x_{ijk} \le b_j p_j \qquad j \in J \qquad (3)$$

$$z_{jk} \le p_j \qquad j \in J, k \in K \qquad (4)$$

$$x_{ijk} \le z_{jk} \qquad i \in I, j \in J, k \in K \qquad (5)$$

$$z_{jk} \le z_{jk-1} \qquad j \in J, k \in K \backslash \{1\} \qquad (6)$$

$$x_{ijk}, p_j, z_{jk} \in \{0, 1\} \qquad i \in I, j \in J, k \in K \qquad (7)$$

Each client is served by one truck at one site

Distance Constraint

[Alberada-Sambola et al. 2009], *Computers & OR*, 36(2): 597-611, 2009.

$$\min \sum_{j \in J} f_j p_j + u \sum_{j \in J} \sum_{k \in K} z_{jk} + \sum_{i \in I} \sum_{j \in J} c_{ij} \sum_{k \in k} x_{ijk}$$

Fixed facility cost

Assignment cost

Vehicle cost

$$\text{s.t.} \quad \sum_{j \in J} \sum_{k \in K} x_{ijk} = 1 \qquad i \in I$$

$$\sum_{i \in I} t_{ij} x_{ijk} \leq l \cdot z_{jk} \qquad j \in J, k \in K$$

$$\sum_{i \in I} \sum_{k \in K} d_i x_{ijk} \leq b_j p_j \qquad j \in J$$

$$z_{jk} \leq p_j \qquad j \in J, k \in K \qquad (4)$$

$$x_{ijk} \leq z_{jk} \qquad i \in I, j \in J, k \in K \qquad (5)$$

$$z_{jk} \leq z_{jk-1} \qquad j \in J, k \in K \backslash \{1\} \qquad (6)$$

$$x_{ijk}, p_j, z_{jk} \in \{0, 1\} \qquad i \in I, j \in J, k \in K \qquad (7)$$

Each client is served by one truck at one site

Distance Constraint

Capacity Constraint

[Alberada-Sambola et al. 2009], *Computers & OR*, 36(2): 597-611, 2009.

$$\min \sum_{j\in J} f_j p_j + u \sum_{j\in J}\sum_{k\in K} z_{jk} + \sum_{i\in I}\sum_{j\in J} c_{ij} \sum_{k\in k} x_{ijk}$$

Fixed facility cost

Assignment cost

Vehicle cost

$$\text{s.t.} \quad \sum_{j\in J}\sum_{k\in K} x_{ijk} = 1 \qquad i \in I$$

$$\sum_{i\in I} t_{ij} x_{ijk} \leq l \cdot z_{jk} \qquad j \in J, k \in K$$

$$\sum_{i\in I}\sum_{k\in K} d_i x_{ijk} \leq b_j p_j \qquad j \in J$$

$$z_{jk} \leq p_j \qquad j \in J, k \in K$$

$$x_{ijk} \leq z_{jk} \qquad i \in I, j \in J, k \in K$$

$$z_{jk} \leq z_{jk-1} \qquad j \in J, k \in K\setminus\{1\} \qquad (6)$$

$$x_{ijk}, p_j, z_{jk} \in \{0,1\} \qquad i \in I, j \in J, k \in K \qquad (7)$$

Each client is served by one truck at one site

Distance Constraint

Capacity Constraint

A client must be served by an open site and an allocated vehicle

[Alberada-Sambola et al. 2009], *Computers & OR*, 36(2): 597-611, 2009.

$$\min \sum_{j \in J} f_j p_j + u \sum_{j \in J} \sum_{k \in K} z_{jk} + \sum_{i \in I} \sum_{j \in J} c_{ij} \sum_{k \in k} x_{ijk}$$

- $\sum_{j \in J} f_j p_j$: Fixed facility cost
- $u \sum_{j \in J} \sum_{k \in K} z_{jk}$: Vehicle cost
- $\sum_{i \in I} \sum_{j \in J} c_{ij} \sum_{k \in k} x_{ijk}$: Assignment cost

s.t.

$$\sum_{j \in J} \sum_{k \in K} x_{ijk} = 1 \qquad i \in I$$

Each client is served by one truck at one site

$$\sum_{i \in I} t_{ij} x_{ijk} \leq l \cdot z_{jk} \qquad j \in J, k \in K$$

Distance Constraint

$$\sum_{i \in I} \sum_{k \in K} d_i x_{ijk} \leq b_j p_j \qquad j \in J$$

Capacity Constraint

$$z_{jk} \leq p_j \qquad j \in J, k \in K$$

$$x_{ijk} \leq z_{jk} \qquad i \in I, j \in J, k \in K$$

A client must be served by an open site and an allocated vehicle

$$z_{jk} \leq z_{jk-1} \qquad j \in J, k \in K \backslash \{1\}$$

Symmetry Constraint

$$x_{ijk}, p_j, z_{jk} \in \{0, 1\} \qquad i \in I, j \in J, k \in K \qquad (7)$$

[Alberada-Sambola et al. 2009], *Computers & OR*, 36(2): 597-611, 2009.

$$\min \sum_{j\in J} f_j p_j + u \sum_{j\in J} \sum_{k\in K} z_{jk} + \sum_{i\in I} \sum_{j\in J} c_{ij} \sum_{k\in k} x_{ijk}$$

Fixed facility cost

Assignment cost

s.t.

...one

...y

Problem: The model doesn't scale

40 clients, 20 possible locations: 75% of problems unsolved in 48 hours

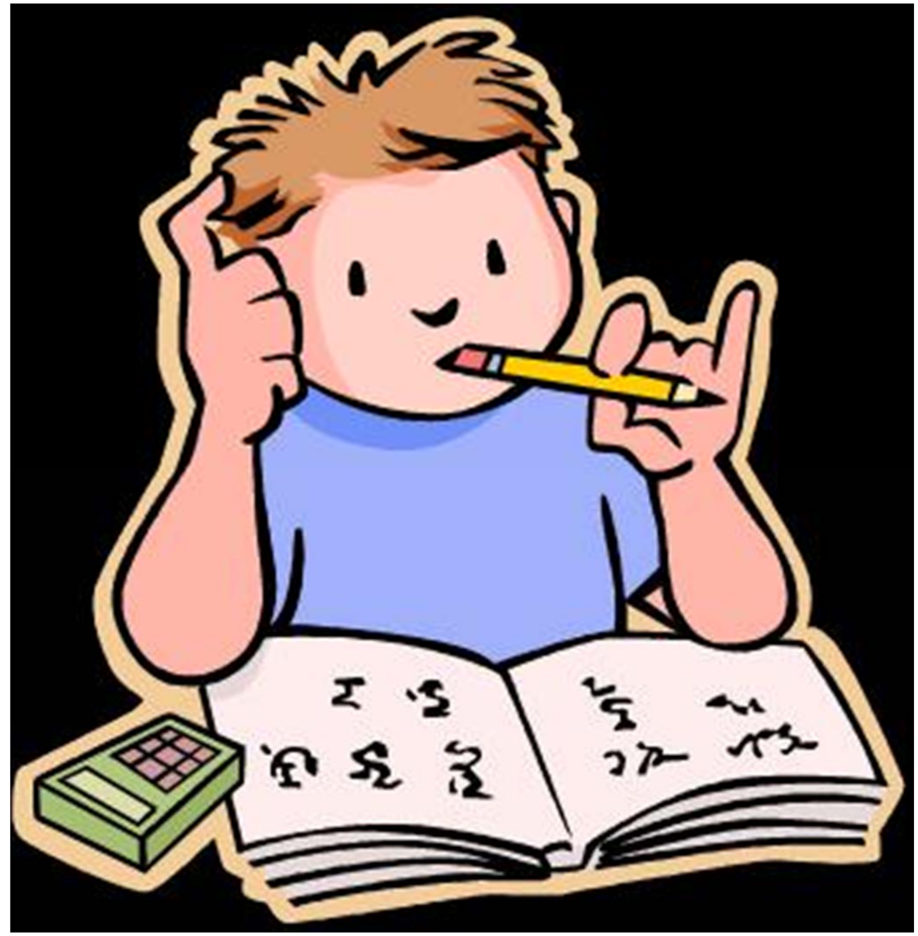
[CP is even worse]

University of Toronto
Mechanical & Industrial Engineering

[Alberada-Sambola et al. 2009], *Computers & OR*, 36(2): 597-611, 2009.

Your Turn

- Develop an LBBD model
  - master problem?
  - sub-problem?
  - sub-problem relaxation?
  - cut?



Decisions to make

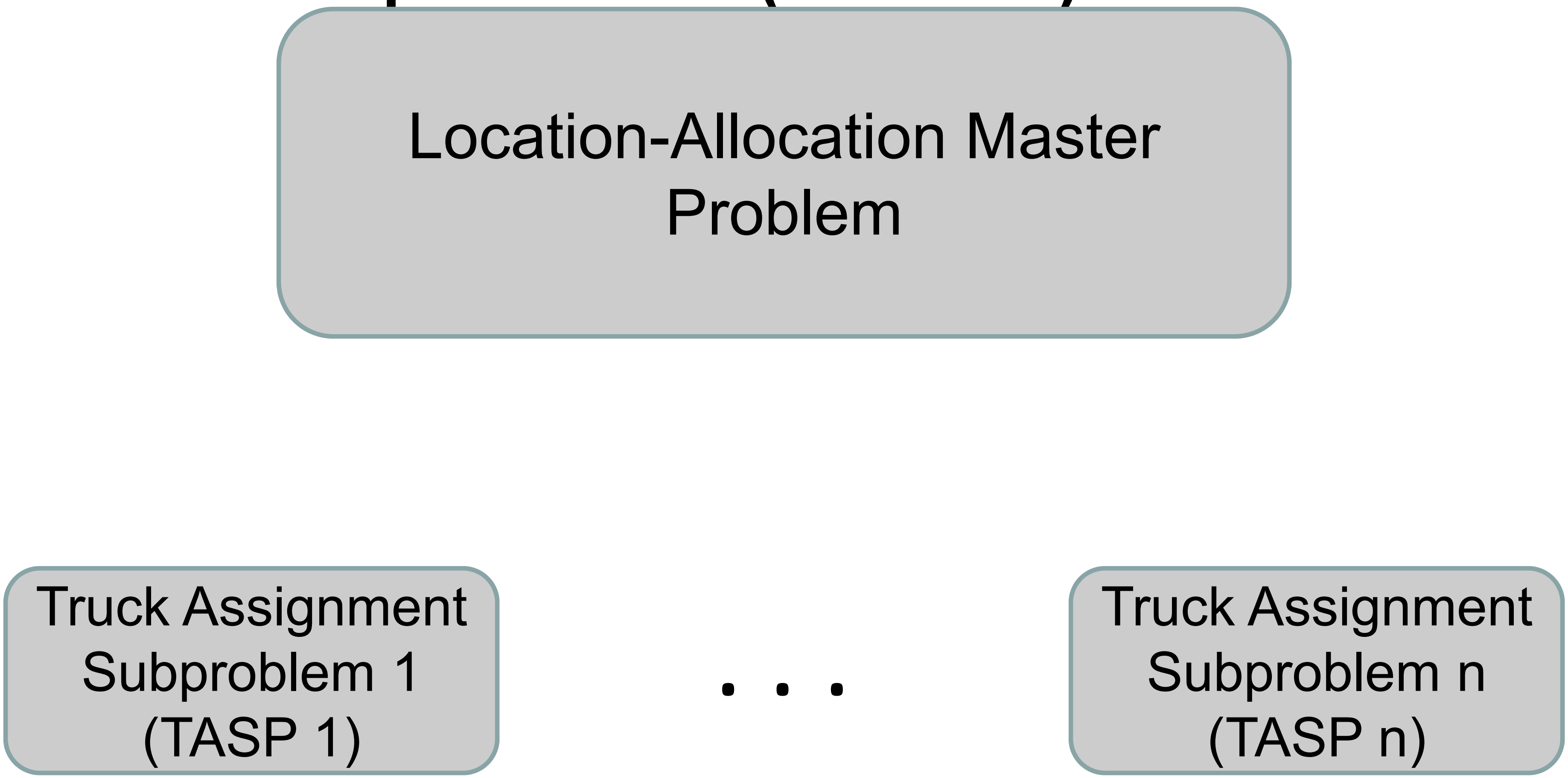
- Which facilities to open
- Which customers to assign to which open facilities
- How many vehicles at each facility
- Which customers to assign to which trucks

Logic-Based Benders Decomposition (LBBD)

Capacity and Distance Constrained Plant Location Problem

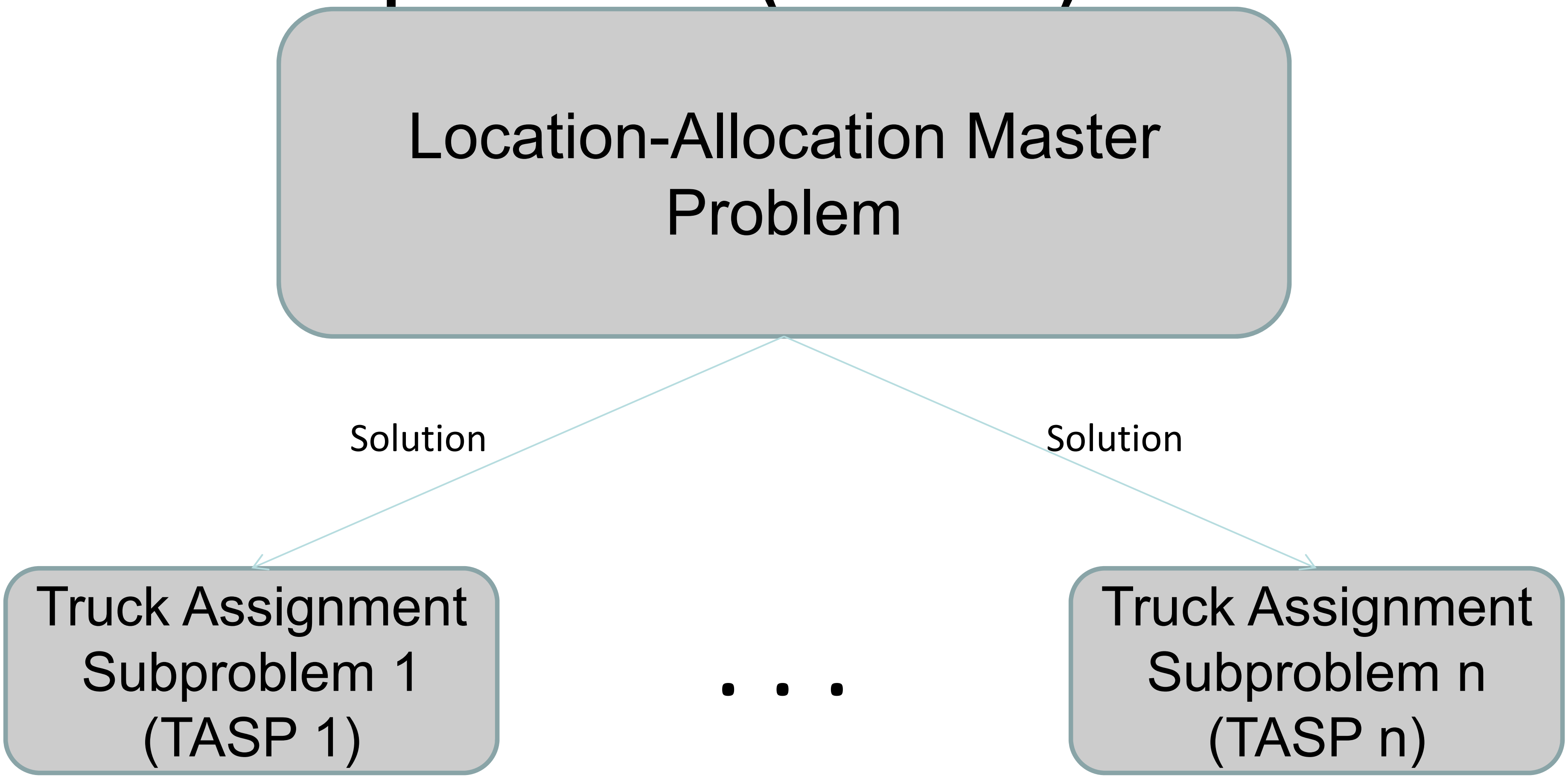
[Fazel-Zarandi & B 2012] *IJOC*, 24, 399-415, 2012.

Logic-Based Benders Decomposition (LBBD)



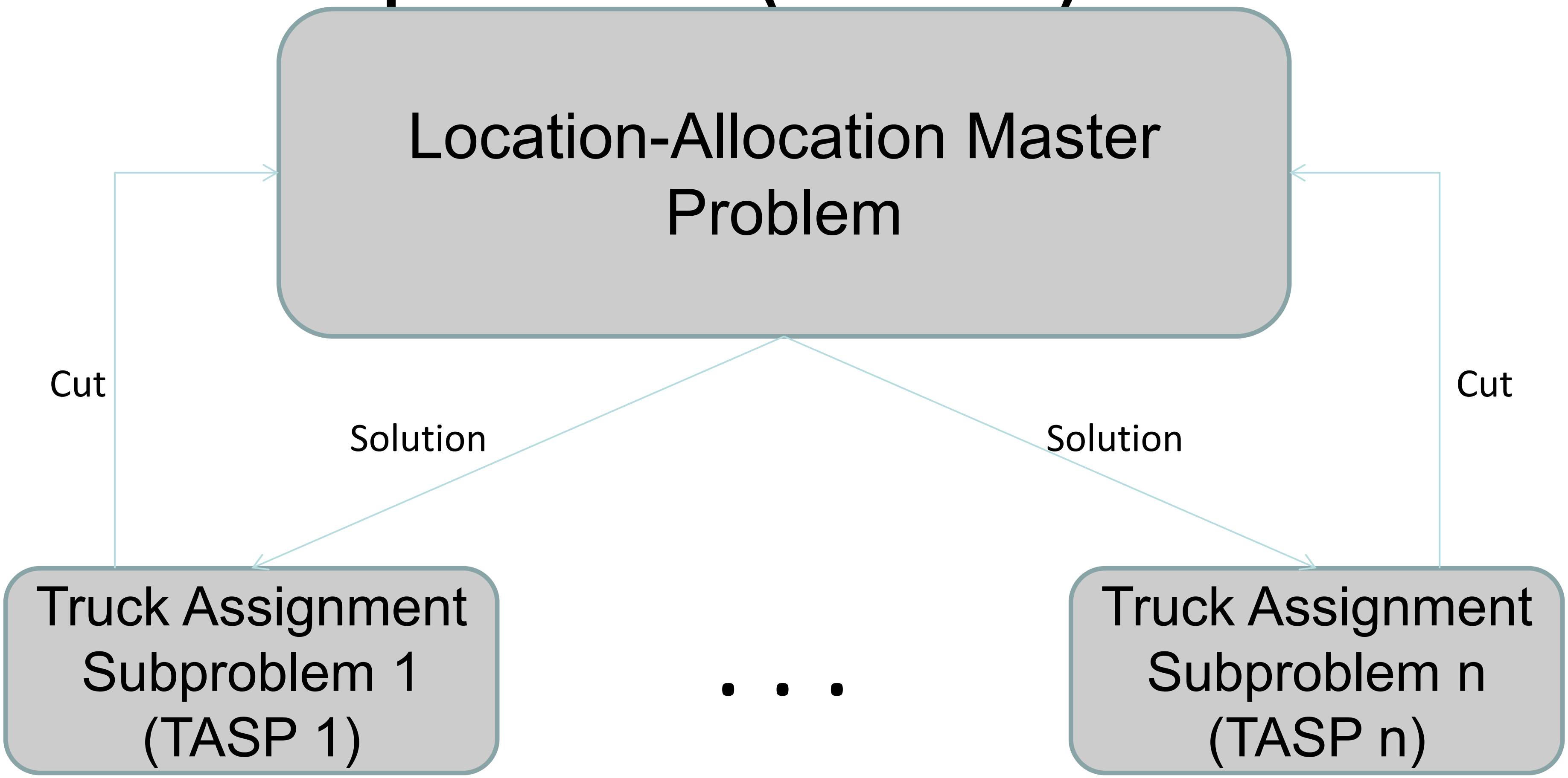
[Fazel-Zarandi & B 2012] *IJOC*, 24, 399-415, 2012.

Logic-Based Benders Decomposition (LBBD)



[Fazel-Zarandi & B 2012] *IJOC*, 24, 399-415, 2012.

Logic-Based Benders Decomposition (LBBD)



[Fazel-Zarandi & B 2012] *IJOC*, 24, 399-415, 2012.

Can We Do Better?

- Why do I have to make the truck assignment right away?
  - introduces a lot of symmetry
  - delay detailed truck assignment until we have a facility and customer assignment that looks good
- Triple index (x_{ijk}) is ugly

Change the Model

$$p_j = \begin{cases} 1: & \text{if site j is open} \\ 0: & \text{otherwise} \end{cases}$$

$$x_{ij} = \begin{cases} 1: & \text{if client i is served by site j} \\ 0: & \text{otherwise} \end{cases}$$

$numVeh_j$: The number of vehicles assigned to facility *j*

$$\min \quad \sum_{j \in J} f_j p_j + \sum_{i \in I} \sum_{j \in J} c_{ij} x_{ij} + u \sum_{j \in J} numVeh_j$$

$$
\begin{aligned}
\text{s.t.} \quad & \sum_{j \in J} x_{ij} = 1 && i \in I && (8)\\
& \sum_{i \in I} t_{ij} x_{ij} \leq l \cdot \overline{k} && j \in J && (9)\\
& t_{ij} x_{ij} \leq l && i \in I, j \in J && (10)\\
& \sum_{i \in I} d_i x_{ij} \leq b_j p_j && j \in J && (11)\\
& numVeh_j \geq \left\lceil \frac{\sum_{i \in I} t_{ij} x_{ij}}{l} \right\rceil && j \in J && (12)\\
& cuts && && (13)\\
& x_{ij} \leq p_j && i \in I, j \in J && (14)\\
& x_{ij}, p_j \in \{0, 1\}, numVeh_j \in \{0, .., \overline{k}\} && i \in I, j \in J && (15)
\end{aligned}
$$

$$\min \sum_{j \in J} f_j p_j + \sum_{i \in I} \sum_{j \in J} c_{ij} x_{ij} + u \sum_{j \in J} numVeh_j$$

$$
\begin{array}{lll}
\text{s.t.} \quad \sum_{j \in J} x_{ij} = 1 & i \in I & \\
\sum_{i \in I} t_{ij} x_{ij} \leq l \cdot \bar{k} & j \in J & (9) \\
t_{ij} x_{ij} \leq l & i \in I, j \in J & (10) \\
\sum_{i \in I} d_i x_{ij} \leq b_j p_j & j \in J & (11) \\
numVeh_j \geq \left\lceil \frac{\sum_{i \in I} t_{ij} x_{ij}}{l} \right\rceil & j \in J & (12) \\
cuts & & (13) \\
x_{ij} \leq p_j & i \in I, j \in J & (14) \\
x_{ij}, p_j \in \{0,1\}, numVeh_j \in \{0, .., \bar{k}\} & i \in I, j \in J & (15)
\end{array}
$$

Each client is served by one facility

$$\min \quad \sum_{j \in J} f_j p_j + \sum_{i \in I} \sum_{j \in J} c_{ij} x_{ij} + u \sum_{j \in J} numVeh_j$$

$$
\begin{array}{llll}
\text{s.t.} & \sum_{j \in J} x_{ij} = 1 & i \in I & \\
& \sum_{i \in I} t_{ij} x_{ij} \leq l \cdot \overline{k} & j \in J & \\
& t_{ij} x_{ij} \leq l & i \in I, j \in J & \\
& \sum_{i \in I} d_i x_{ij} \leq b_j p_j & j \in J & (11) \\
& numVeh_j \geq \left\lceil \frac{\sum_{i \in I} t_{ij} x_{ij}}{l} \right\rceil & j \in J & (12) \\
& cuts & & (13) \\
& x_{ij} \leq p_j & i \in I, j \in J & (14) \\
& x_{ij}, p_j \in \{0,1\}, numVeh_j \in \{0,..,\overline{k}\} & i \in I, j \in J & (15)
\end{array}
$$

Each client is served by one facility

Distance Constraints

$$\min \sum_{j\in J} f_j p_j + \sum_{i\in I}\sum_{j\in J} c_{ij}x_{ij} + u\sum_{j\in J} numVeh_j$$

$$
\begin{aligned}
\text{s.t.}\quad & \sum_{j\in J} x_{ij} = 1 && i \in I \\
& \sum_{i\in I} t_{ij}x_{ij} \le l\cdot\bar{k} && j \in J \\
& t_{ij}x_{ij} \le l && i \in I, j \in J \\
& \sum_{i\in I} d_i x_{ij} \le b_j p_j && j \in J \\
& numVeh_j \ge \left\lceil \frac{\sum_{i\in I} t_{ij}x_{ij}}{l} \right\rceil && j \in J && (12) \\
& cuts && && (13) \\
& x_{ij} \le p_j && i \in I, j \in J && (14) \\
& x_{ij}, p_j \in \{0,1\}, numVeh_j \in \{0,..,\bar{k}\} && i \in I, j \in J && (15)
\end{aligned}
$$

Each client is served by one facility

Distance Constraints

Capacity Constraint

$$\min \sum_{j \in J} f_j p_j + \sum_{i \in I} \sum_{j \in J} c_{ij} x_{ij} + u \sum_{j \in J} numVeh_j$$

$$\text{s.t.} \quad \sum_{j \in J} x_{ij} = 1 \qquad i \in I$$

Each client is served by one facility

$$\sum_{i \in I} t_{ij} x_{ij} \leq l \cdot \overline{k} \qquad j \in J$$

$$t_{ij} x_{ij} \leq l \qquad i \in I, j \in J$$

Distance Constraints

$$\sum_{i \in I} d_i x_{ij} \leq b_j p_j \qquad j \in J$$

Capacity Constraint

$$numVeh_j \geq \left\lceil \frac{\sum_{i \in I} t_{ij} x_{ij}}{l} \right\rceil \qquad j \in J \qquad (12)$$

$$cuts \qquad (13)$$

$$x_{ij} \leq p_j \qquad i \in I, j \in J \qquad (14)$$

$$x_{ij}, p_j \in \{0, 1\}, numVeh_j \in \{0, .., \overline{k}\} \qquad i \in I, j \in J \qquad (15)$$

$$\min \quad \sum_{j \in J} f_j p_j + \sum_{i \in I} \sum_{j \in J} c_{ij} x_{ij} + u \sum_{j \in J} numVeh_j$$

$$\text{s.t.} \quad \sum_{j \in J} x_{ij} = 1 \qquad i \in I$$

Each client is served by one facility

$$\sum_{i \in I} t_{ij} x_{ij} \le l \cdot \bar{k} \qquad j \in J$$

$$t_{ij} x_{ij} \le l \qquad i \in I, j \in J$$

Distance Constraints

$$\sum_{i \in I} d_i x_{ij} \le b_j p_j \qquad j \in J$$

Capacity Constraint

$$numVeh_j \ge \left\lceil \frac{\sum_{i \in I} t_{ij} x_{ij}}{l} \right\rceil \qquad j \in J \qquad (12)$$

$$cuts \qquad (13)$$

We'll talk about these later…

$$x_{ij} \le p_j \qquad i \in I, j \in J \qquad (14)$$

$$x_{ij}, p_j \in \{0, 1\}, numVeh_j \in \{0, .., \bar{k}\} \qquad i \in I, j \in J \qquad (15)$$

Solving this model, we get:

- The open facilities (p_j)
- The assignment of customers to facilities (x_{ij})
- The number of trucks at each facility ($numVeh_j$)

Solving this model, we get:

- The open facilities (p_j)
- The assignment of customers to facilities (x_{ij})
- The number of trucks at each facility ($numVeh_j$)

So we're done, right?

A Problem

- The customers assigned to a facility might not fit in the trucks we have allocated to that facility

$$numVeh_j \geq \left\lceil \frac{\sum_{i \in I} t_{ij} x_{ij}}{l} \right\rceil \quad j \in J$$

Truck Assignment Subproblem (TASP)

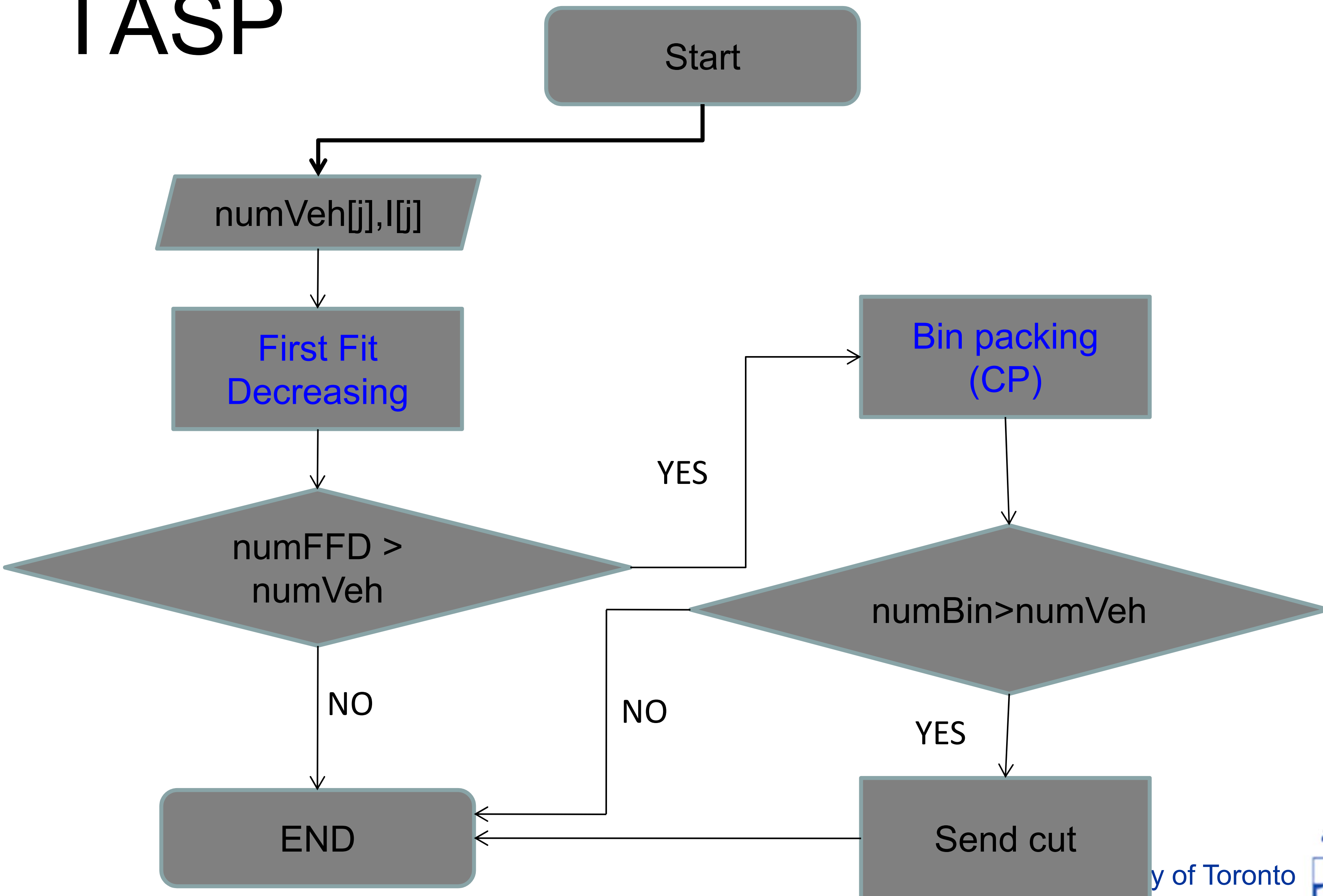
- Given: Assigned clients & number of vehicles at each open facility
- Goal: Assign clients to vehicles such that the vehicle distance constraints are satisfied

Truck Assignment Subproblem (TASP)

- Given: Assigned clients & number of vehicles at each open facility
- Goal: Assign clients to vehicles such that the vehicle distance constraints are satisfied

TASP = bin packing
[distance = "capacity"]

TASP



y of Toronto
Mechanical & Industrial Engineering

Bin Packing Using CP

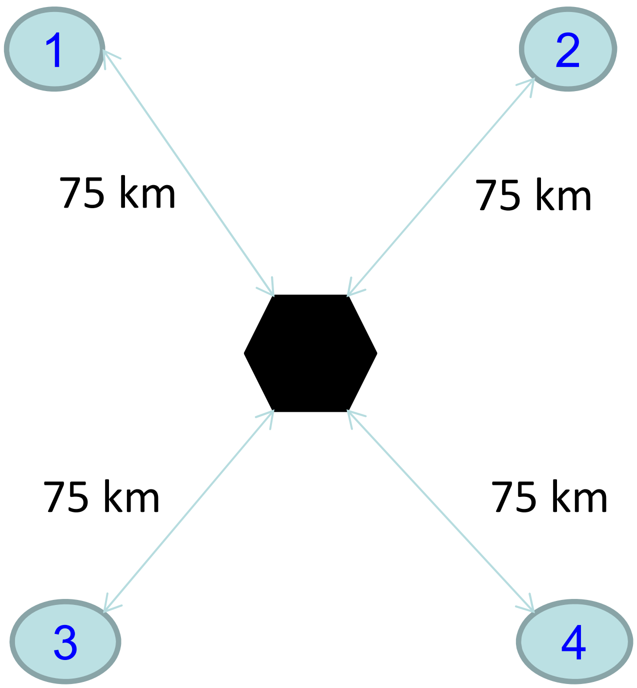
$$\min \ numVehBinPacking_j$$

$$S.t:$$

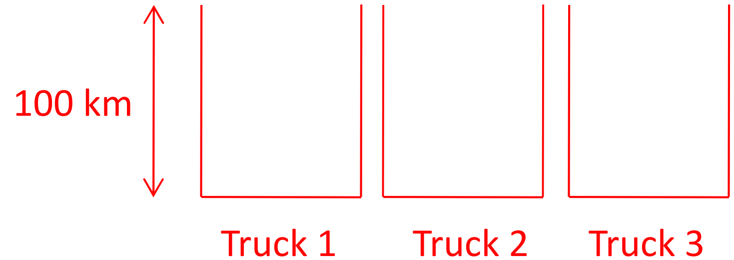
$$pack(vehicleDist, I_j, dist_{ij})$$

$$numVeh_j \leq numVehBinPacking_j \leq numVehFFD_j$$

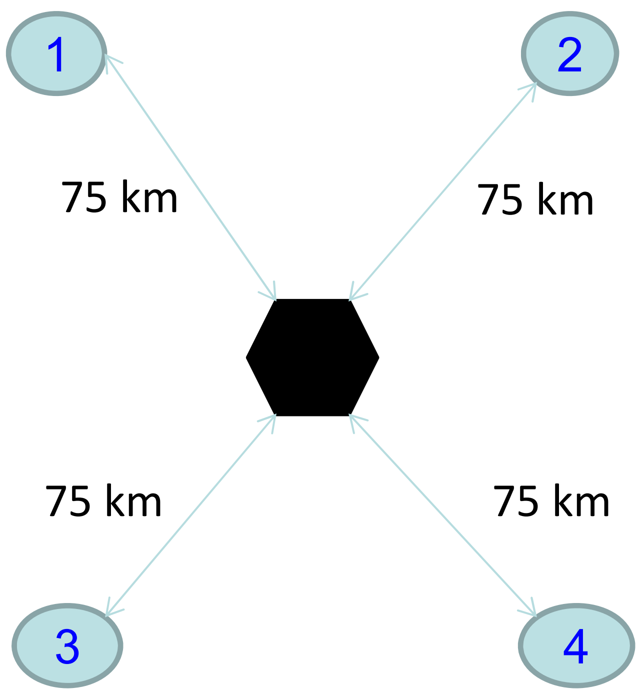
What about the cut? $\left\lceil \frac{\sum t_{ij}}{l} \right\rceil = \left\lceil \frac{4 \times 75}{100} \right\rceil = 3$



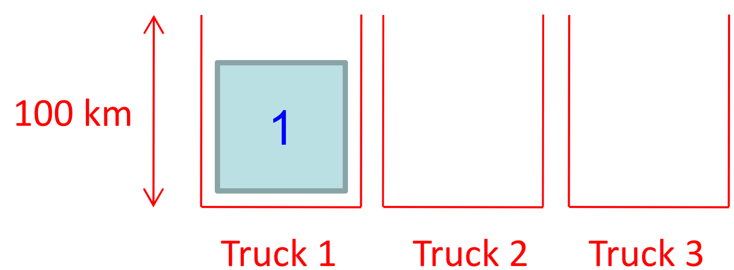
Truck distance capacity (l)= 100 km



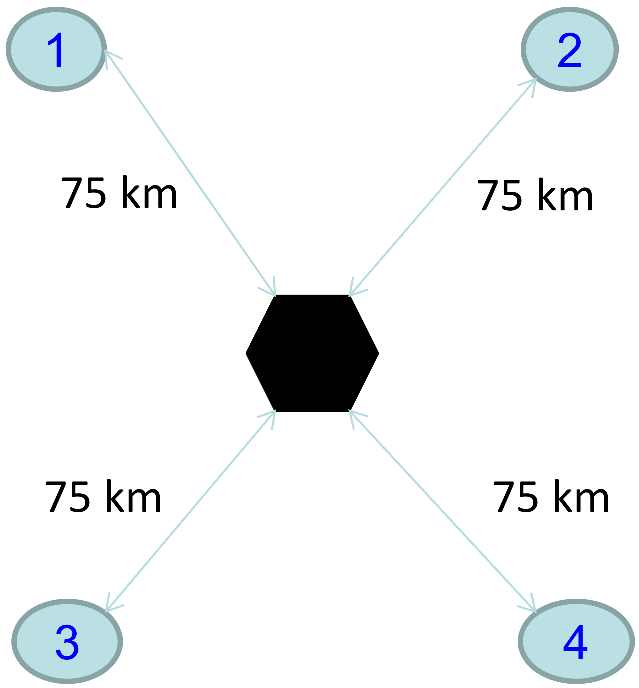
What about the cut? $\left\lceil \frac{\sum t_{ij}}{l} \right\rceil = \left\lceil \frac{4 \times 75}{100} \right\rceil = 3$



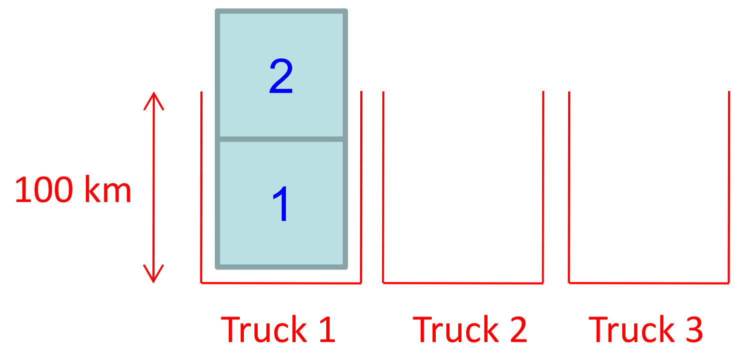
Truck distance capacity (l)= 100 km



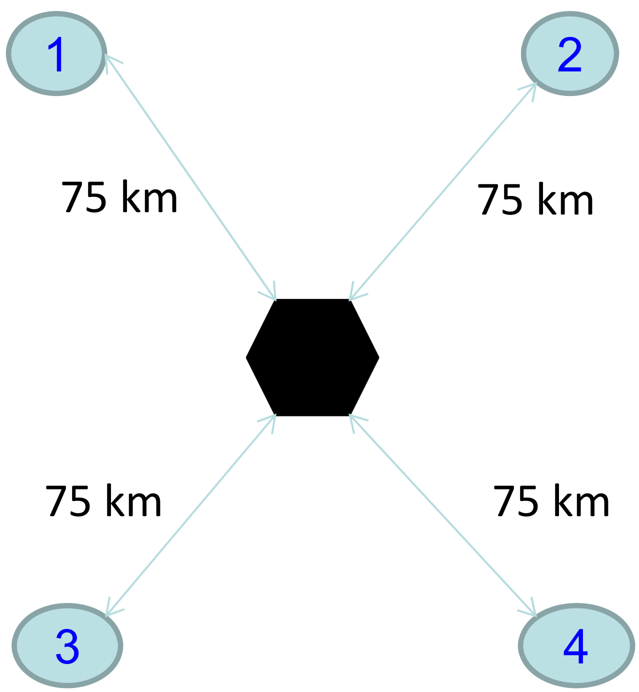
What about the cut? $\left\lceil \frac{\sum t_{ij}}{l} \right\rceil = \left\lceil \frac{4 \times 75}{100} \right\rceil = 3$



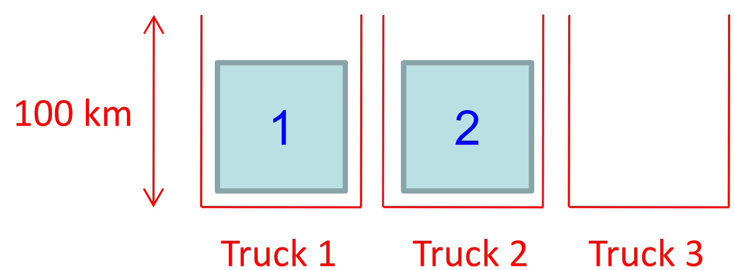
Truck distance capacity (l)= 100 km



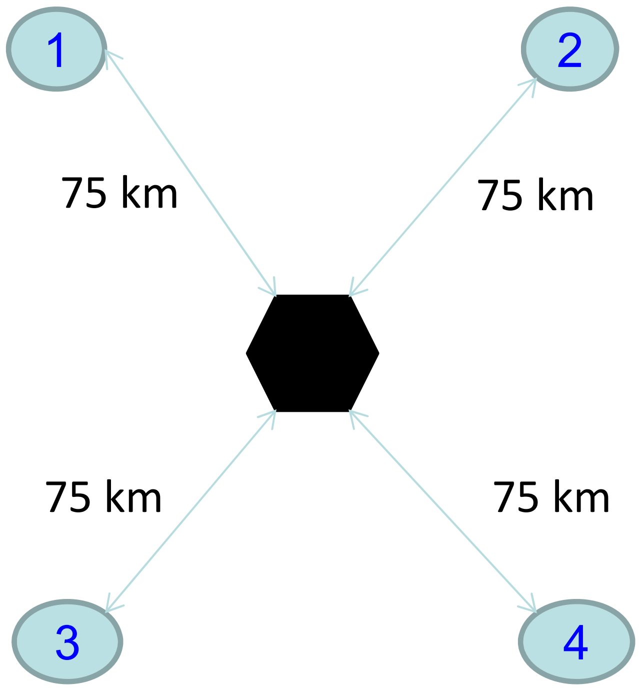
What about the cut? $\left\lceil \frac{\sum t_{ij}}{l} \right\rceil = \left\lceil \frac{4 \times 75}{100} \right\rceil = 3$



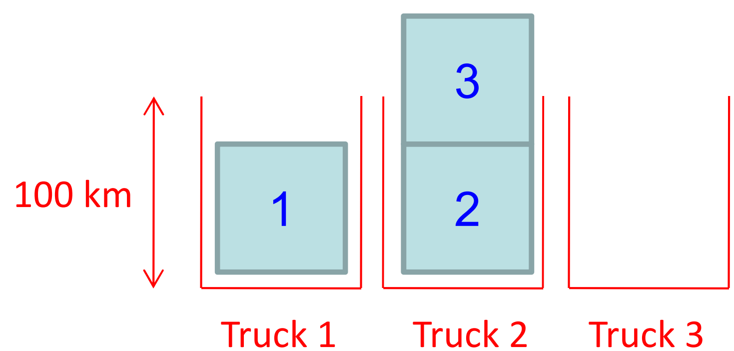
Truck distance capacity (l)= 100 km



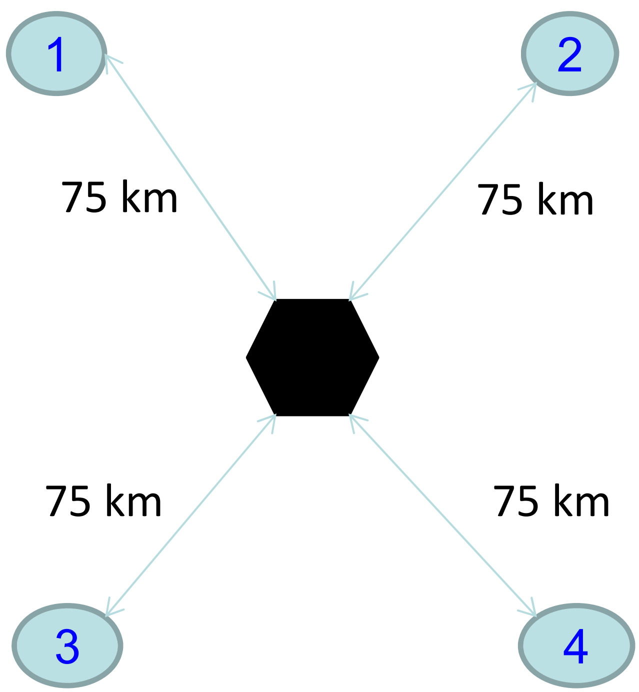
What about the cut? $\left\lceil \frac{\sum t_{ij}}{l} \right\rceil = \left\lceil \frac{4 \times 75}{100} \right\rceil = 3$



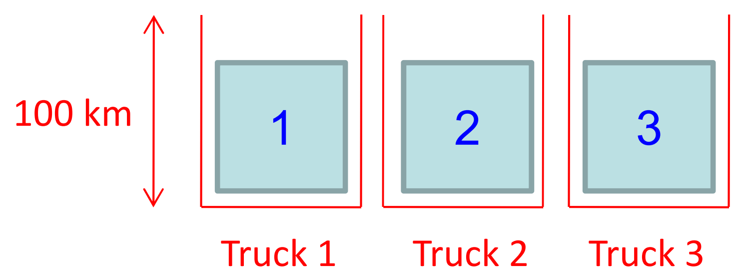
Truck distance capacity (l)= 100 km



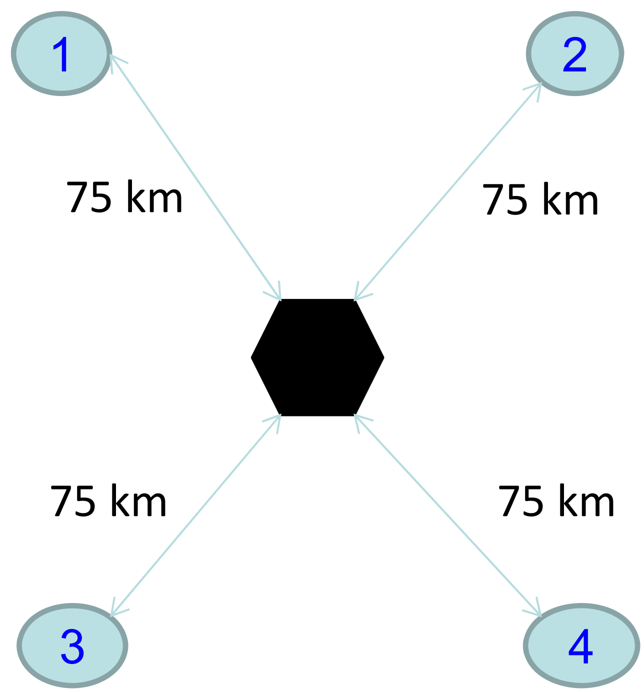
What about the cut? $\left\lceil \frac{\sum t_{ij}}{l} \right\rceil = \left\lceil \frac{4 \times 75}{100} \right\rceil = 3$



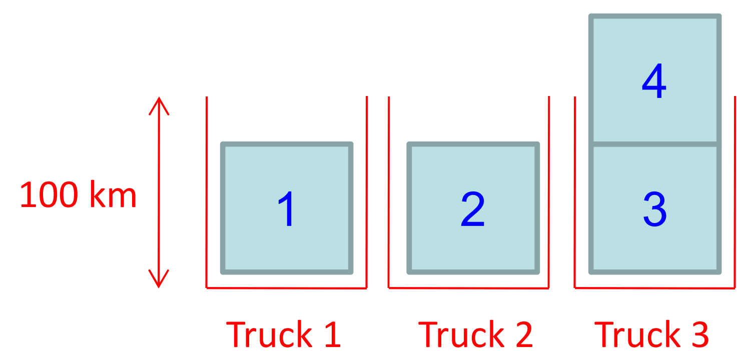
Truck distance capacity (l)= 100 km



What about the cut? $\left\lceil \frac{\sum t_{ij}}{l} \right\rceil = \left\lceil \frac{4 \times 75}{100} \right\rceil = 3$

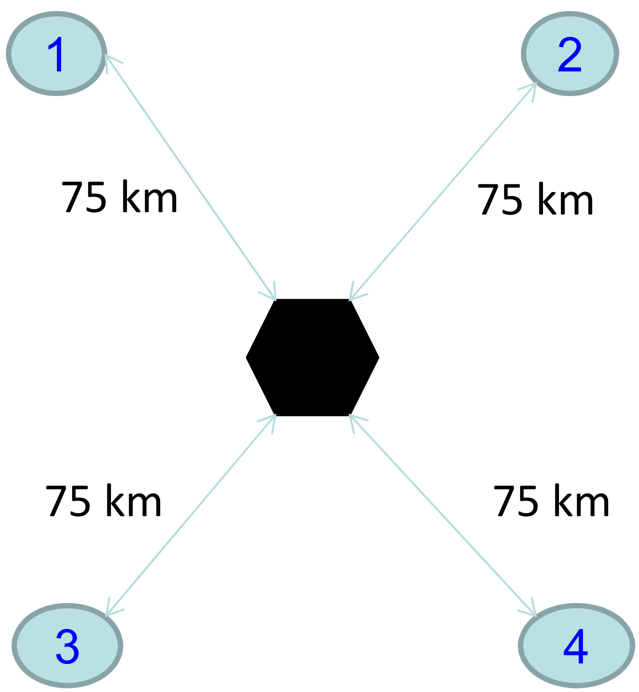


Truck distance capacity (l)= 100 km

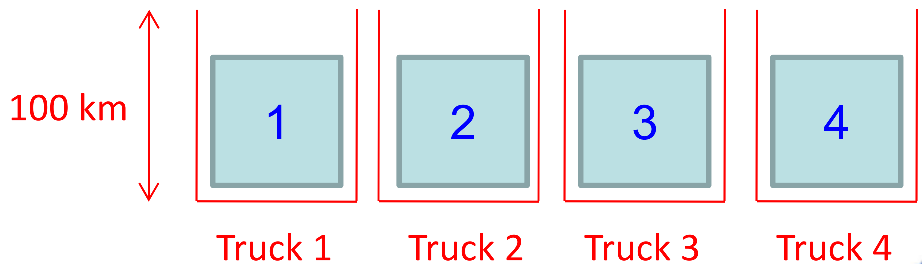


University of Toronto
Mechanical & Industrial Engineering

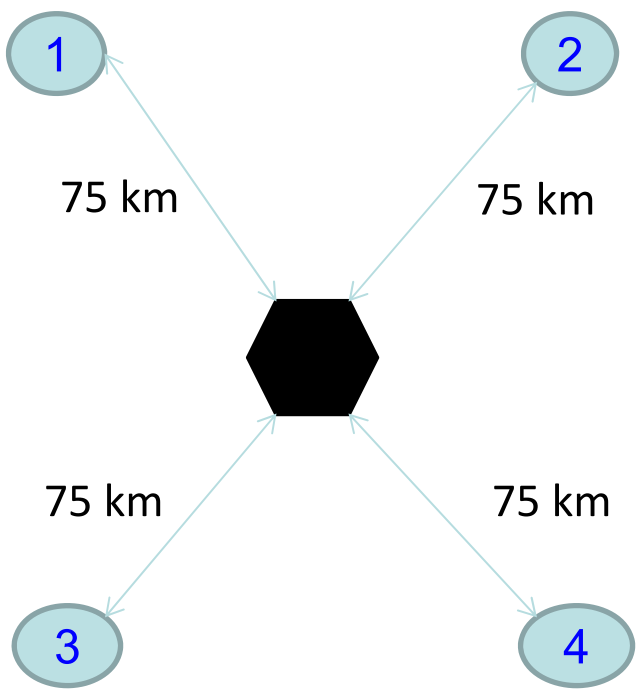
What about the cut? $\left\lceil \frac{\sum t_{ij}}{l} \right\rceil = \left\lceil \frac{4 \times 75}{100} \right\rceil = 3$



Truck distance capacity (l)= 100 km

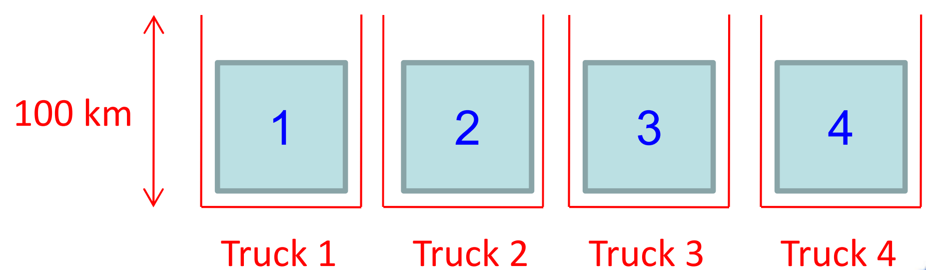


What about the cut? $\left\lceil \frac{\sum t_{ij}}{l} \right\rceil = \left\lceil \frac{4 \times 75}{100} \right\rceil = 3$

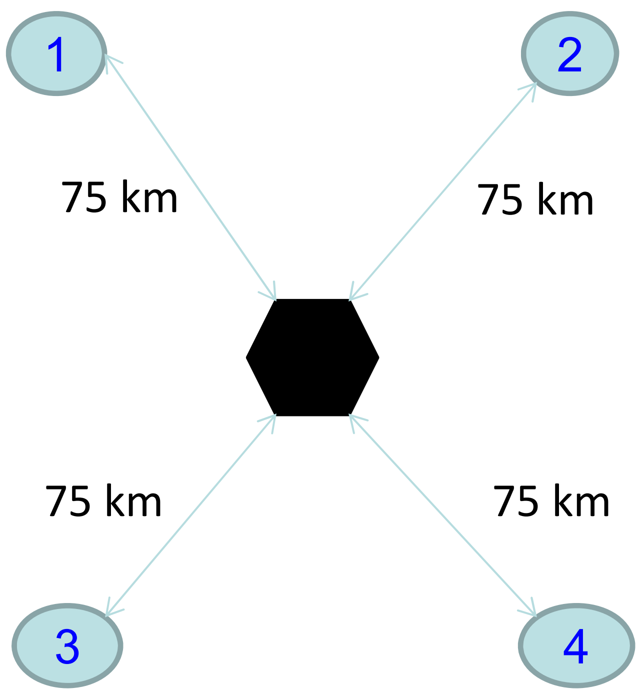


Truck distance capacity (l)= 100 km

$$numVeh_j \geq numVeh^*_{jh} - \sum_{i \in I_h} (1 - x_{ij})$$



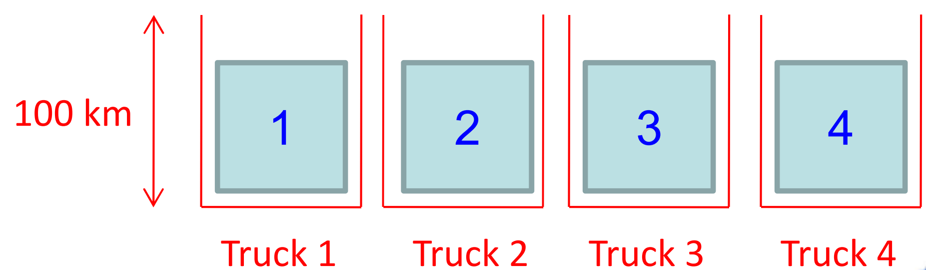
What about the cut? $\left\lceil \frac{\sum t_{ij}}{l} \right\rceil = \left\lceil \frac{4 \times 75}{100} \right\rceil = 3$



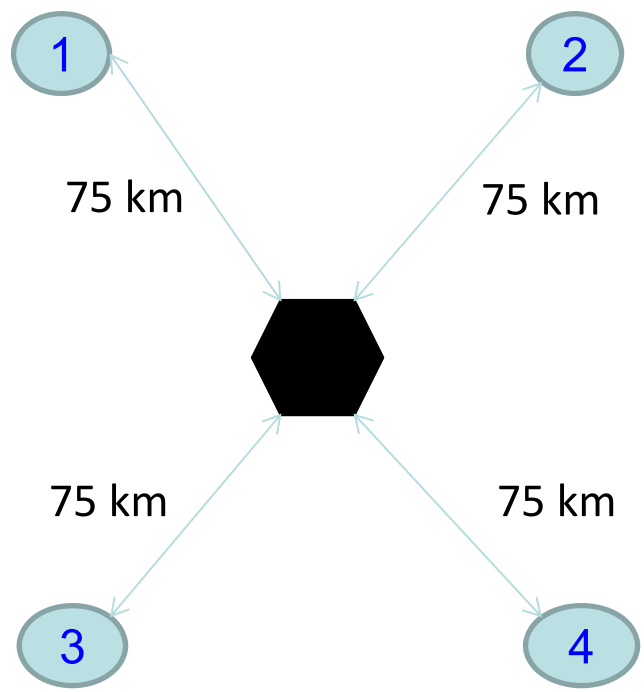
Truck distance capacity (l)= 100 km

$$numVeh_j \geq numVeh^*_{jh} - \sum_{i \in I_h} (1 - x_{ij})$$

4 trucks



What about the cut? $\left\lceil \frac{\sum t_{ij}}{l} \right\rceil = \left\lceil \frac{4 \times 75}{100} \right\rceil = 3$

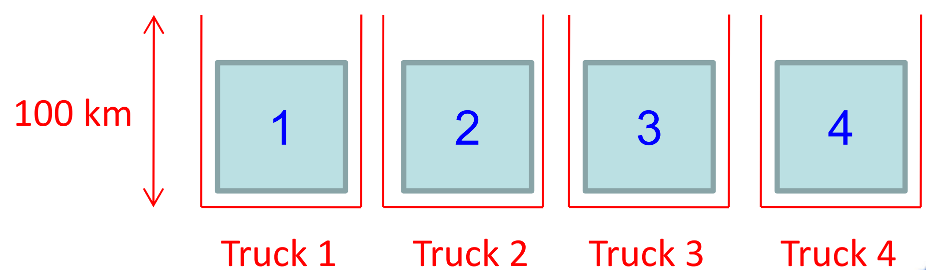


Truck distance capacity (l)= 100 km

$$numVeh_j \geq numVeh^*_{jh} - \sum_{i \in I_h} (1 - x_{ij})$$

4 trucks

1 truck



Cuts

- Constraints added to the MP each time one of the sub-problems is not able to find a feasible solution

$$numVeh_j \geq numVeh_{jh}^{*} - \sum_{i \in I_h} \left(1 - x_{ij}\right) \quad j \in J_h$$

Cuts

- Constraints added to the MP each time one of the sub-problems is not able to find a feasible solution

$$numVeh_j \geq numVeh_{jh}^* - \sum_{i \in I_h} (1 - x_{ij}) \quad j \in J_h$$

vehicles at site *j* assigned in iteration *h*

Cuts

- Constraints added to the MP each time one of the sub-problems is not able to find a feasible solution

$$numVeh_j \geq numVeh^*_{jh} - \sum_{i \in I_h} \left(1 - x_{ij}\right) \quad j \in J_h$$

vehicles at site *j* assigned in iteration *h*

Max. decrease in the # vehicles needed given reassigned clients

$$\min \sum_{j \in J} f_j p_j + \sum_{i \in I} \sum_{j \in J} c_{ij} x_{ij} + u \sum_{j \in J} numVeh_j$$

$$\text{s.t.} \quad \sum_{j \in J} x_{ij} = 1 \qquad i \in I$$

Each client is served by one facility

$$\sum_{i \in I} t_{ij} x_{ij} \le l \cdot \overline{k} \qquad j \in J$$

$$t_{ij} x_{ij} \le l \qquad i \in I, j \in J$$

Distance Constraints

$$\sum_{i \in I} d_i x_{ij} \le b_j p_j \qquad j \in J$$

Capacity Constraint

$$numVeh_j \ge \left\lceil \frac{\sum_{i \in I} t_{ij} x_{ij}}{l} \right\rceil \qquad j \in J \qquad (12)$$

$$cuts \qquad (13)$$

$$x_{ij} \le p_j \qquad i \in I, j \in J \qquad (14)$$

$$x_{ij}, p_j \in \{0, 1\}, numVeh_j \in \{0, .., \overline{k}\} \qquad i \in I, j \in J \qquad (15)$$

$$\min \sum_{j\in J} f_j p_j + \sum_{i\in I}\sum_{j\in J} c_{ij}x_{ij} + u\sum_{j\in J} numVeh_j$$

$$
\begin{aligned}
\text{s.t.}\quad & \sum_{j\in J} x_{ij} = 1 && i\in I \\
& \sum_{i\in I} t_{ij}x_{ij} \le l\cdot \bar{k} && j\in J \\
& t_{ij}x_{ij} \le l && i\in I, j\in J \\
& \sum_{i\in I} d_i x_{ij} \le b_j p_j && j\in J \\
& numVeh_j \ge \left\lceil \frac{\sum_{i\in I} t_{ij}x_{ij}}{l} \right\rceil && j\in J \\
& cuts && && (13) \\
& x_{ij} \le p_j && i\in I, j\in J && (14) \\
& x_{ij}, p_j \in \{0,1\}, numVeh_j \in \{0,..,\bar{k}\} && i\in I, j\in J && (15)
\end{aligned}
$$

Each client is served by one facility

Distance Constraints

Capacity Constraint

Sub-problem Relaxation

$$\min \sum_{j \in J} f_j p_j + \sum_{i \in I} \sum_{j \in J} c_{ij} x_{ij} + u \sum_{j \in J} numVeh_j$$

$$\text{s.t.} \quad \sum_{j \in J} x_{ij} = 1 \qquad i \in I$$

Each client is served by one facility

$$\sum_{i \in I} t_{ij} x_{ij} \le l \cdot \bar{k} \qquad j \in J$$

$$t_{ij} x_{ij} \le l \qquad i \in I, j \in J$$

Distance Constraints

$$\sum_{i \in I} d_i x_{ij} \le b_j p_j \qquad j \in J$$

Capacity Constraint

$$numVeh_j \ge \left\lceil \frac{\sum_{i \in I} t_{ij} x_{ij}}{l} \right\rceil \qquad j \in J$$

Sub-problem Relaxation

$$cuts$$

Benders cuts

$$x_{ij} \le p_j \qquad i \in I, j \in J \quad (14)$$

$$x_{ij}, p_j \in \{0,1\}, numVeh_j \in \{0, .., \bar{k}\} \qquad i \in I, j \in J \quad (15)$$

Location-Allocation Master Problem (LAMP)

$$\min \sum_{j \in J} f_j p_j + \sum_{i \in I} \sum_{j \in J} c_{ij} x_{ij} + u \sum_{j \in J} numVeh_j$$

$$\text{s.t.} \quad \sum_{j \in J} x_{ij} = 1 \qquad i \in I$$

Each client is served by one facility

$$\sum_{i \in I} t_{ij} x_{ij} \leq l \cdot \overline{k} \qquad j \in J$$

$$t_{ij} x_{ij} \leq l \qquad i \in I, j \in J$$

Distance Constraints

$$\sum_{i \in I} d_i x_{ij} \leq b_j p_j \qquad j \in J$$

Capacity Constraint

$$numVeh_j \geq \left\lceil \frac{\sum_{i \in I} t_{ij} x_{ij}}{l} \right\rceil \qquad j \in J$$

Sub-problem Relaxation

$$cuts$$

Benders cuts

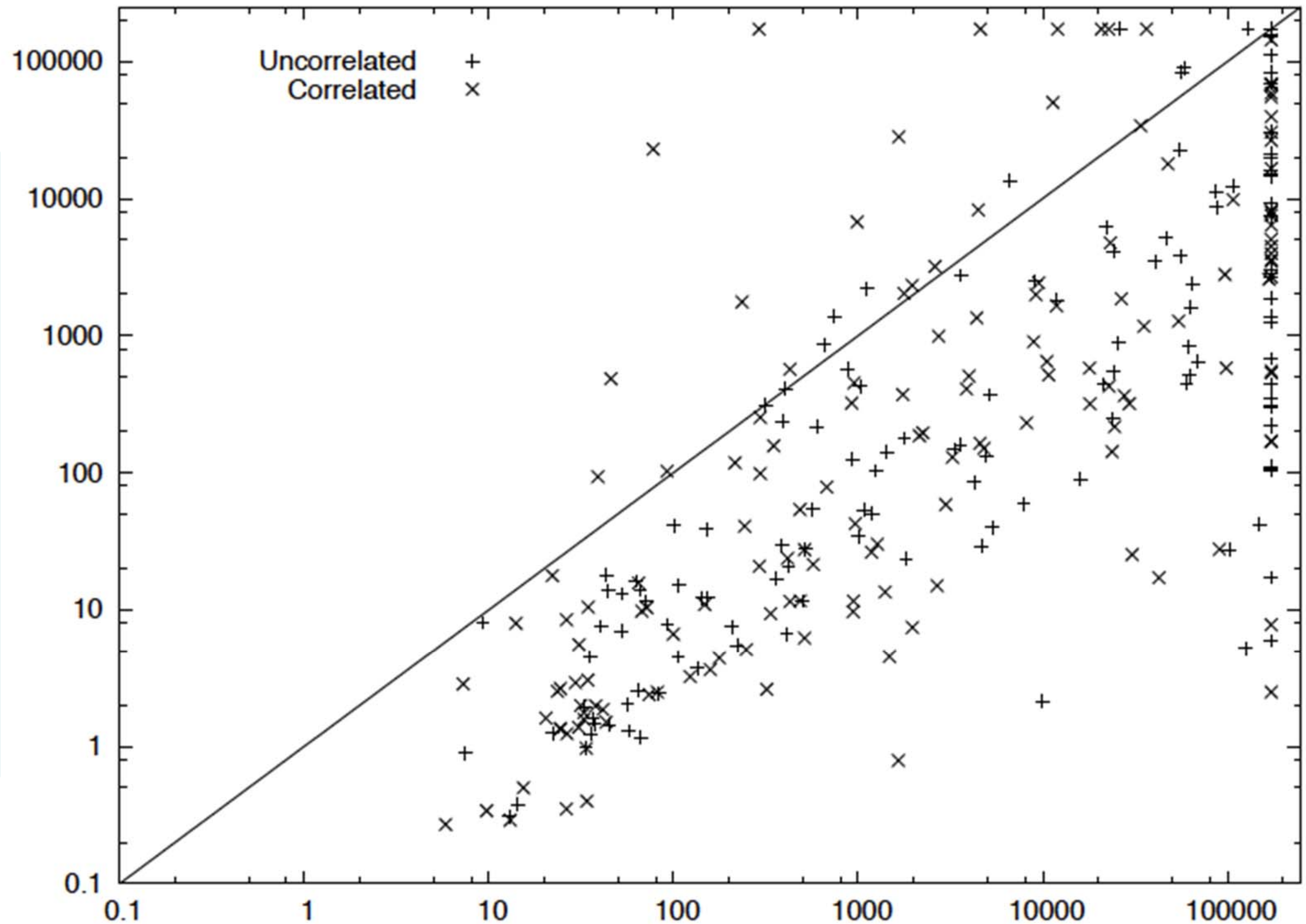
$$x_{ij} \leq p_j \qquad i \in I, j \in J \quad (14)$$

$$x_{ij}, p_j \in \{0, 1\}, numVeh_j \in \{0, .., \overline{k}\} \qquad i \in I, j \in J \quad (15)$$

LBBD vs IP

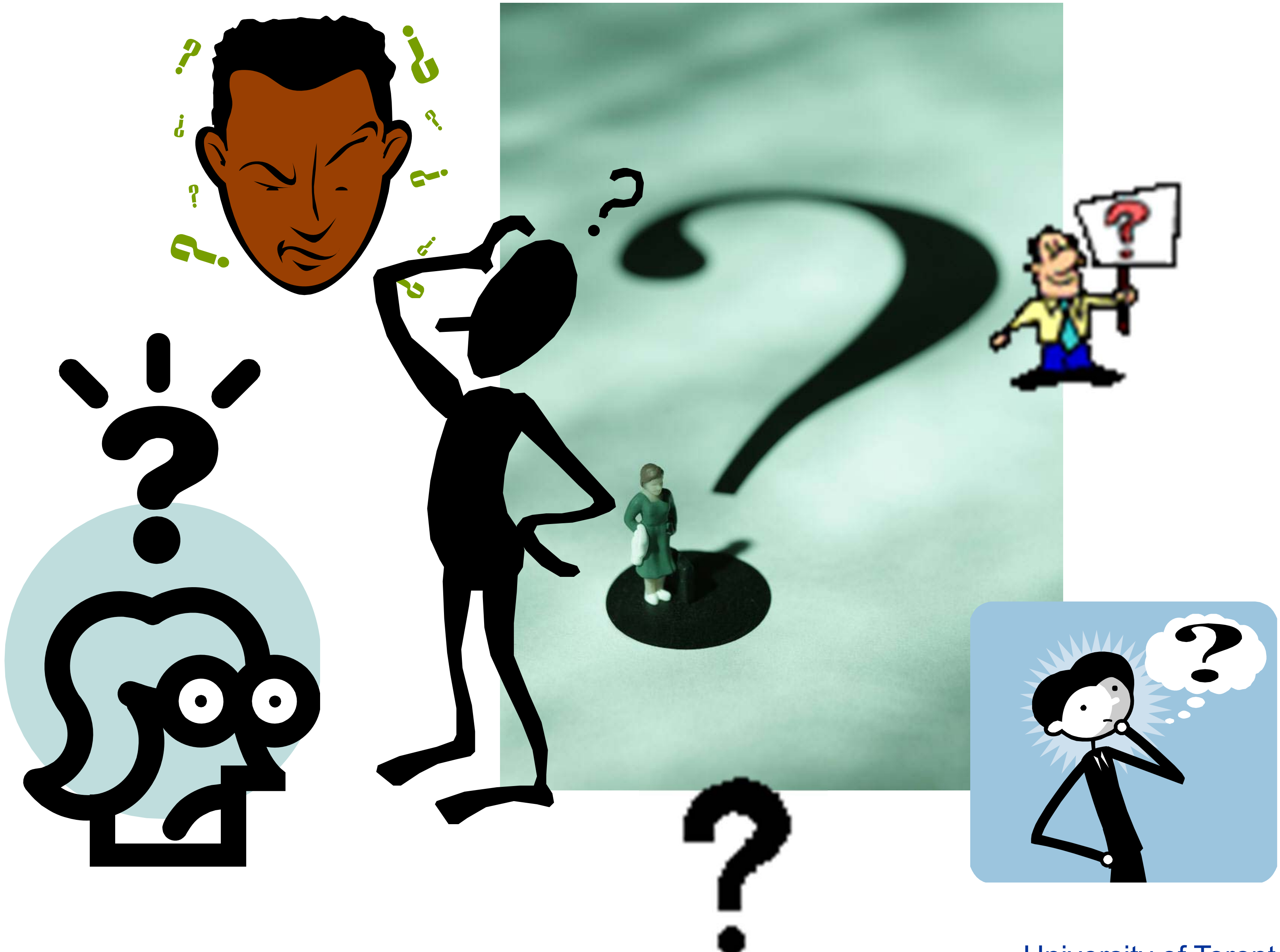
LBBD > 300 times faster than IP

LBBD CPU Time

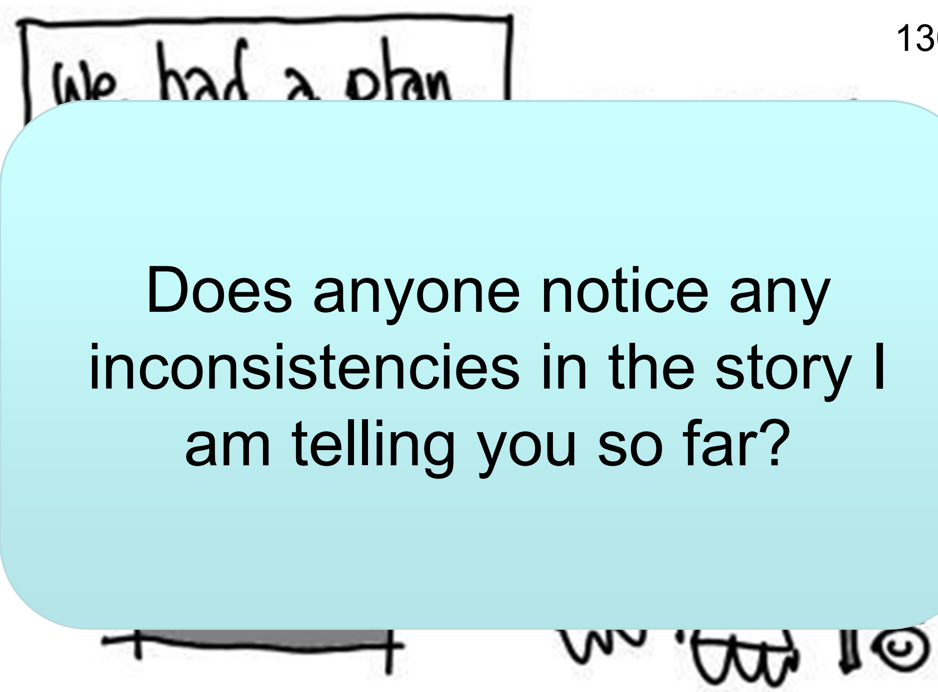


IP CPU Time

[Fazel-Zarandi & B 2012] *IJOC*, 24, 399-415, 2012.



The Plan



Does anyone notice any inconsistencies in the story I am telling you so far?

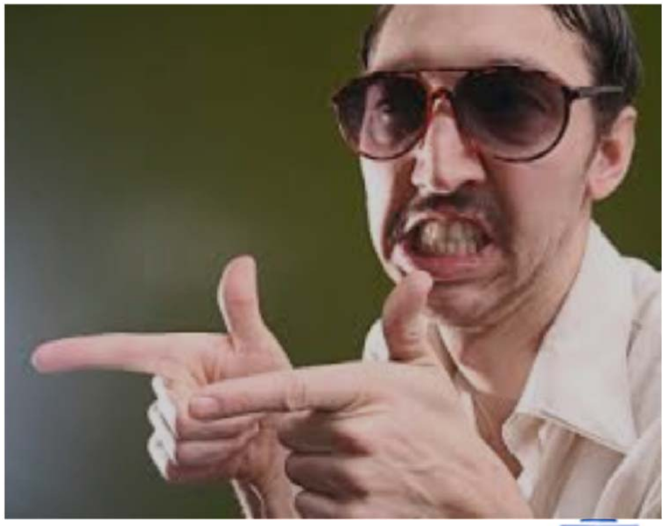
- Decomposition & Modeling
- Logic-based Benders Decomposition (LBBD)
- Applying LBBD to Problems Somewhat Related to Computational Sustainability
- Beyond Decomposition

Decomposition

- Hierarchical (the standard way)
  - overall problem is split into sub-problems solved one at at time or independently
    - e.g., infrastructure layout after turbine placement
  - no feedback
- Integrated
  - decisions really depend on each other but problem too big to solve in one model
  - decomposition with feedback

Decomposition

- Hierarchical (the standard way)
  - overall problem is split into sub-problems solved one at at time or independently
    - e.g., infrastructure layout after turbine placement
  - no feedback
- Integrated
  - decisions really depend on each other but problem too big to solve in one model
  - decomposition with feedback



A Weakness in My Story

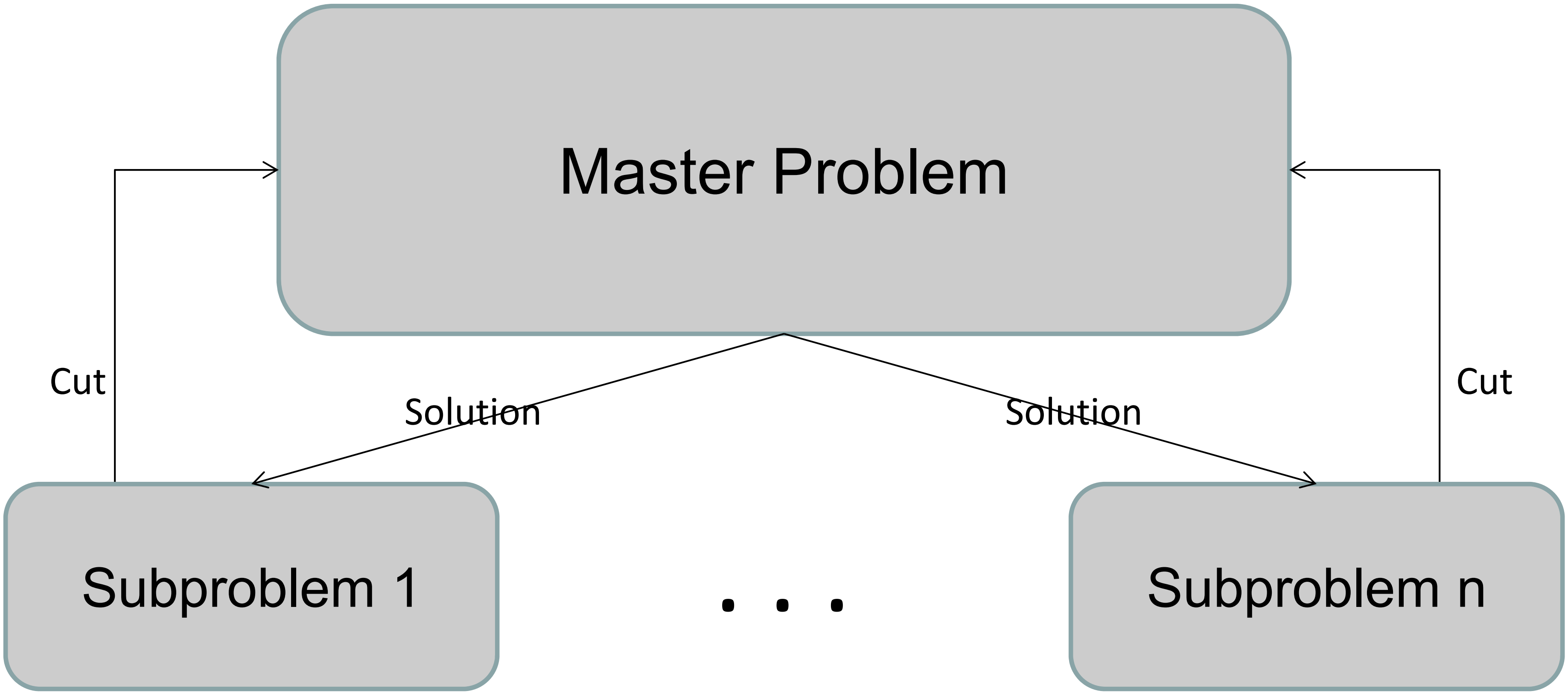
- Motivation was about taking really big complex problems and decomposing
- But all my examples have really been "small" problems (the type we normally solve in CP/AI/OR)
  - e.g., all the aspects of building a wind farm not just turbine placement

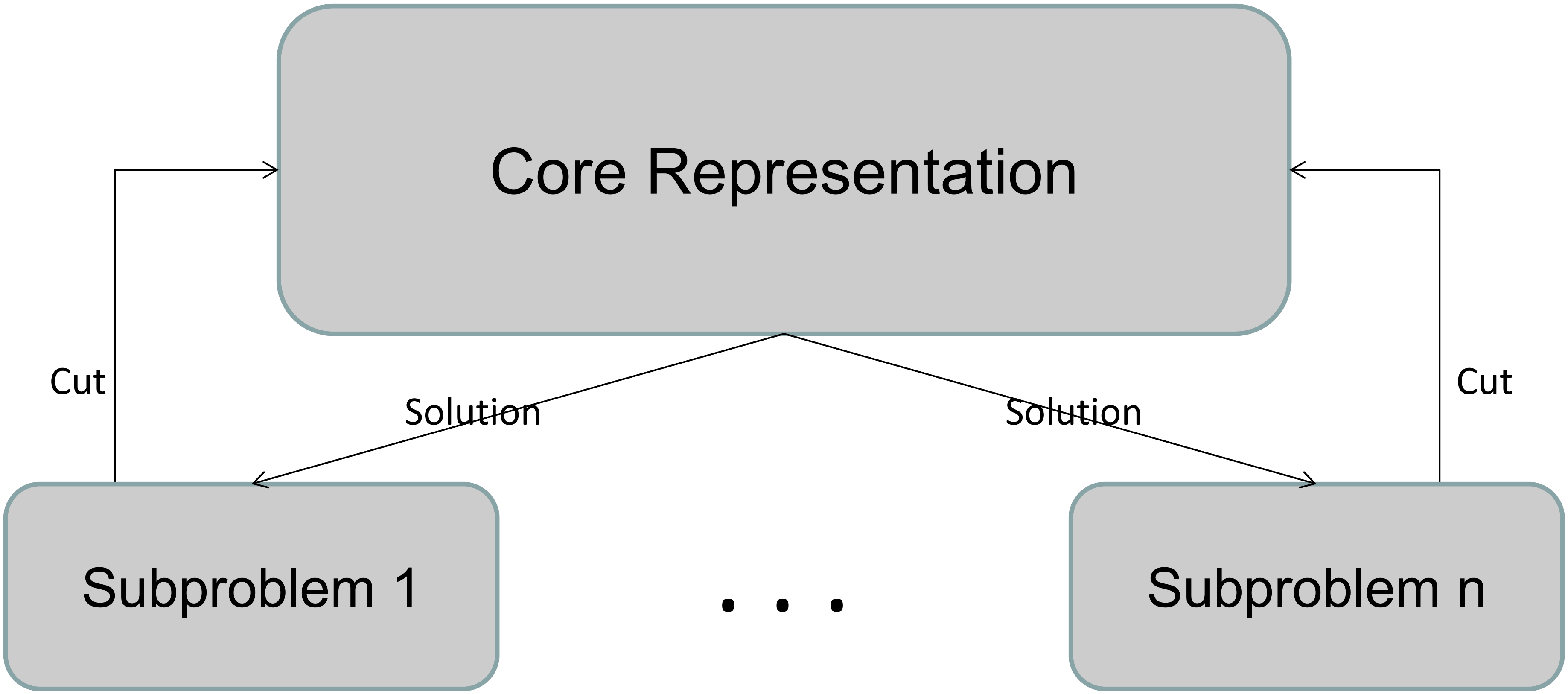
A Challenge

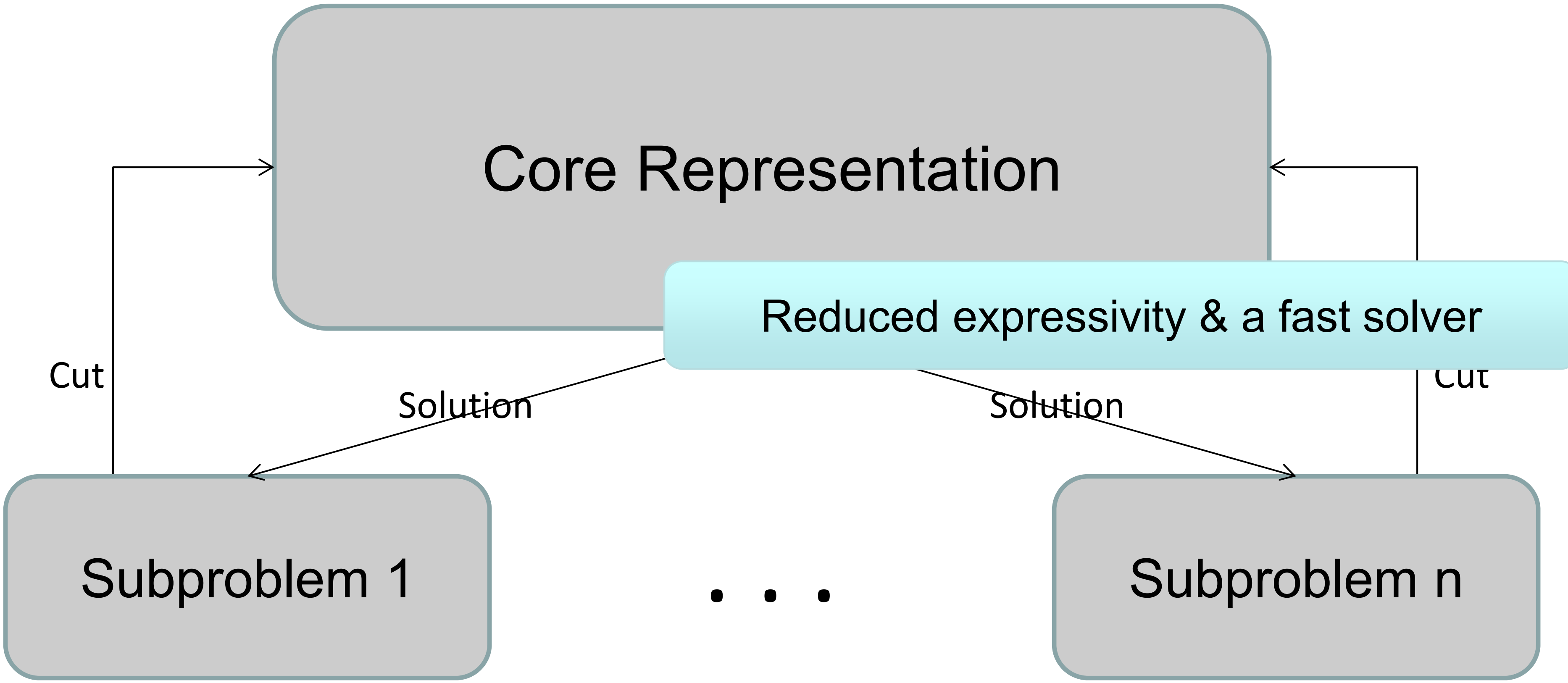
- Rather than decomposing what we already see as a single problem, can we unify what we think of as separate problems?

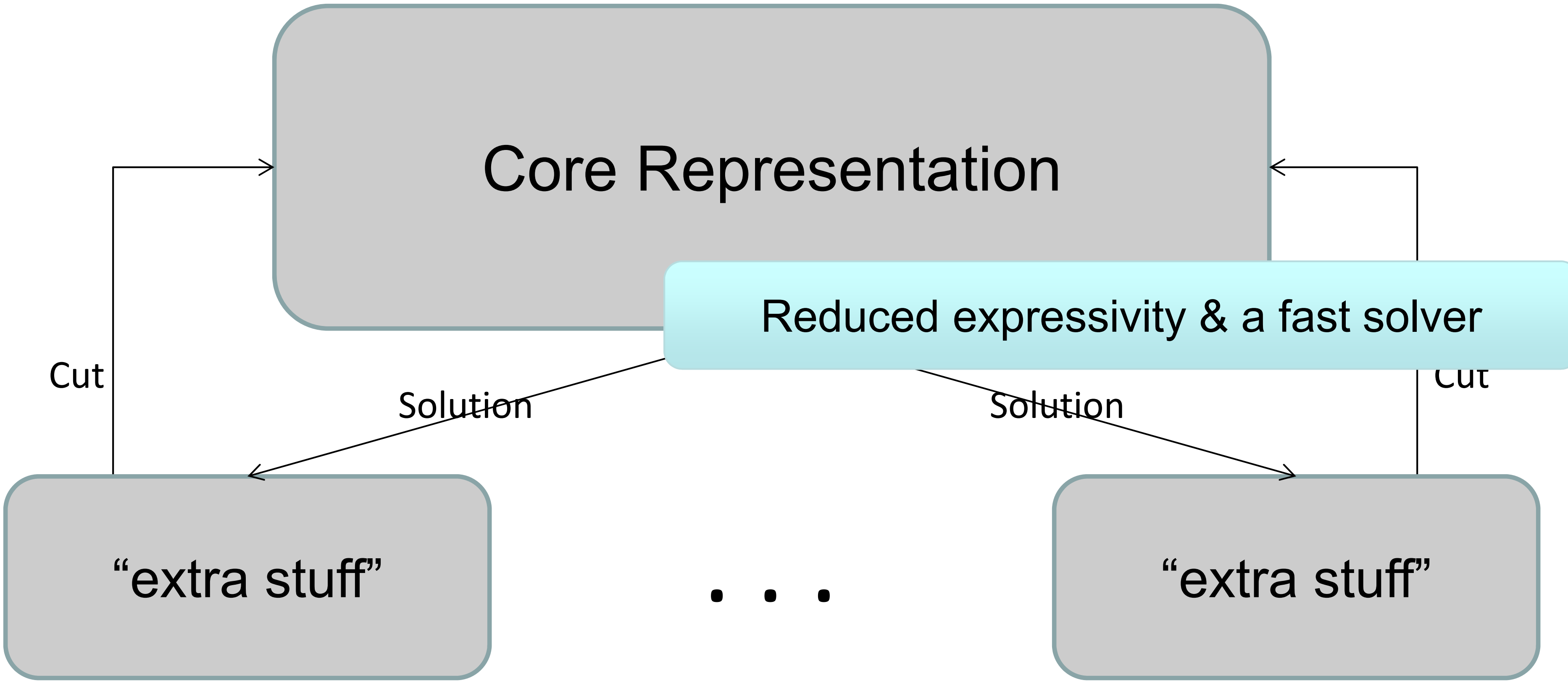
Directions

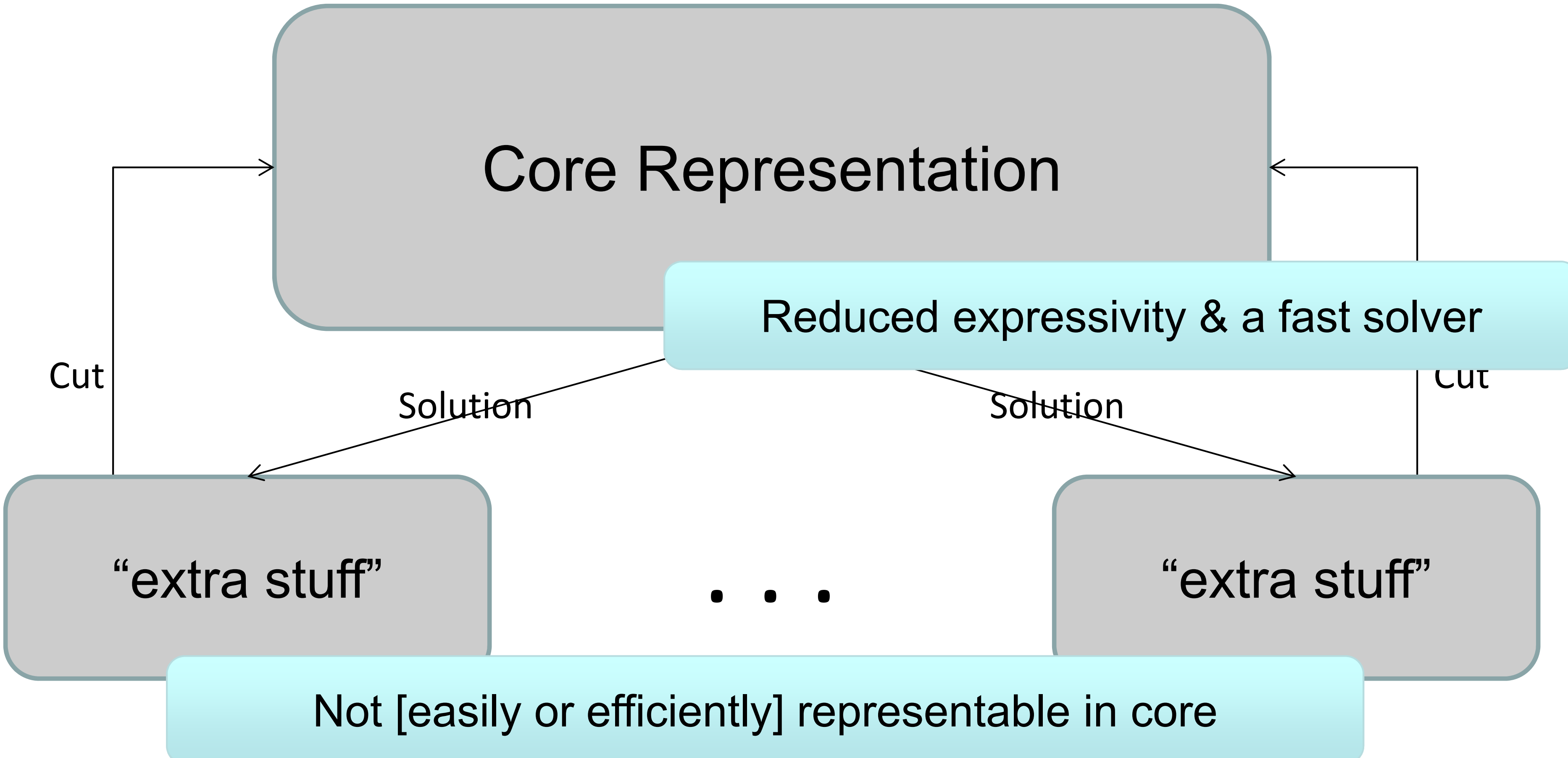
- Integrating maintenance planning and production scheduling
  - long-term stochastic reasoning combined with short-term combinatorial reasoning
    [Aramon Bajestani, forthcoming] PhD dissertation, University of Toronto

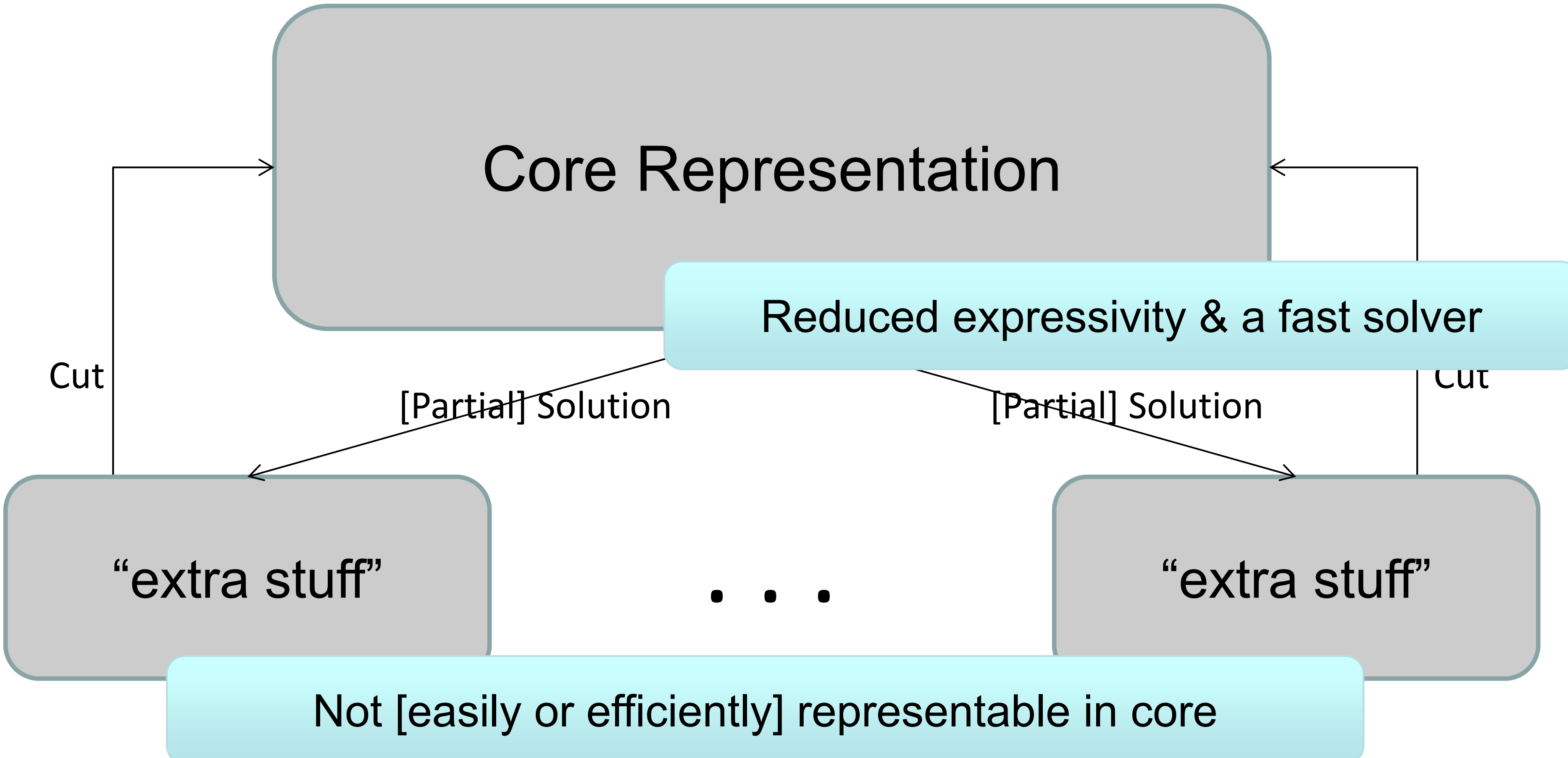

Master Problem
Cut
Solution
Solution
Cut
Subproblem 1
. . .
Subproblem n

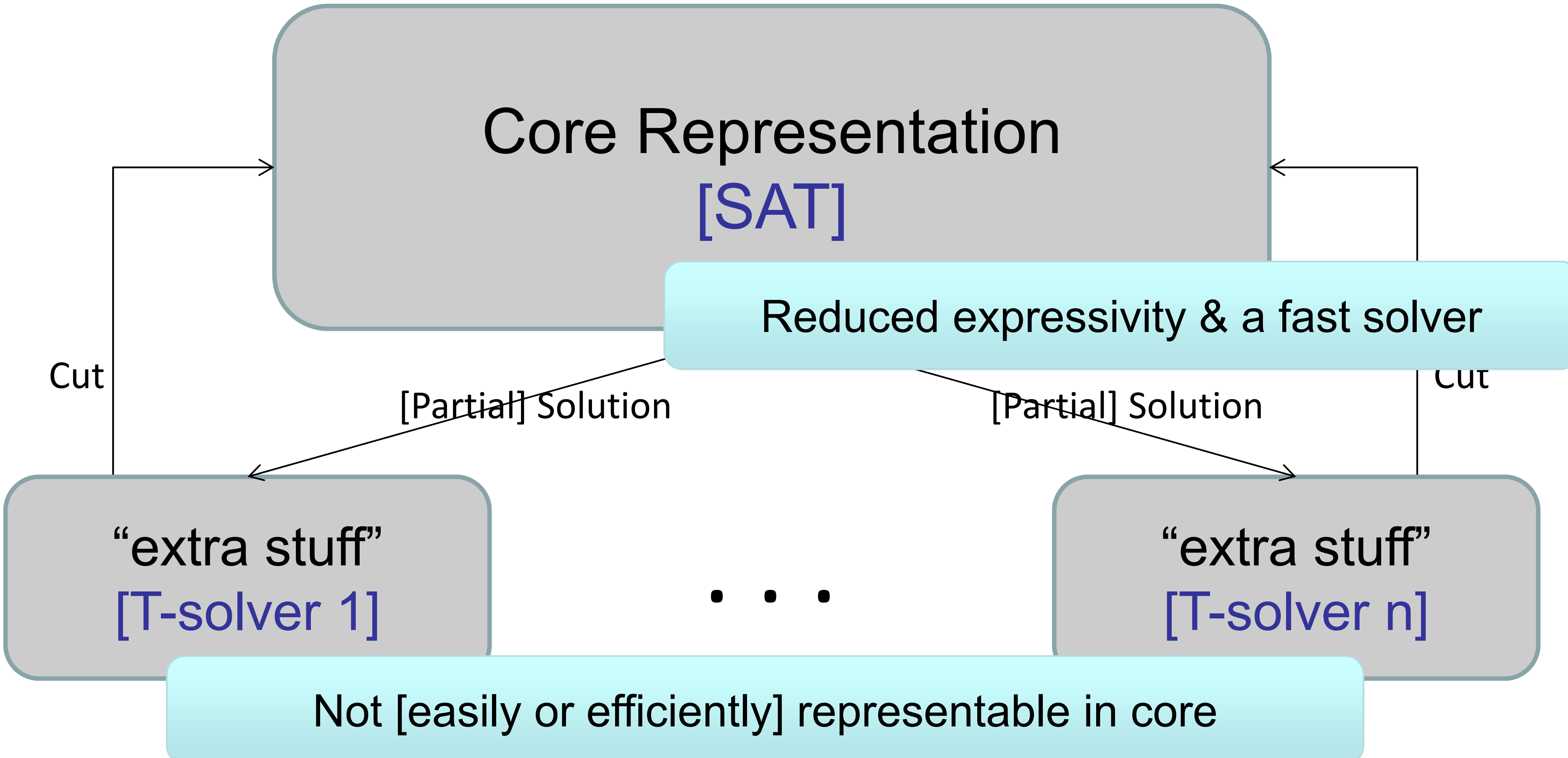

Core Representation
Cut
Solution
Solution
Cut
Subproblem 1
. . .
Subproblem n


Core Representation
Reduced expressivity & a fast solver
Cut
Solution
Solution
Cut
Subproblem 1
. . .
Subproblem n

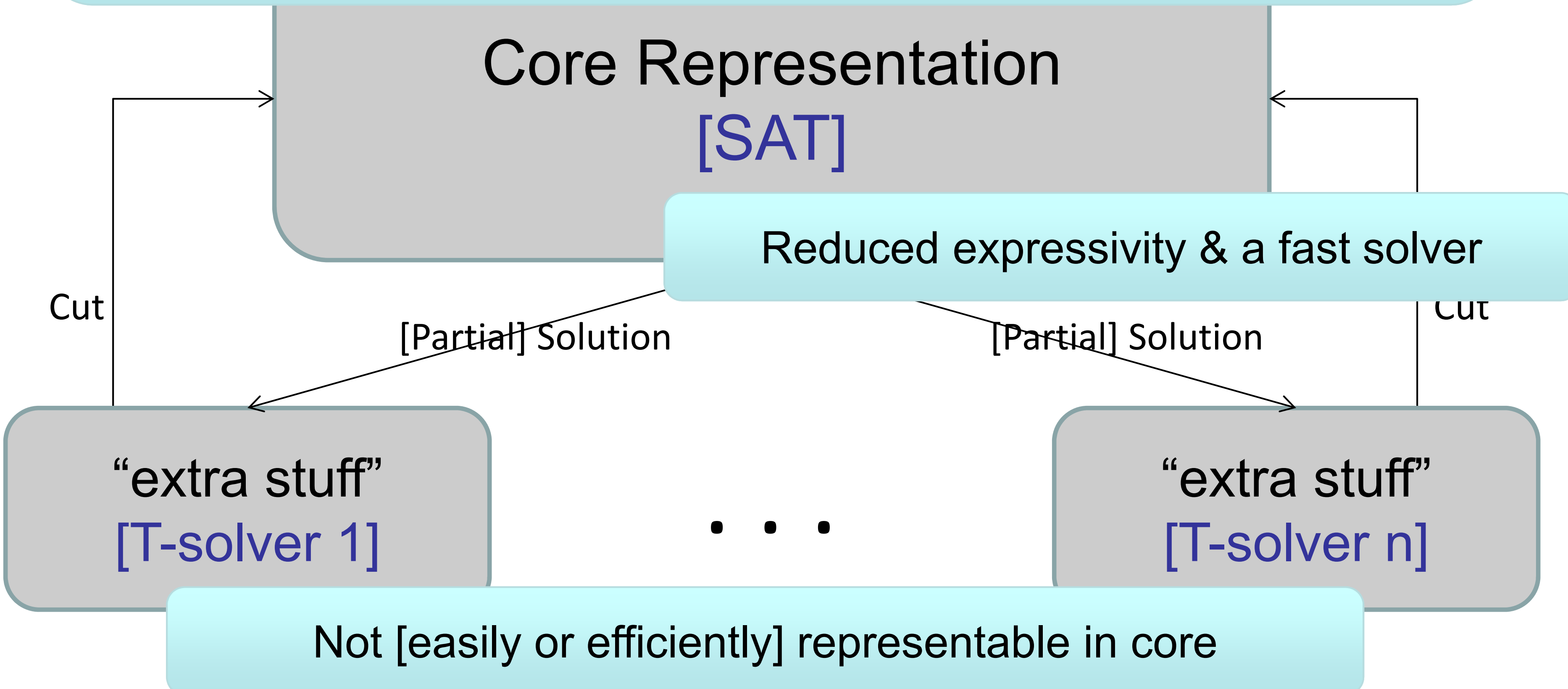

Core Representation
Reduced expressivity & a fast solver
Cut
Solution
Solution
Cut
“extra stuff”
. . .
“extra stuff”

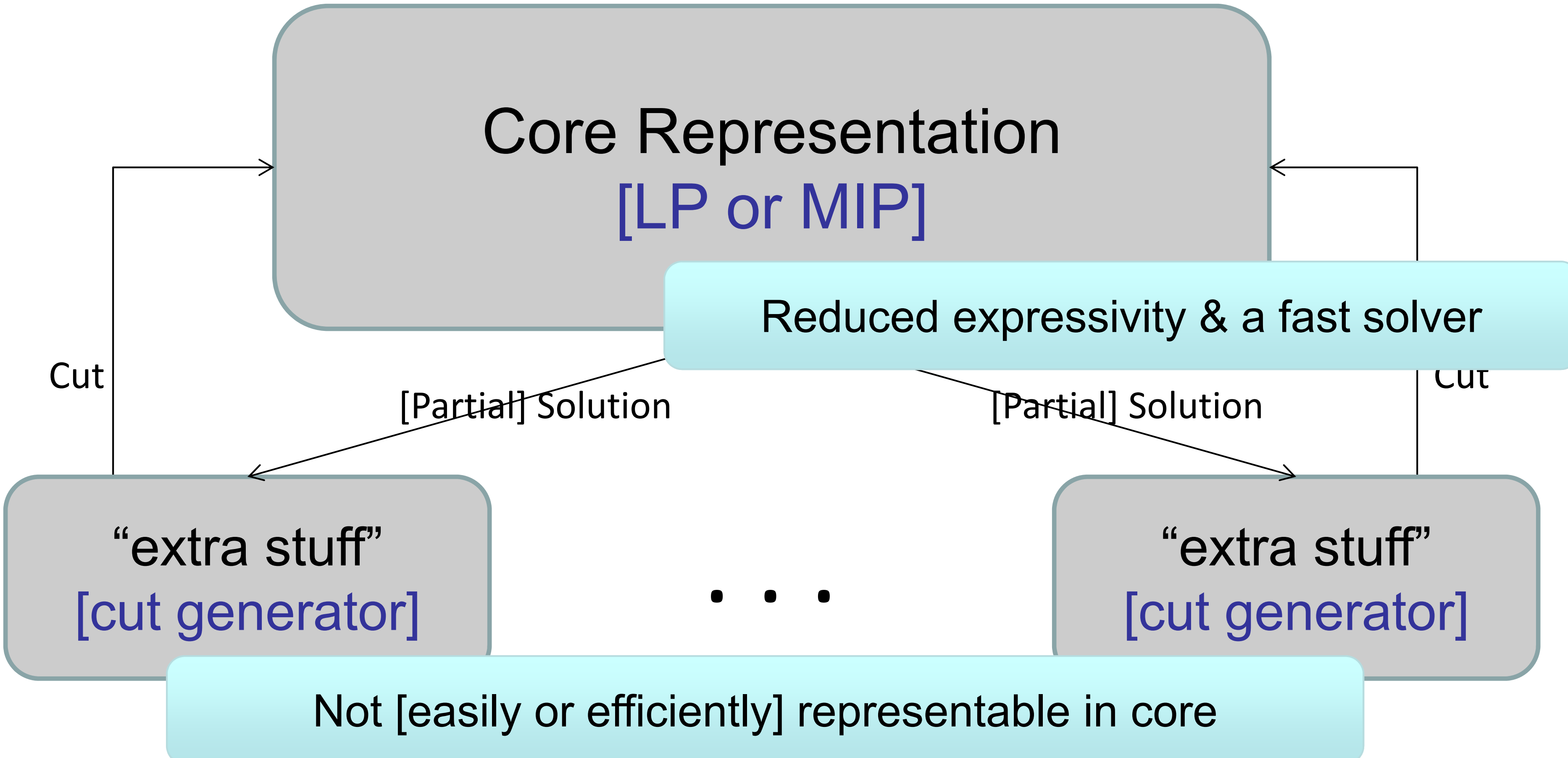

Core Representation
Reduced expressivity & a fast solver
Cut
Solution
Solution
Cut
“extra stuff”
. . .
“extra stuff”
Not [easily or efficiently] representable in core


Core Representation
Reduced expressivity & a fast solver
Cut
[Partial] Solution
[Partial] Solution
Cut
"extra stuff"
. . .
"extra stuff"
Not [easily or efficiently] representable in core


Core Representation
[SAT]
Reduced expressivity & a fast solver
Cut
[Partial] Solution
[Partial] Solution
Cut
"extra stuff"
[T-solver 1]
. . .
"extra stuff"
[T-solver n]
Not [easily or efficiently] representable in core

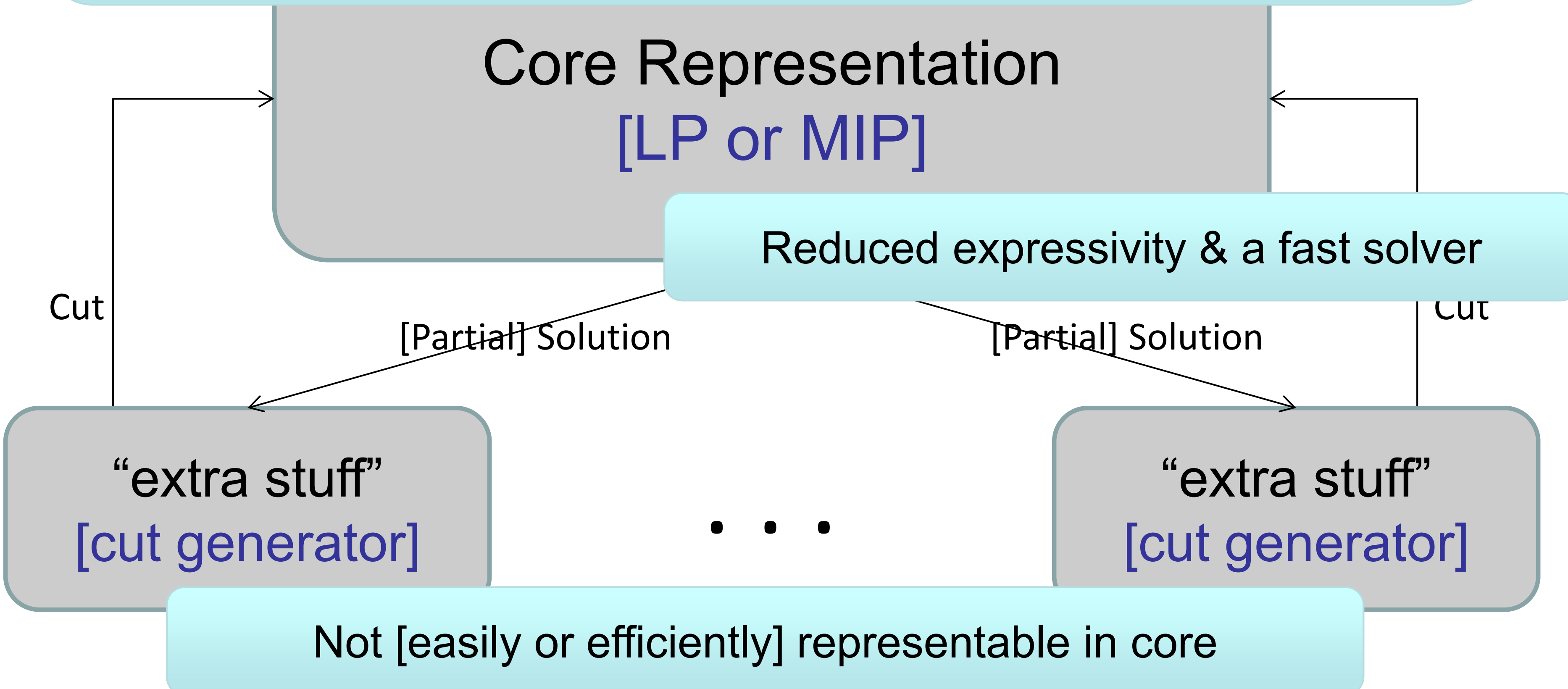
SAT Modulo Theory (SMT)

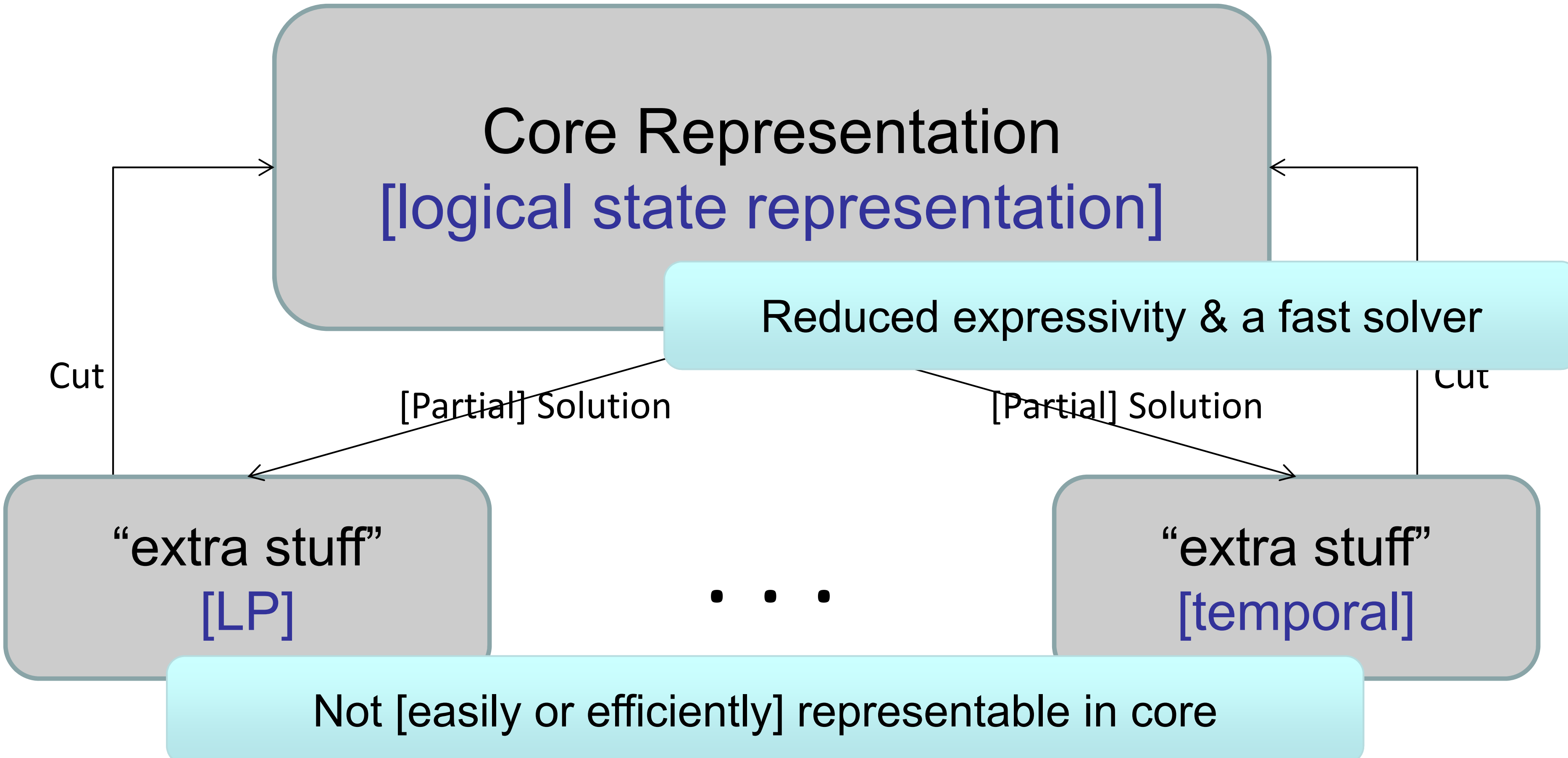




Core Representation
[LP or MIP]
Reduced expressivity & a fast solver
Cut
[Partial] Solution
[Partial] Solution
Cut
"extra stuff"
[cut generator]
. . .
"extra stuff"
[cut generator]
Not [easily or efficiently] representable in core

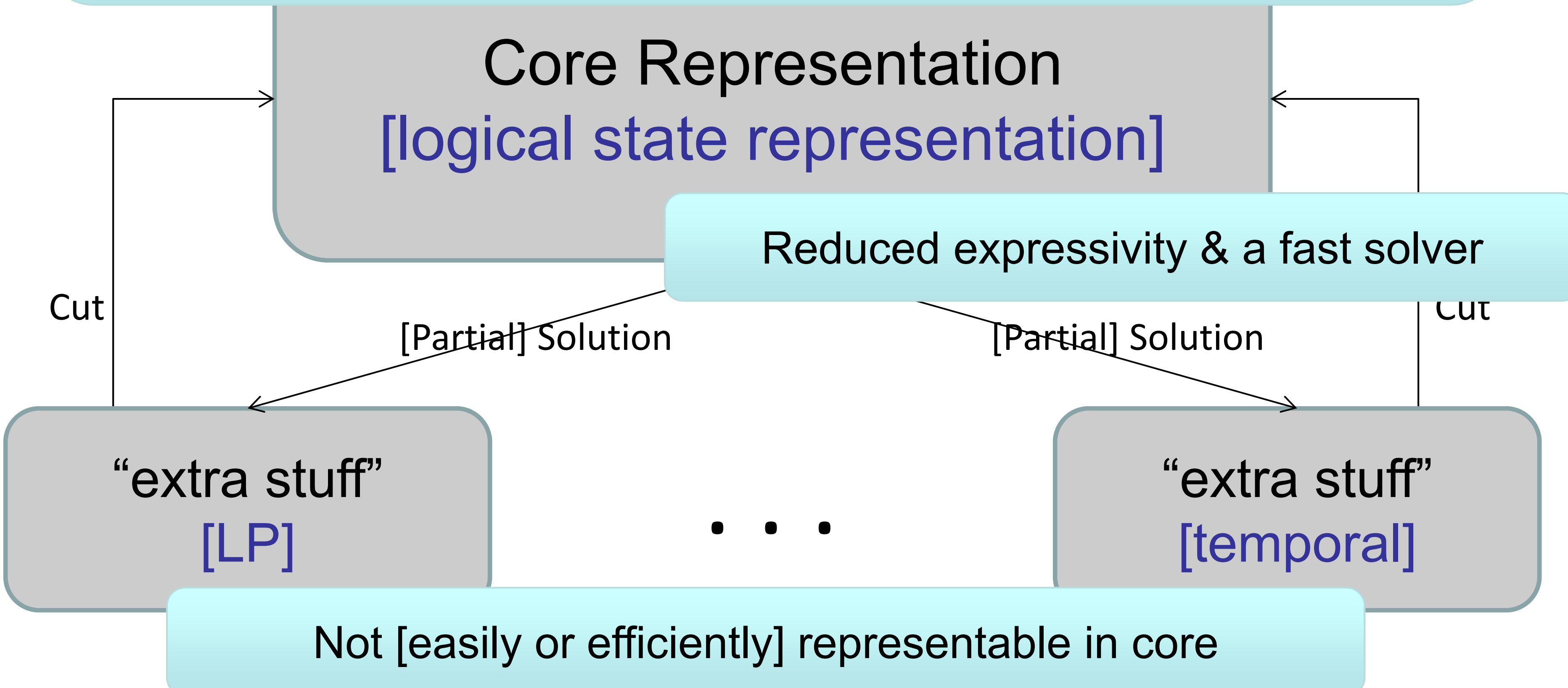
Branch-and-cut



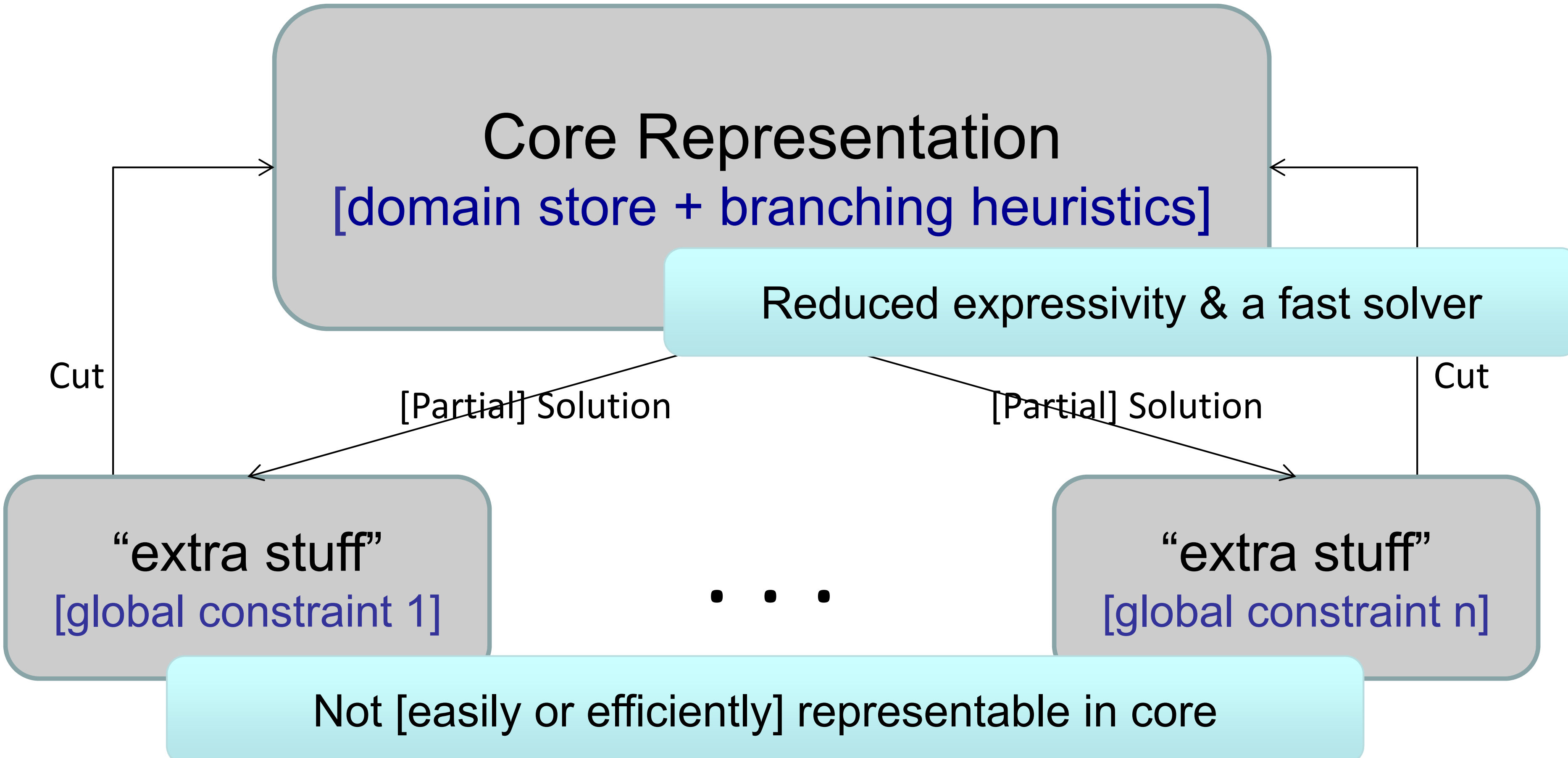


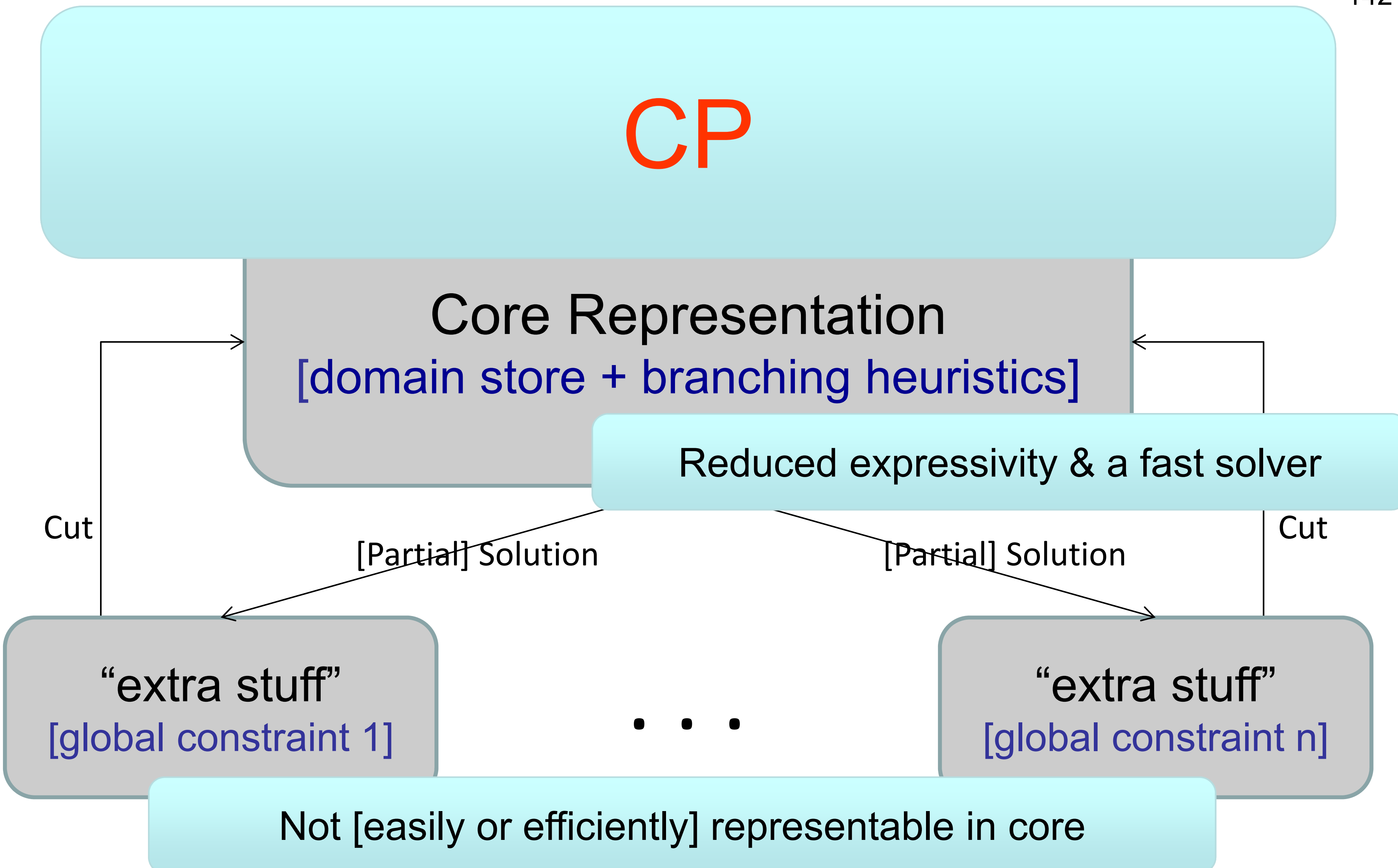
[Gregory et al. 2012] *ICAPS*, 65-73, 2012.

(AI) Planning Modulo Theory



[Gregory et al. 2012] *ICAPS*, 65-73, 2012.


Core Representation
[domain store + branching heuristics]
Reduced expressivity & a fast solver
Cut
[Partial] Solution
[Partial] Solution
Cut
"extra stuff"
[global constraint 1]
. . .
"extra stuff"
[global constraint n]
Not [easily or efficiently] representable in core


CP
Core Representation
[domain store + branching heuristics]
Reduced expressivity & a fast solver
Cut
[Partial] Solution
[Partial] Solution
Cut
"extra stuff"
[global constraint 1]
. . .
"extra stuff"
[global constraint n]
Not [easily or efficiently] representable in core

Thesis

- CP itself can be seen as an instance of this decomposition pattern
- But a sub-problem "solver" (i.e. a constraint) has been almost always consistency enforcement
- It is time to move beyond this narrow view of a constraint and really exploit the choice of a rich constraint representation

Things a Constraint Can Do

- Automatically detect independent sub-problems and solve them
  [Heinz, Ku, & B. 2013] *CPAIOR*, 12-27, 2013.
- Automated remodeling via dual presolving
  [Heinz, Schulz, & B. 2013] *Constraints*, **18**, 166-201, 2013.
- Provide heuristic information (solution counting)
  [Pesant et al. 2012] *JAIR*, **43**, 173-210, 2012.
- Generate clauses/explanations
  [Schutt et al. 2011] *Constraints*, **16**, 173-194, 2011.

Take Home Message I

- Decomposition (LBBD) is a valuable approach to solving hard combinatorial optimization problems
  - But it is non-trivial to use
  - Sub-problem relaxation and cuts critical
- Can it be used to integrate related problems currently solved separately?

Take Home Message II

- LBBD is a pattern of delayed constraint posting that can be seen in a number of techniques: SMT, B&Cut, and PMT
  - thinking of global constraints as such a sub-problem (and more than just an inference mechanism) is a promising direction

No zombies were optimized in the making of this presentation