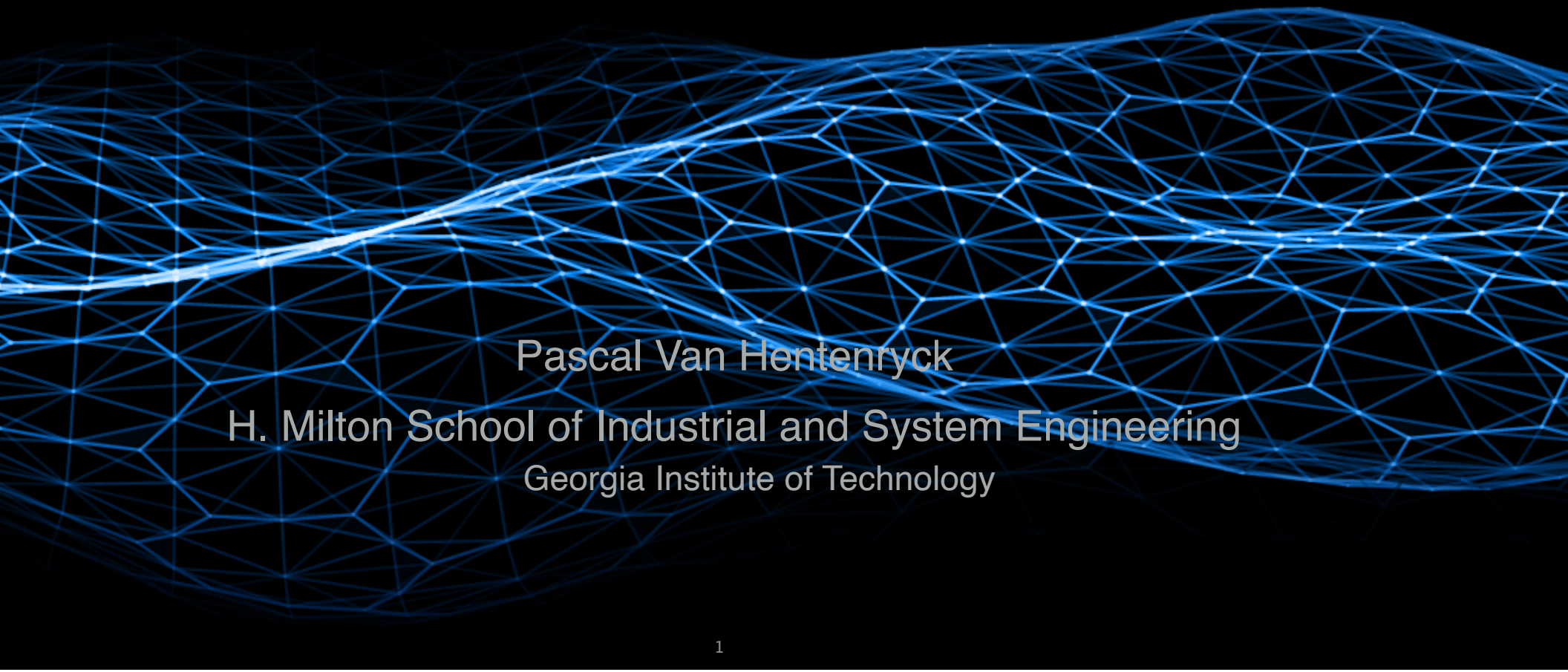
Constraint Programming for Scheduling



Pascal Van Hentenryck

H. Milton School of Industrial and System Engineering

Georgia Institute of Technology

Constraint Programming

AI Institute for Advances in Optimization

- IEEE Software 2010
  - "The application and importance of CP has grown remarkably in the past two decades"
- OR/MS August 2011
  - "A must-have tool for any O.R. Practitioner's toolkit"

Progress in Optimization

- "If you only knew optimization from 10 years ago, you probably don't have the techniques needed to solve real-world sport scheduling problems"
  - Mike Trick, Professor at CMU, 2008

- "The following do make a big difference (and are much more recent ideas)"
  - Complicated variables
  - Large neighborhood search
  - Constraint programming (ideally combined with integer programming).

Scheduling in the Last Decade

AI Institute for Advances in Optimization

Outline

AI Institute for Advances in Optimization

- Brief overview to constraint programming
- Scheduling with constraint programming
- Hybridizing constraint programming and MIP
- Learning-based constraint programming
- Learning Optimization Proxies

Constraint Programming

- A language for combinatorial optimization
  - Expressing substructures of applications
  - Expressing constraints at a high level of abstraction
  - Programming search
- A computational approach to optimization
  - Focus on feasibility
  - Focus on pruning infeasible solutions
    - Explicit representation of the search space


You Must Unlearn
What You Have Learned

Scheduling

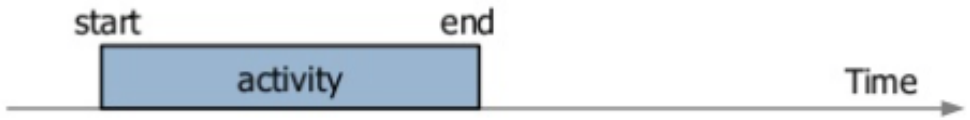
AI Institute for Advances in Optimization

- One of the killer applications for constraint programming
  - heavily used in industry
  - many MIP models face significant challenges
- Modeling layer
  - high-level concepts for scheduling
    - interval variables, global constraints, …
- Dedicated solving capabilities
  - sophisticated algorithms
  - lower bounds
  - branching strategies

What is an Interval Variable?

AI Institute for Advances in Optimization

- An object that has
  - a start variable
  - an end variable
  - a duration variable (possibly a constant)
- Ideal for scheduling

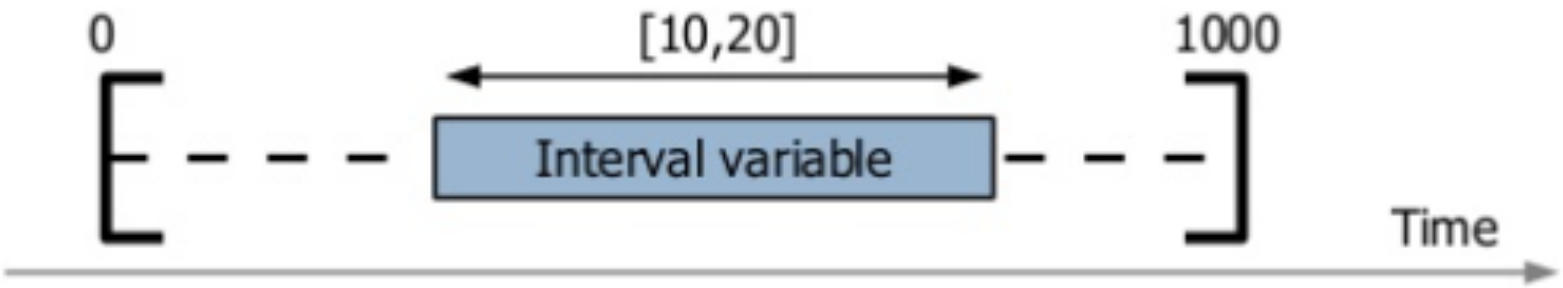


Interval Variable

AI Institute for Advances in Optimization

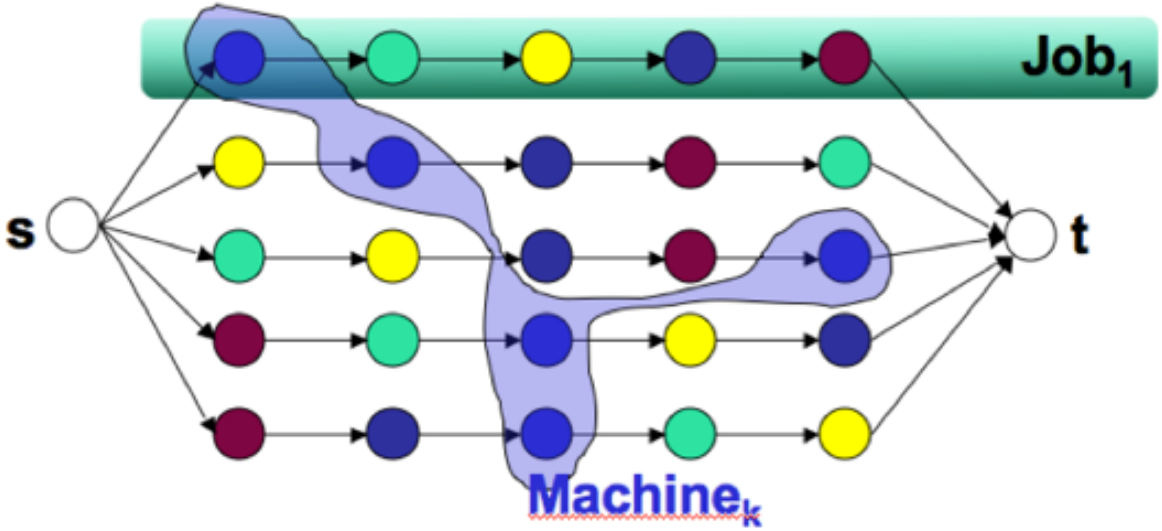
- Durations can be a variable as well

```
dvar interval x in 0..1000 size in 10..20;
```

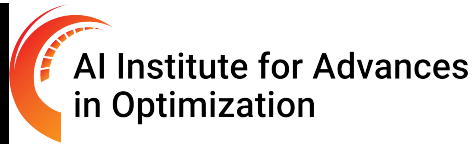


Jobshop Scheduling

AI Institute for Advances in Optimization



Jobshop Scheduling



- We are given a set of jobs
  - each job is a sequence of tasks
  - each
  - each
  - each
- We ar
  - each
- The g
  - the to

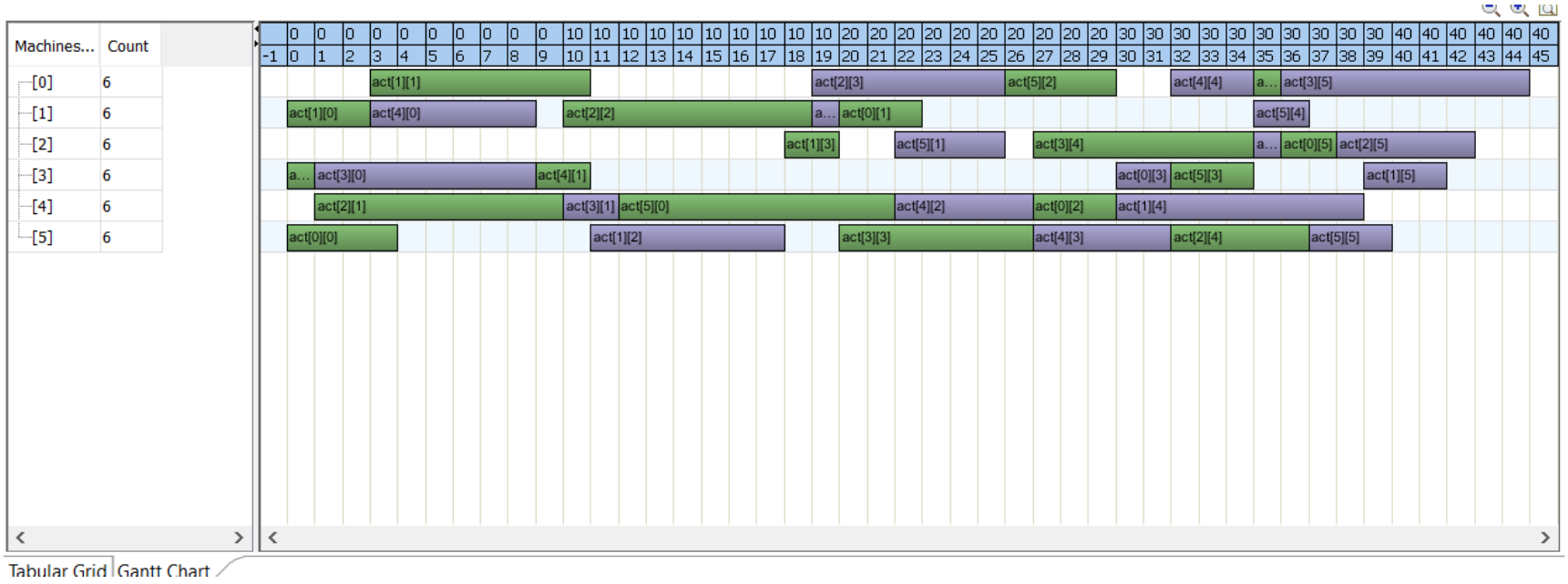
```
nbJobs = 6;
nbMachines = 6;

op = [
 [ <5,4>, <1,3>, <4,3>, <3,2>, <0,1>, <2,2> ],
 [ <1,3>, <0,8>, <5,7>, <2,2>, <4,9>, <3,3> ],
 [ <3,1>, <4,9>, <1,9>, <0,7>, <5,5>, <2,5> ],
 [ <3,8>, <4,2>, <1,1>, <5,7>, <2,8>, <0,9> ],
 [ <1,6>, <3,2>, <4,5>, <5,5>, <0,3>, <2,1> ],
 [ <4,10>, <2,4>, <0,4>, <3,3>, <1,2>, <5,3> ]
];
```

Jobshop Scheduling

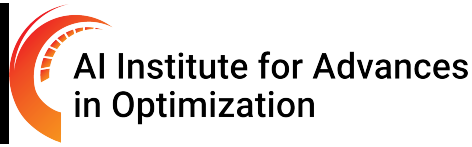

```
int nbJobs = ...;
int nbTasks = ...;
range Jobs = 0..nbJobs-1;
range Tasks = 0..nbtasks-1;
tuple Operation {int machine; int pt; };
Operation op[Jobs,Tasks] = ...;

dvar interval act[j in Jobs,o in Tasks] size op[j,o].pt;
dvar sequence machine[m in Machines] in
     all(j in Jobs,o in Tasks: op[j,o].machine == m) act[j,o];

minimize max(j in Jobs) endOf(act[j,nbTasks-1]);
subject to {
  forall (m in Machines)
    noOverlap(machine[m]);
  forall (j in Jobs, o in 0..nbTasks-2)
    endBeforeStart(act[j,o],act[j,o+1]);
}
```

Jobshop Scheduling

AI Institute for Advances in Optimization



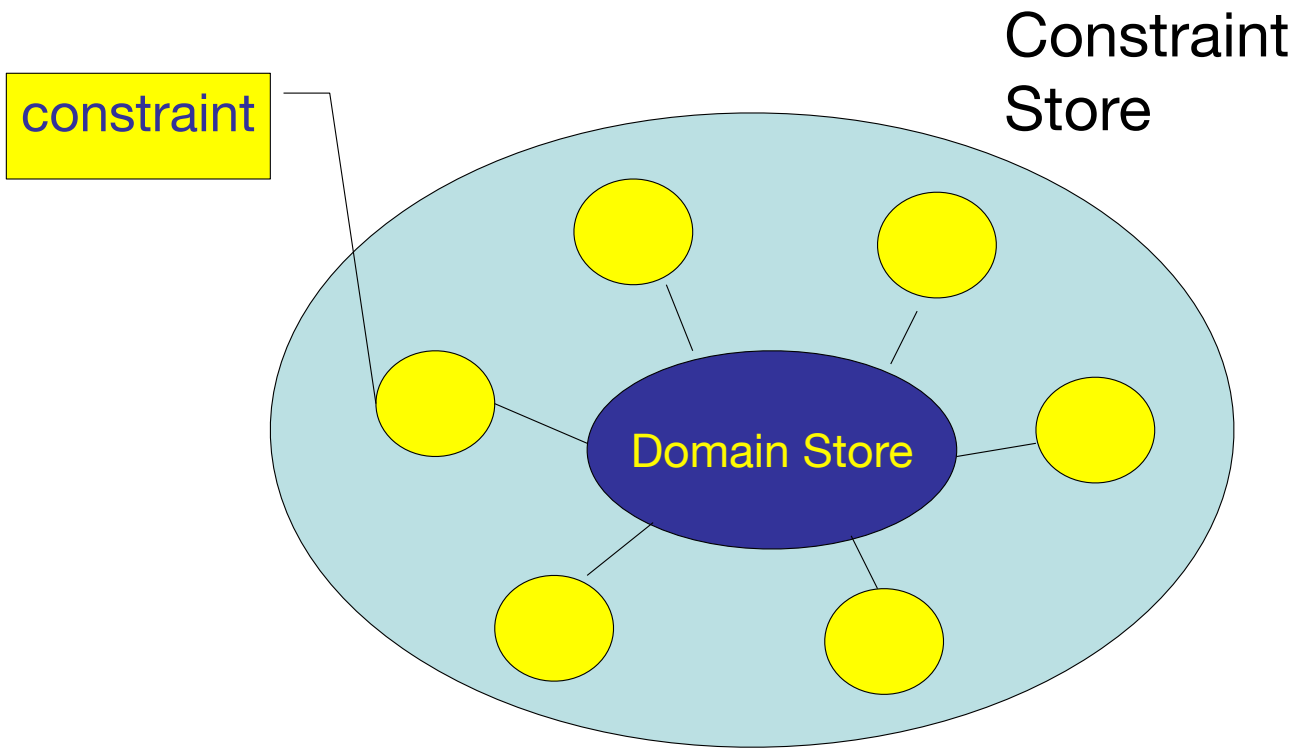
Computational Model of CP



- Branch and prune
  - Iterate two steps: pruning and branching
- Pruning: reduce the search space
  - Explicit representation of the search space
  - Each variable has its set of possible values
  - Use constraints to remove infeasible values
- Branching: decompose the problem into subproblems
  - Use feasibility information to determine how to branch

Constraint Programming For Scheduling

AI Institute for Advances in Optimization



Constraint Solving

AI Institute for Advances in Optimization

- What does a constraint do?
  - Feasibility Checking
    - Can the constraint be satisfied given the variable domains?
  - Pruning
    - remove values from the domains if they do not appear in any solution of the constraint
    - Holy grail: domain consistency
- How we solve these algorithmic problems, if fast enough, is irrelevant
    - dedicated to each constraint

Constraint Propagation

AI Institute for Advances in Optimization

```
dvar int x[1..5] in 1..4;
constraints {
   allDifferent(all(i in 1..4) x[i]);
   x[2] <= x[5];
   sum(i in 1..4) x[i] <= 9;
}
```

	x[1] =1	alldiff	x[2]≤ x[5]	x[5] >2	x[2] ≤ x[5]	alldiff	sum≤9	alldiff
x[1]	*1*	*1*	*1*	*1*	*1*	*1*	*1*	*1*
x[2]	*1..4*	*2..4*	*2..4*	*2..4*	*2*	*2*	*2*	*2*
x[3]	*1..4*	*2..4*	*2..4*	*2..4*	*2..4*	*3..4*	*3*	✖
x[4]	*1..4*	*2..4*	*2..4*	*2..4*	*2..4*	*3..4*	*3*	✖
x[5]	*1..4*	*1..4*	*2..4*	*3..4*	*2*	*2*	*2*	*2*

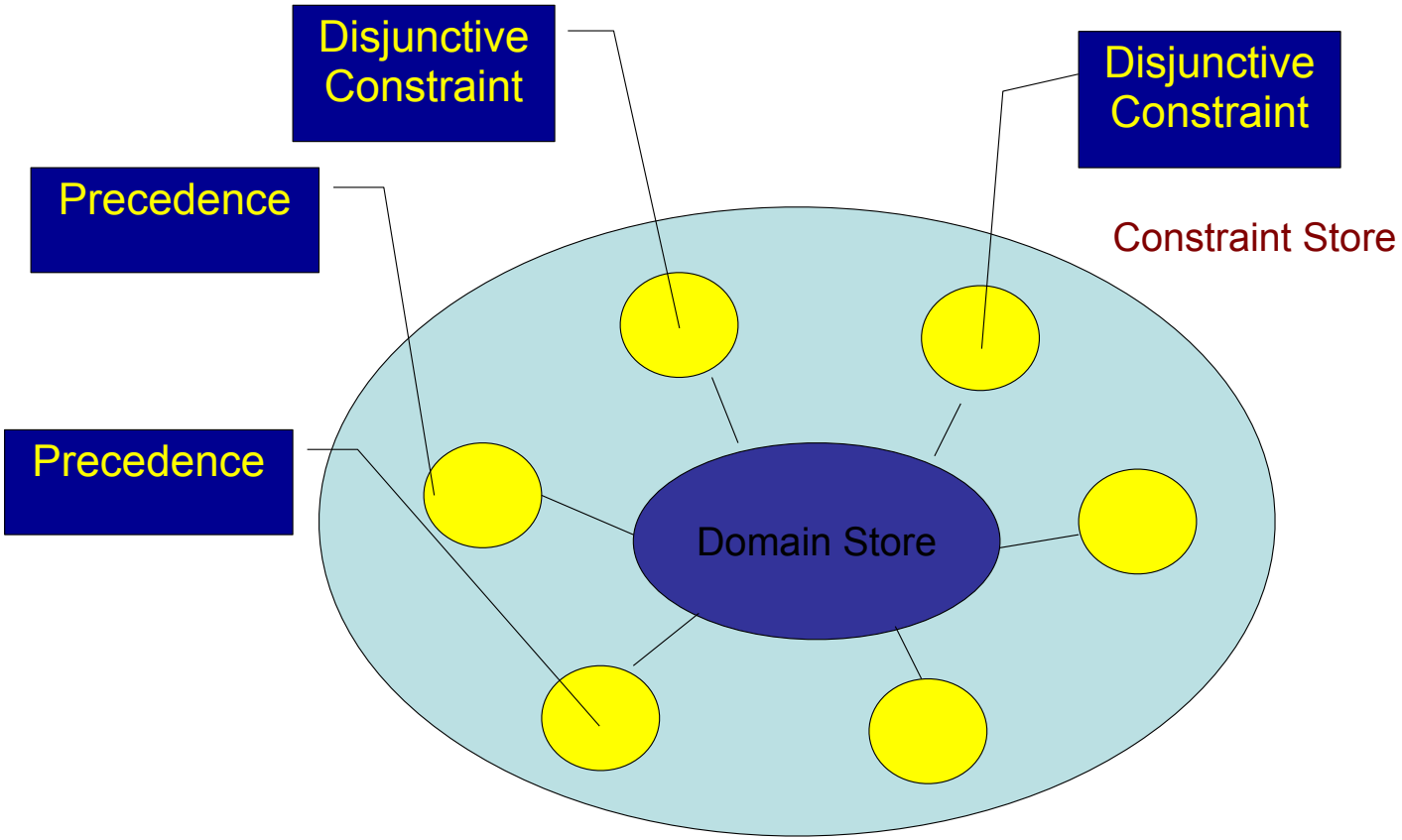
Jobshop Scheduling

AI Institute for Advances in Optimization

```
int nbJobs = ...;
int nbTasks = ...;
range Jobs = 0..nbJobs-1;
range Tasks = 0..nbtasks-1;
tuple Operation {int machine; int pt; };
Operation op[Jobs,Tasks] = ...;

dvar interval act[j in Jobs,o in Tasks] size op[j,o].pt;
dvar sequence machine[m in Machines] in
     all(j in Jobs,o in Tasks: op[j,o].machine == m) act[j,o];

minimize max(j in Jobs) endOf(act[j,nbTasks-1]);
subject to {
  forall (m in Machines)
    noOverlap(machine[m]);
  forall (j in Jobs, o in 0..nbTasks-2)
    endBeforeStart(act[j,o],act[j,o+1]);
}
```

Jobshop Scheduling

AI Institute for Advances in Optimization



No-Overlap Constraints

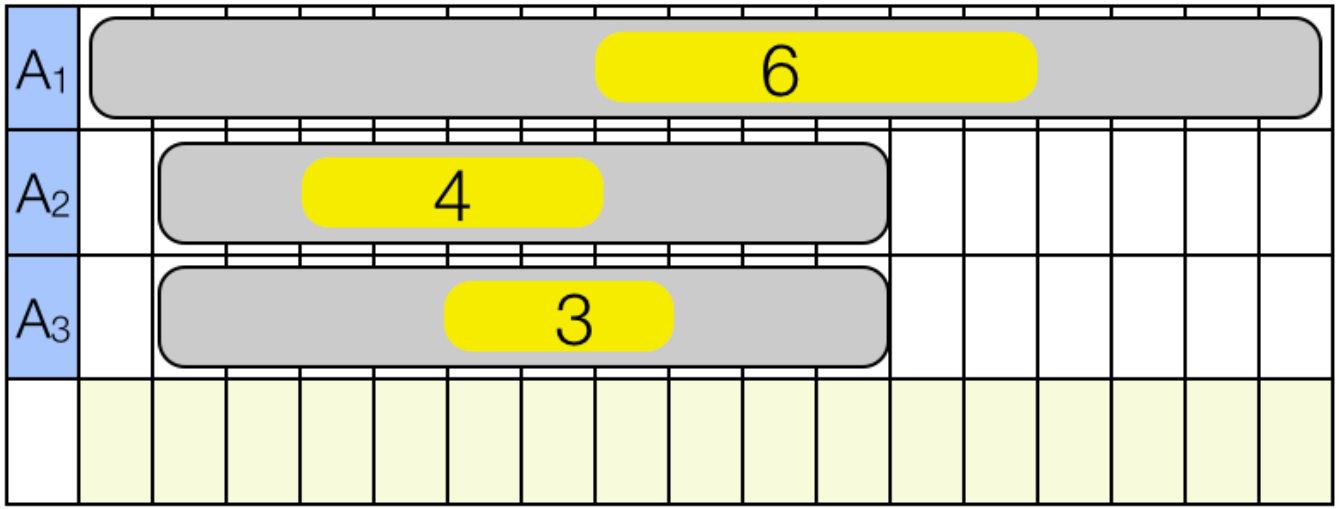
AI Institute for Advances in Optimization

- Given a set of tasks
  - each task has a release date, due date, and duration
- Determine if there exist start dates for the tasks
  - so that no two tasks overlap in time
- One-Machine Problem

No-Overlap Constraints

AI Institute for Advances in Optimization

- One-Machine Feasibility

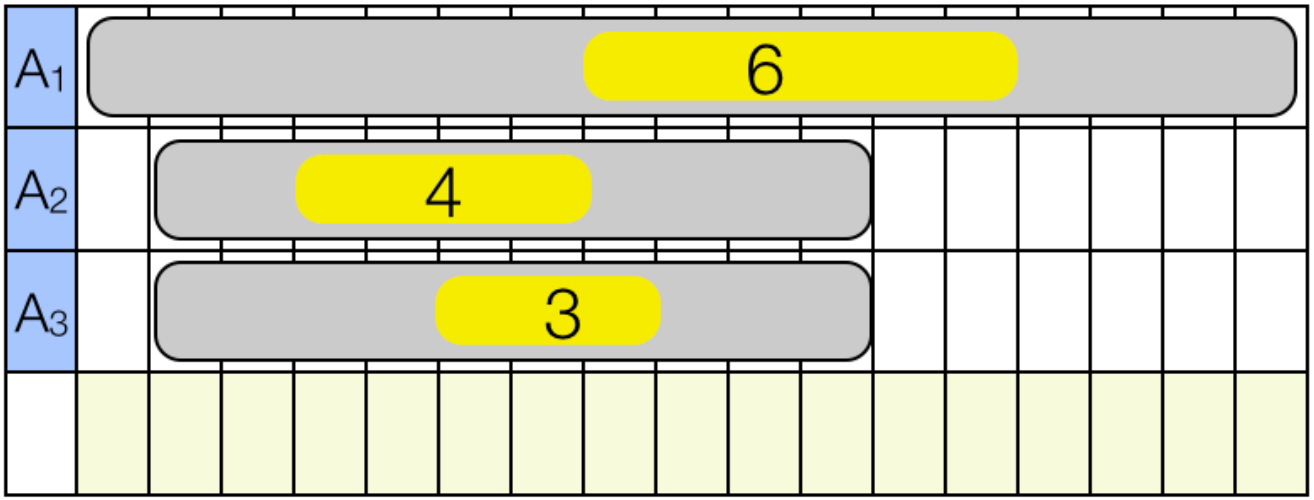


One-Machine Feasibility is NP-Hard

No-Overlap Constraints

AI Institute for Advances in Optimization

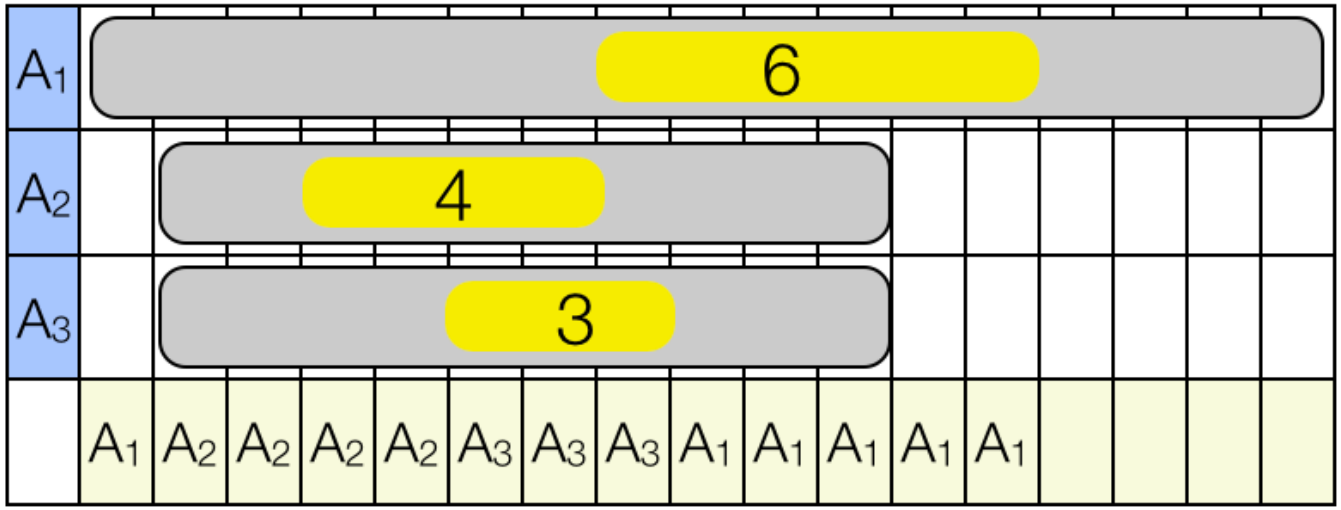
- Relax! One-Machine Preemptive Feasibility



- One-Machine Preemptive Feasibility can be computed in O(n log n) time.

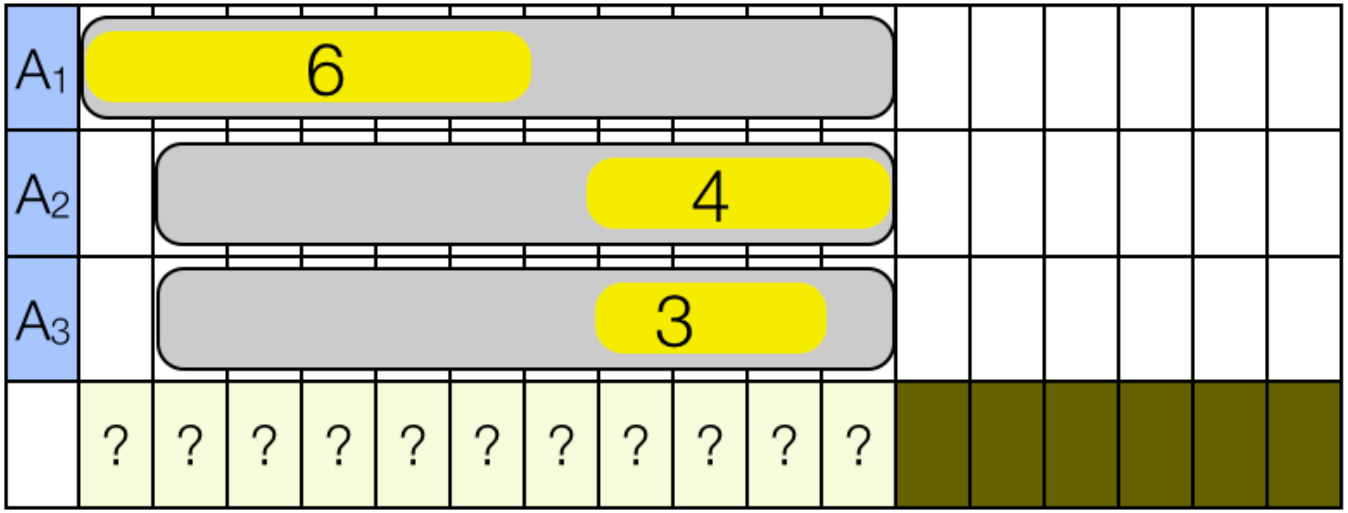
Pruning: Edge Finder Rules

- Pruning: Must A_1 start after A_2 and A_3?



Pruning: Edge Finder Rules

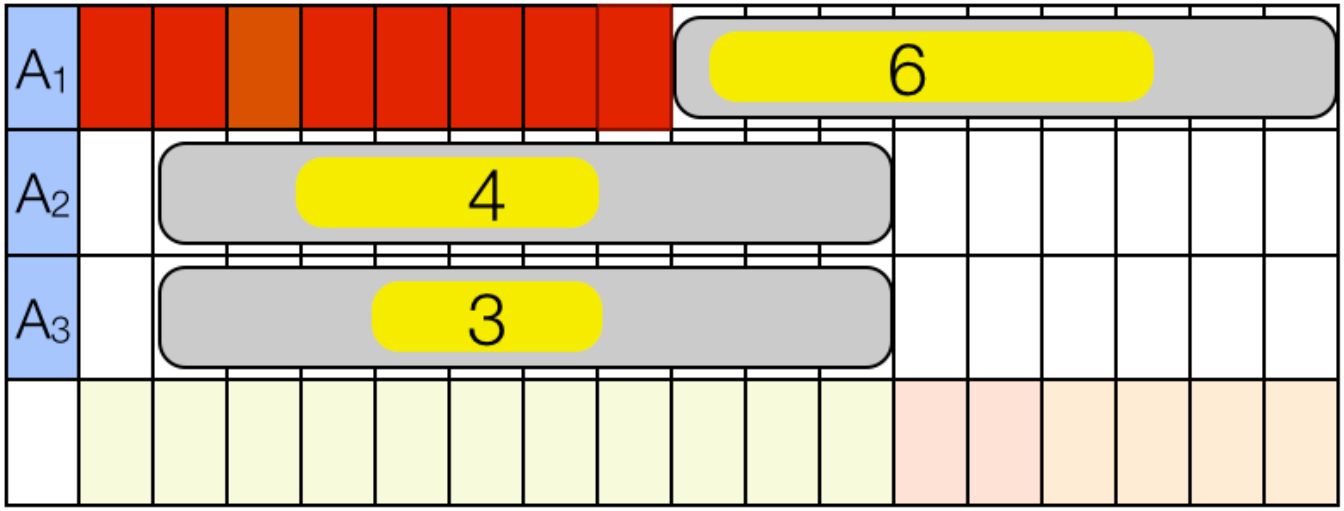
- Pruning: Must A_1 start after A_2 and A_3?



Pruning: Edge Finder Rules

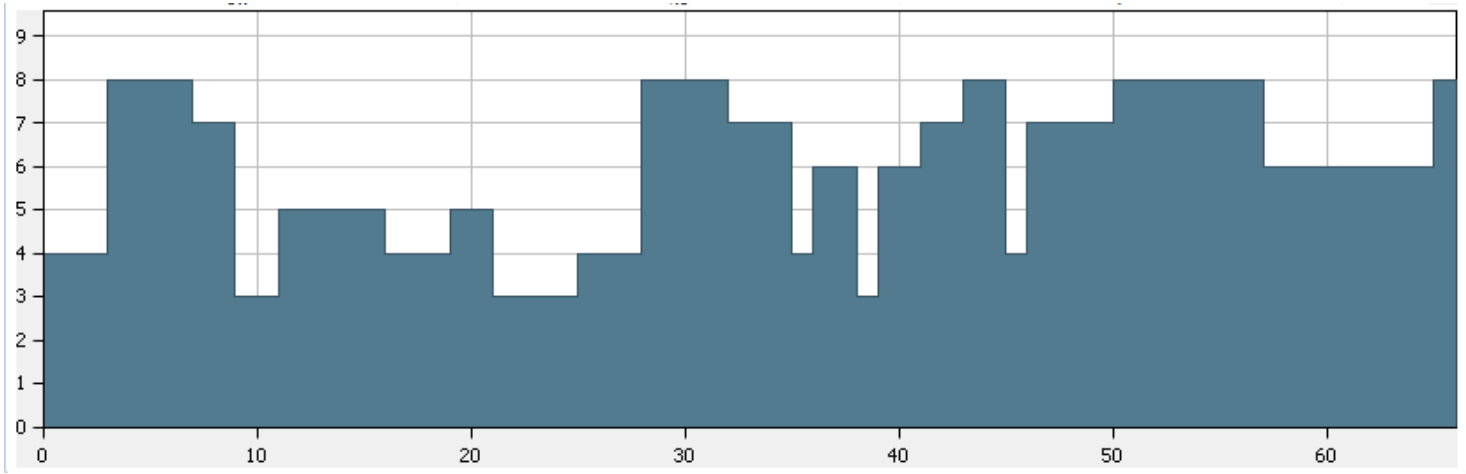
AI Institute for Advances in Optimization

- Pruning: A_1 must start after A_2 and A_3



Cumulative Constraints

AI Institute for Advances in Optimization

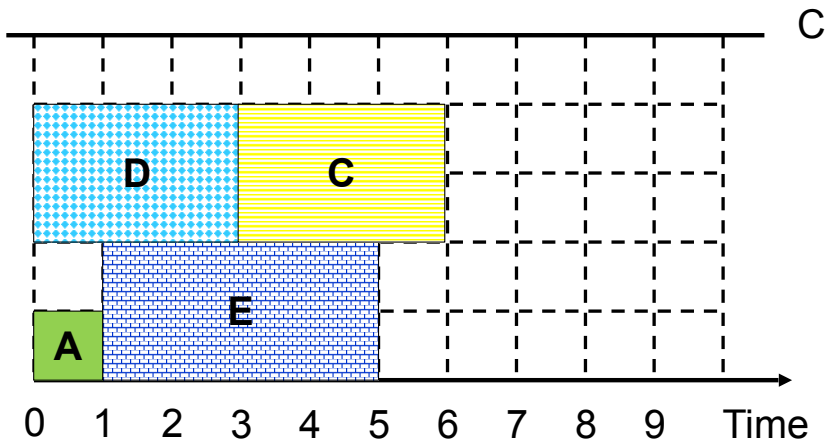


Cumulative Constraints

AI Institute for Advances in Optimization

- Given a task set T and a resource with a capacity of C then an assignment S of the start times is a solution iff

$$\forall t \in [t, t_{\max}): \sum_{i \in T: S(i) \le t < S(i)+p_i} c_i \le C$$



Modern CP Solvers

AI Institute for Advances in Optimization

Key Modeling Features

AI Institute for Advances in Optimization

- Variables
  - interval variables
  - optional interval variables
  - sequences
- Constraints
  - no-overlap, cumulative, state constraints
  - sequence dependent transition times and costs
  - alternative constraints
- Other features
  - calendars
  - …

Key Features of the Solver

AI Institute for Advances in Optimization

- Lower bounds
  - temporal and linear relaxations
- Primal solutions
  - large neighborhood search
  - restarts
- Search no-goods
  - never explore the same subtree twice
- primal and dual threads
  - dedicated search methods for proving optimality

Outline

AI Institute for Advances in Optimization

- Brief Overview to constraint programming
- Scheduling with constraint programming
- Hybridizing constraint programming and MIP
- Learning-based constraint programming
- Learning Optimization Proxies

AI Institute for Advances in Optimization

European Journal of Operational Research 286 (2020) 945–962

Contents lists available at ScienceDirect

European Journal of Operational Research

journal homepage: www.elsevier.com/locate/ejor

Production, Manufacturing, Transportation and Logistics

Constraint programming models for integrated container terminal operations☆

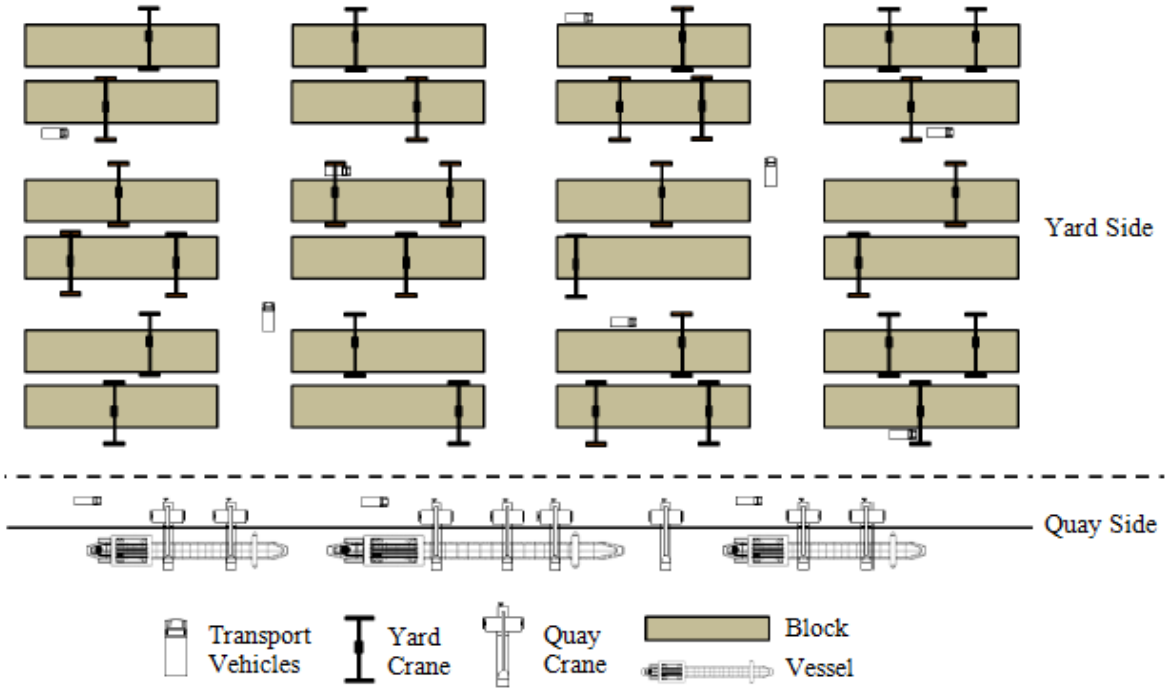
Damla Kizilay[a,*], Pascal Van Hentenryck[b], Deniz T. Eliiyi[c]

[a] Department of Industrial Engineering, Izmir Democracy University, Izmir 35140, Turkey
[b] Milton Stewart School of Industrial and Systems Engineering, Georgia Institute of Technology, USA
[c] Department of Industrial Engineering, Izmir Bakircay University, Izmir 35665, Turkey

Container Terminals

AI Institute for Advances in Optimization

General Picture of the Container Terminal



Quay Cranes

AI Institute for Advances in Optimization

Yard Cranes

AI Institute for Advances in Optimization

Yard Cranes

AI Institute for Advances in Optimization

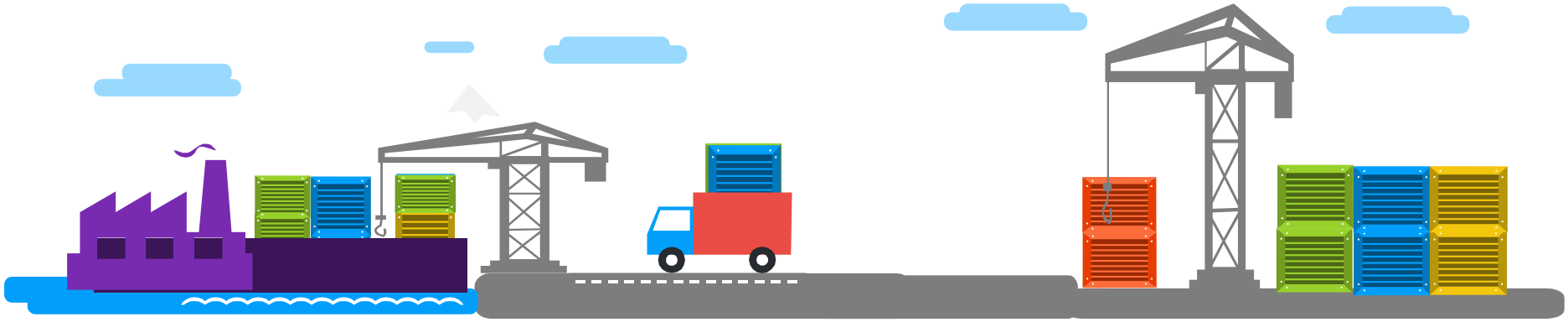
Yard Trucks

AI Institute for Advances in Optimization

Container Terminals

AI Institute for Advances in Optimization

General Operations in Container Terminal



Berth Allocation Problem (BAP): When and in which part of the port the coming vessel should berth. Vessel Stowage Plan has known.

BAP

Quay Crane Assignment and Scheduling Problem (QCASP): When loading and discharging operations should start by which QC.

QCASP

Yard Truck Dispatching Problem (YTDP): When containers should be transported between quay and yard by which YT.

TDP

Yard Crane Assignment and Scheduling Problem: Storage location and stacking position of containers should be determined.

YCASP

Yard Location Assignment and Stacking Problem: Storage location and stacking position of containers should be determined.

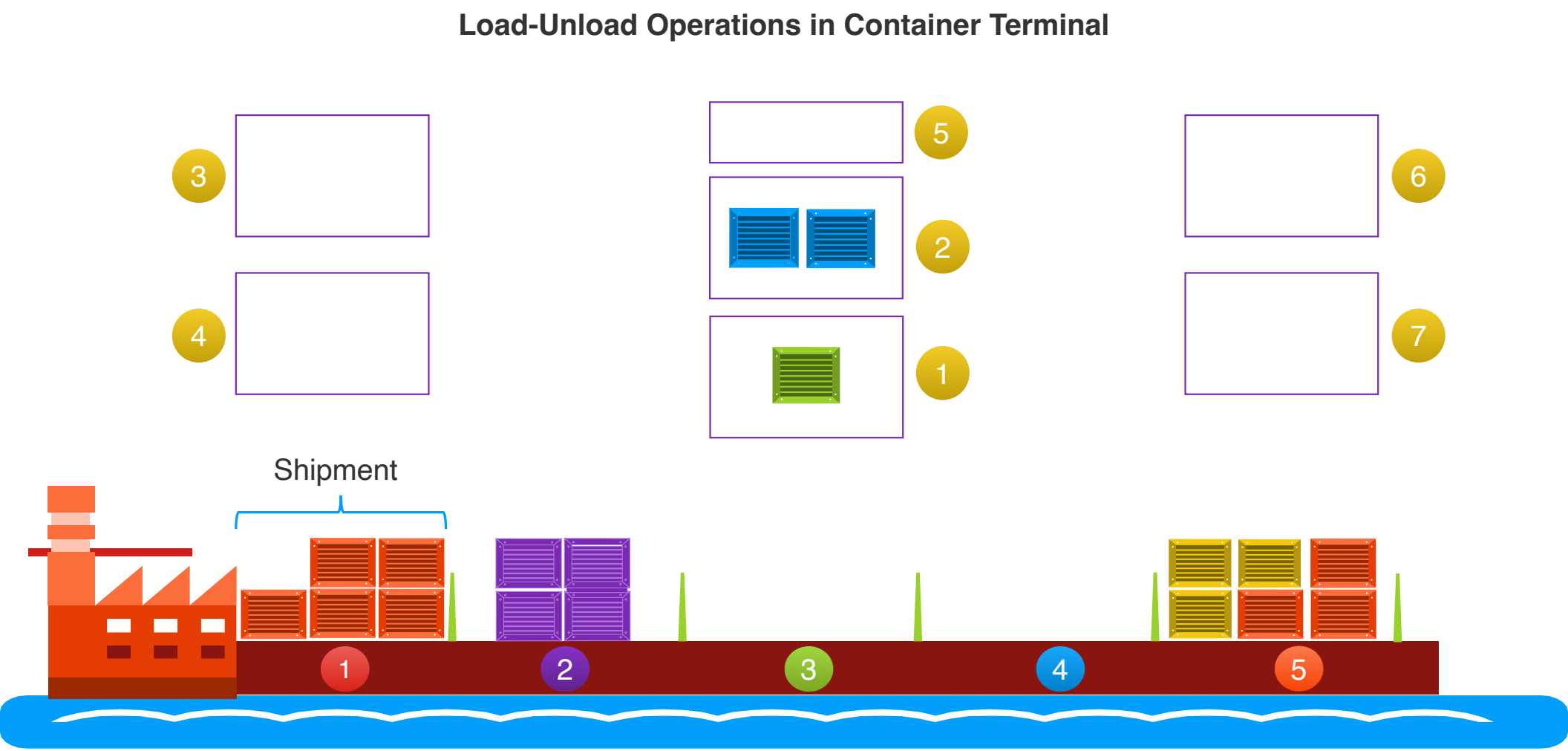
Stacking

Container Terminal in Izmir

- Berth allocation factors:
  - Priorities between customers
  - Berthing privileges of certain vessels at specific ports
  - Vessel sizes
  - Depth of water

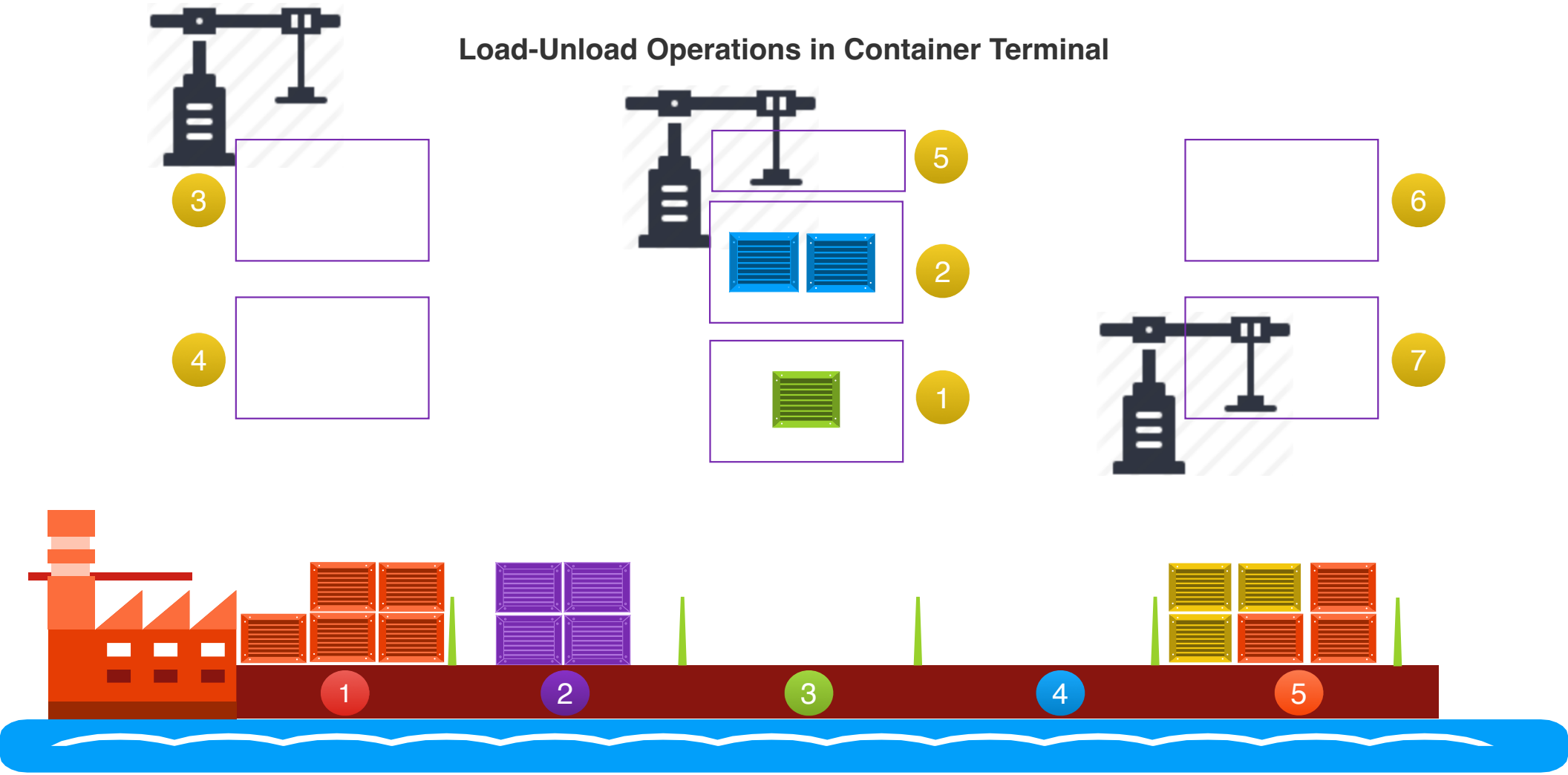
Problem Definition

AI Institute for Advances in Optimization



Problem Definition

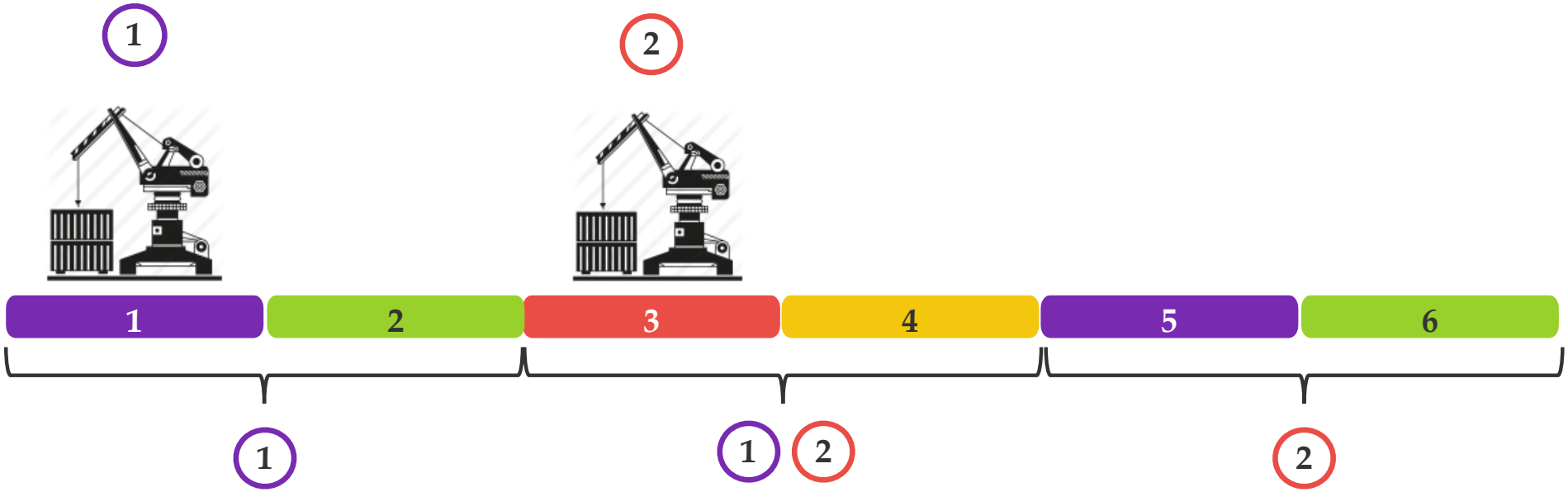
AI Institute for Advances in Optimization



Problem Definition

AI Institute for Advances in Optimization

QC Eligibility



Overview

AI Institute for Advances in Optimization

qc_i : Interval variable for the QC handling of shipment i
yc_i : Interval variable for the YC handling of shipment i
$aqc_{i,j}$: Optional interval variable for shipment i on QC j with duration between Q_i and M
$ayc_{i,k}$: Optional interval variable for shipment i on YC at yard location k with duration between Y_i and M located at yard location k on YT j with duration between 0 and M
qcs_j : Sequence variable for QC j over $\{aqc_{i,j} \mid i \in C\}$
ycs_j : Sequence variable for YC j over $\{ayc_{i,k} \mid i \in C \wedge YC(k) = j\}$
$interfere_{i,v,j,w}$: Sequence variable over $\{aqc_{i,v}, aqc_{j,w}\}$

Listing 1. The simplified CP model: variables.

Objective

$$\min \sum_{s \in S} w_s \left(max\left(\max_{i \in C_u^s} \text{ENDOF}\left(yc_i\right), \max_{j \in C_l^s} \text{ENDOF}\left(qc_j\right)\right)\right) \quad (01)$$

Constraints

$$\text{ALTERNATIVE}\left(qc_i, \text{all}\left(q \text{ in } QC\left(i\right)\right) aqc_{i,q}\right) \quad \forall i \in C \quad (02)$$

$$\text{ALTERNATIVE}\left(yc_i, \text{all}\left(k \text{ in } L_u\right) ayc_{i,k}\right) \quad \forall i \in C_u \quad (03)$$

$$\sum_{i \in C_u} \left(CN_i * \text{PRESENCEOF}\left(ayc_{i,k}\right)\right) \leq Cap_k \quad \forall k \in L_u \quad (04)$$

$$\text{PRESENCEOF}\left(ayc_{i,l_i}\right) = 1 \quad \forall i \in C_l \quad (05)$$

$$\text{STARTBEFORESTART}\left(aqc_{i,n}, ayc_{i,k}, Qt_i + tt_k\right) \forall i \in C_u, k \in L_u, n \in qc_i \quad (06)$$

$$\text{STARTBEFORESTART}\left(ayc_{i,l_i}, aqc_{i,n}, Yt_i + tyt_i\right) \quad \forall i \in C_l, n \in qc_i \quad (07)$$

$$\text{STARTBEFORESTART}\left(aqc_{i,m}, aqc_{j,n}\right) \quad \forall i, j \in P, m \in qc_i, n \in qc_j \quad (08)$$

$$\text{NOOVERLAP}\left(qcs_m, eqc_{i,j}\right) \quad \forall m \in QC \quad (09)$$

$$\text{NOOVERLAP}\left(ycs_m, eyc_{i,j}\right) \quad \forall m \in YC \quad (10)$$

$$\text{NOOVERLAP}\left(interfere_{i,v,j,w}, \Delta_{i,j}^{v,w}\right) \quad \forall i, j \in C, v \in qc_i, w \in qc_j, \Delta_{i,j}^{v,w} > 0 \quad (11)$$

Listing 2. The simplified CP Model: Objective and Constraints.

Decision Variables

AI Institute for Advances in Optimization

Variables

qc_i : Interval variable for the QC handling of shipment i

yc_i : Interval variable for the YC handling of shipment i

$aqc_{i,j}$: Optional interval variable for shipment i on QC j with duration Q_i

$ayc_{i,k}$: Optional interval variable for shipment i on YC at yard location k with duration Y_i

No Overlap

AI Institute for Advances in Optimization

qcs_j : Sequence variable for QC j over $\{aqc_{i,j} \mid i \in C\}$

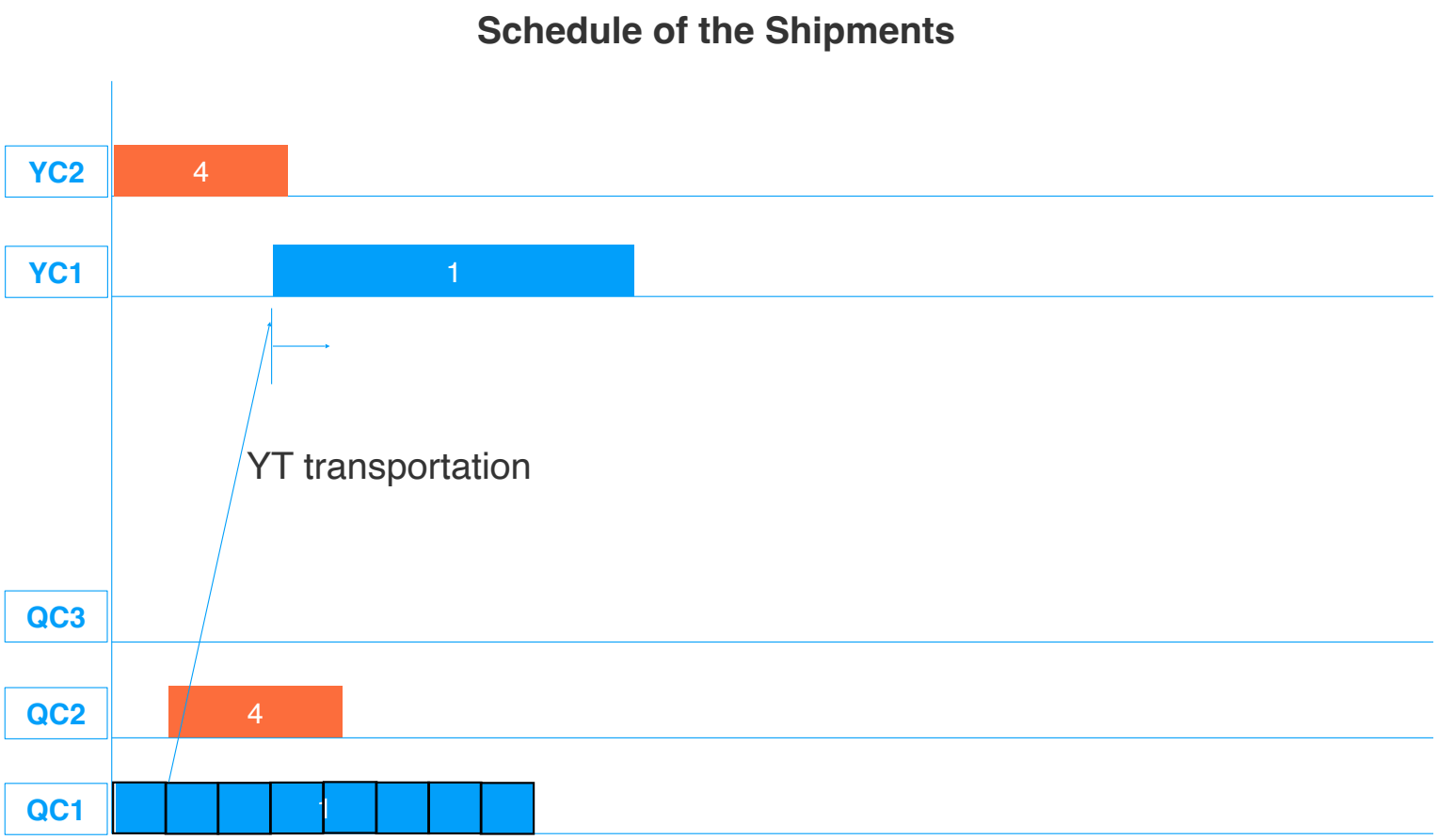
ycs_j : Sequence variable for YC j over $\{ayc_{i,k} \mid i \in C \wedge YC(k) = j\}$

$$\text{NOOVERLAP}\,(ycs_m, eyc_{i,j}) \quad \forall m \in YC$$

$$\text{NOOVERLAP}\,(qcs_m, eqc_{i,j}) \quad \forall m \in QC$$

Precedence Constraints

AI Institute for Advances in Optimization



Interferences

AI Institute for Advances in Optimization

- No overlap + distance

$interfere_{i,v,j,w}$: Sequence variable over $\{aqc_{i,v}, aqc_{j,w}\}$

$$\text{NOOVERLAP}\left(interfere_{i,v,j,w}, \Delta_{i,j}^{v,w}\right) \quad \forall i,j \in C, v \in qc_i, w \in qc_j, \Delta_{i,j}^{v,w} > 0$$

The Model

AI Institute for Advances in Optimization

- Minimizing the weighted completion times of each vessel

$$\min \sum_{s \in S} w_s \left(max \left(\max_{i \in C_u^s} \text{ENDOF}\left(yc_i\right), \max_{j \in C_l^s} \text{ENDOF}\left(qc_j\right) \right) \right)$$

Scheduling Yard Trucks

AI Institute for Advances in Optimization

yt_i : Interval variable for the YT handling of container i
$ayt_{i,k,j}$: Optional interval variable for container i transported to or initially located at yard location k on YT j with duration between 0 and M
yts_j : Sequence variable for YT j over $\{ayt_{i,j} \mid i \in K\}$
$contQC_i$: Interval variable for container i on QC with duration $Qtime_i$
$contYC_i$: Interval variable for container i on YC with duration $Ytime_i$

Listing 3. The CP Model for the IPCTP: Additional variables.

$$\text{ALTERNATIVE}\left(yt_i, \text{all}\left(v \text{ in } YT\right) ayt_{i,l_j,v}\right) \quad \forall i \in H_j, j \in C_l \tag{12}$$
$$\text{ALTERNATIVE}\left(yt_i, \text{all}\left(v \text{ in } YT, k \text{ in } L_u\right) ayt_{i,k,v}\right) \quad \forall i \in K_u \tag{13}$$
$$\sum_{v \in YT} \text{PRESENCEOF}\left(ayt_{i,l_j,v}\right) = 1 \quad \forall i \in H_j, j \in C_l \tag{14}$$
$$\sum_{v \in YT} \text{PRESENCEOF}\left(ayt_{i,k,v}\right) = \text{PRESENCEOF}\left(ayc_{j,k}\right) \quad \forall i \in H_j, j \in C_u, k \in L_u \tag{15}$$
$$\sum_{v \in YT} \text{SIZEOF}\left(ayt_{i,l_j,v}\right) \geq tyt_j \quad \forall i \in H_j, j \in C_l \tag{16}$$
$$\sum_{v \in YT} \text{SIZEOF}\left(ayt_{i,k,v}\right) \geq tt_k * \text{PRESENCEOF}\left(ayc_{j,k}\right) \forall k \in L_u, i \in H_j, j \in C_u \tag{17}$$
$$\text{STARTOF}\left(contQC_i\right) = \sum_{q \in QC} \text{STARTOF}\left(aqc_{j,q}\right) \quad i = \min\left(H_j\right) \tag{18}$$
$$\text{STARTOF}\left(contQC_j\right) \geq \text{ENDOF}\left(contQC_i\right) \quad \forall k \in C, i, j \in H_k | j = i + 1 \tag{19}$$
$$\text{STARTOF}\left(contYC_i\right) = \sum_{k \in L} \text{STARTOF}\left(ayc_{j,k}\right) \quad i = \min\left(H_j\right) \tag{20}$$
$$\text{STARTOF}\left(contYC_j\right) \geq \text{ENDOF}\left(contYC_i\right) \quad \forall k \in C, i, j \in H_k | j = i + 1 \tag{21}$$
$$\max_{i \in H_j} \text{ENDOF}\left(contQC_i\right) = \sum_{q \in QC} \text{ENDOF}\left(aqc_{j,q}\right) \quad \forall j \in C \tag{22}$$
$$\max_{i \in H_j} \text{ENDOF}\left(contYC_i\right) = \sum_{k \in L} \text{ENDOF}\left(ayc_{j,k}\right) \quad \forall j \in C \tag{23}$$
$$\text{ENDBEFORESTART}\left(contQC_i, ayt_{i,k,v}\right) \quad \forall i \in K_u, k \in L_u, v \in YT \tag{24}$$
$$\text{ENDBEFORESTART}\left(ayt_{i,k,v}, contYC_i\right) \quad \forall i \in K_u, k \in L_u, v \in YT \tag{25}$$
$$\text{ENDBEFORESTART}\left(contYC_i, ayt_{i,l_j,v}\right) \quad \forall i \in H_j, j \in C_l, v \in YT \tag{26}$$
$$\text{ENDBEFORESTART}\left(ayt_{i,l_j v}, contQC_i\right) \quad \forall i \in H_j, j \in C_l, v \in YT \tag{27}$$
$$\text{NOOVERLAP}\left(yts_m, eyt_{i,j}\right) \quad \forall m \in YT \tag{28}$$

Listing 4. The CP Model for the IPCTP: Additional Constraints.

Import-Export Rate 20%

AI Institute for Advances in Optimization

			MIP			CP		
U-L Rate	# of Bays	# of Shipment	Avg. Obj.	Avg. CPU	Avg. GAP%	Avg. Obj.	Avg. CPU	RPD%
2	4	5	293	0.40	0.00	293	0.05	0.00
		10	542$^{0/5}$	3600.05	0.36	542	9.41	0.00
		15	740$^{0/5}$	3601.68	0.69	740^{1}	734.91	0.00
		20	822$^{0/5}$	3601.39	0.75	820^{1}	788.81	0.00
		25	1018$^{2/3}$	3600.81	0.77	998^{2}	1791.34	0.00
	6	5	179	0.16	0.00	179	0.14	0.00
		10	360$^{0/1}$	1717.16	0.05	360	6.20	0.00
		15	544$^{0/5}$	3600.50	0.58	541	36.47	0.00
		20	661$^{1/4}$	3600.75	0.67	647	274.15	0.00
		25	NA	NA	NA	752	479.15	0.00
	8	5	427	0.16	0.00	427	0.40	0.00
		10	507	98.30	0.00	507	10.26	0.00
		15	844$^{0/5}$	3600.26	0.48	841	150.59	0.00
		20	967$^{2/3}$	3600.13	0.57	1052	542.60	0.00
		25	1466$^{4/1}$	3614.05	0.64	1320	1255.55	1.26

Import-Export Rate 20%

AI Institute for Advances in Optimization

			MIP			CP		
U-L Rate	# of Bays	# of Shipment	Avg. Obj.	Avg. CPU	Avg. GAP%	Avg. Obj.	Avg. CPU	RPD%
3	4	5	207	**0.40**	0.00	207	**0.21**	0.00
		10	426$^{0/4}$	**3297.84**	0.31	426	**6.80**	0.00
		15	643$^{0/5}$	**3605.24**	0.65	641	**79.30**	0.00
		20	865$^{0/5}$	**3600.71**	0.75	860^{1}	**897.30**	0.00
		25	824$^{3/2}$	**3601.27**	0.73	842^{2}	**1895.17**	0.00
	6	5	248	**0.16**	0.00	248	**0.15**	0.00
		10	401$^{0/1}$	**1449.26**	0.08	401	**8.70**	0.00
		15	482$^{0/1}$	**3600.39**	0.50	482	**539.06**	0.00
		20	626$^{1/4}$	**3600.50**	0.65	614^{1}	**778.93**	0.00
		25	988$^{2/3}$	**3600.22**	0.74	856^{4}	**2938.55**	0.00
	8	5	418	**0.22**	0.00	418	**1.27**	0.00
		10	655	**449.44**	0.00	655	**26.02**	0.00
		15	895$^{0/5}$	**3600.45**	0.47	882	**160.79**	0.00
		20	1272$^{1/4}$	**3611.86**	0.63	1113^{1}	**1524.57**	0.38
		25	NA	**NA**	NA	1382^{3}	**2681.45**	2.87

Outline

AI Institute for Advances in Optimization

- Brief Overview to constraint programming
- Scheduling with constraint programming
- Hybridizing constraint programming and MIP
- Learning-based constraint programming
- Learning Optimization Proxies

AI Institute for Advances in Optimization

ARTICLE

A branch-and-price-and-check model for the vehicle routing problem with location congestion

Authors: Edward Lam, Pascal Van Hentenryck Authors Info & Claims

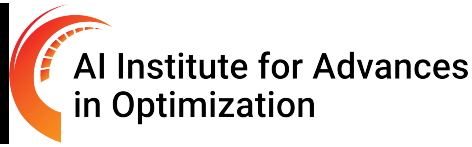
Constraints, Volume 21, Issue 3 • July 2016 • pp 394–412 • https://doi.org/10.1007/s10601-016-9241-2

CP and MIP

AI Institute for Advances in Optimization

- Trends in combinatorial optimization
  - logical Benders decomposition, and branch & check
  - generalizes Benders decomposition with combinatorial subproblems
- Branch and price and check
  - generalizes the idea to branch and price
- Beautiful synergies between
  - mathematical and constraint programming

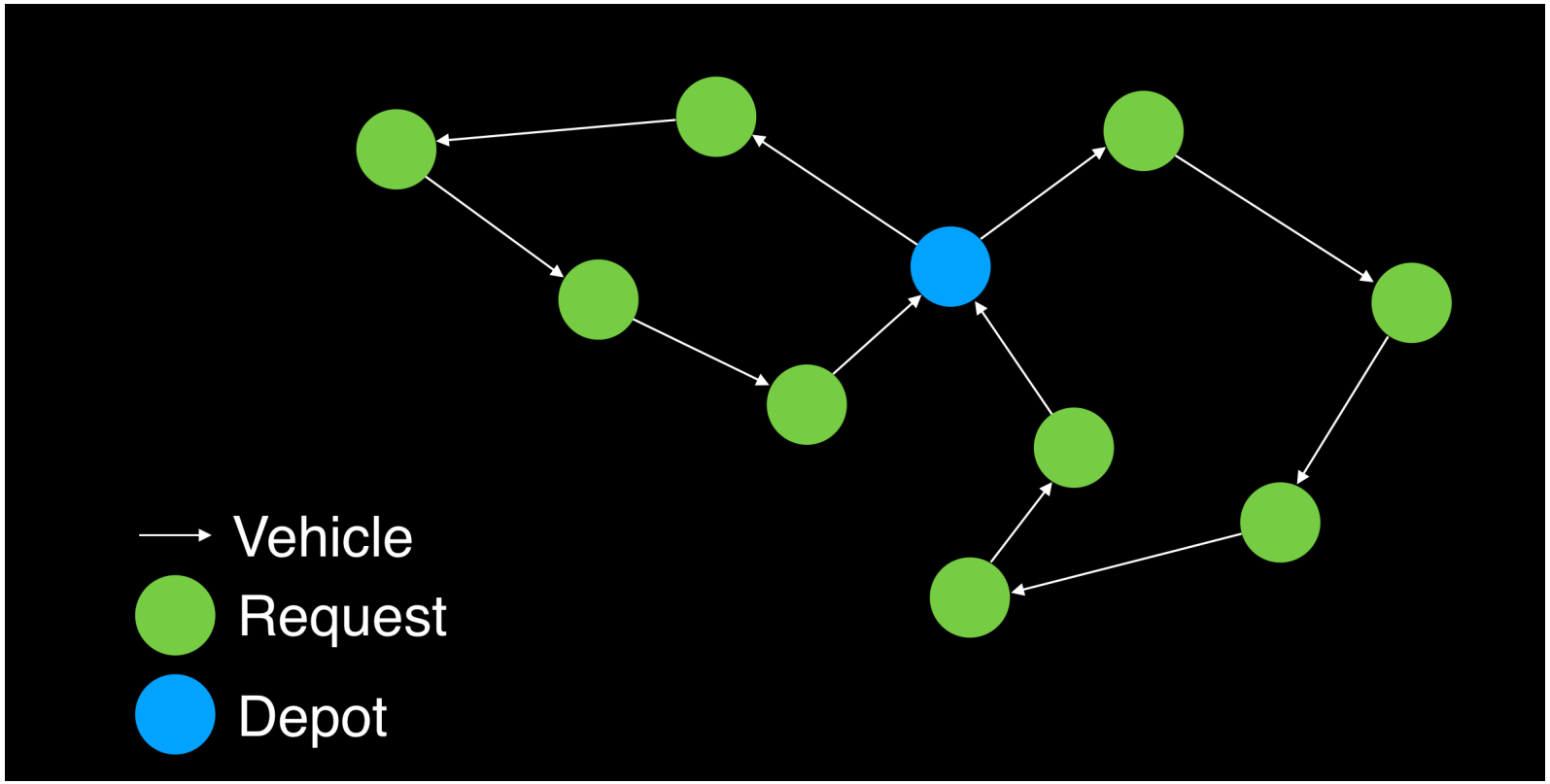
VRP with Location Constraints



- The problem
  - multiple vehicles
  - pickup and delivery constraints
  - time windows
  - capacity constraints
- Location constraints
  - number of parking slots at an airport
  - number of landings and takeoffs at an airport in a given intervals

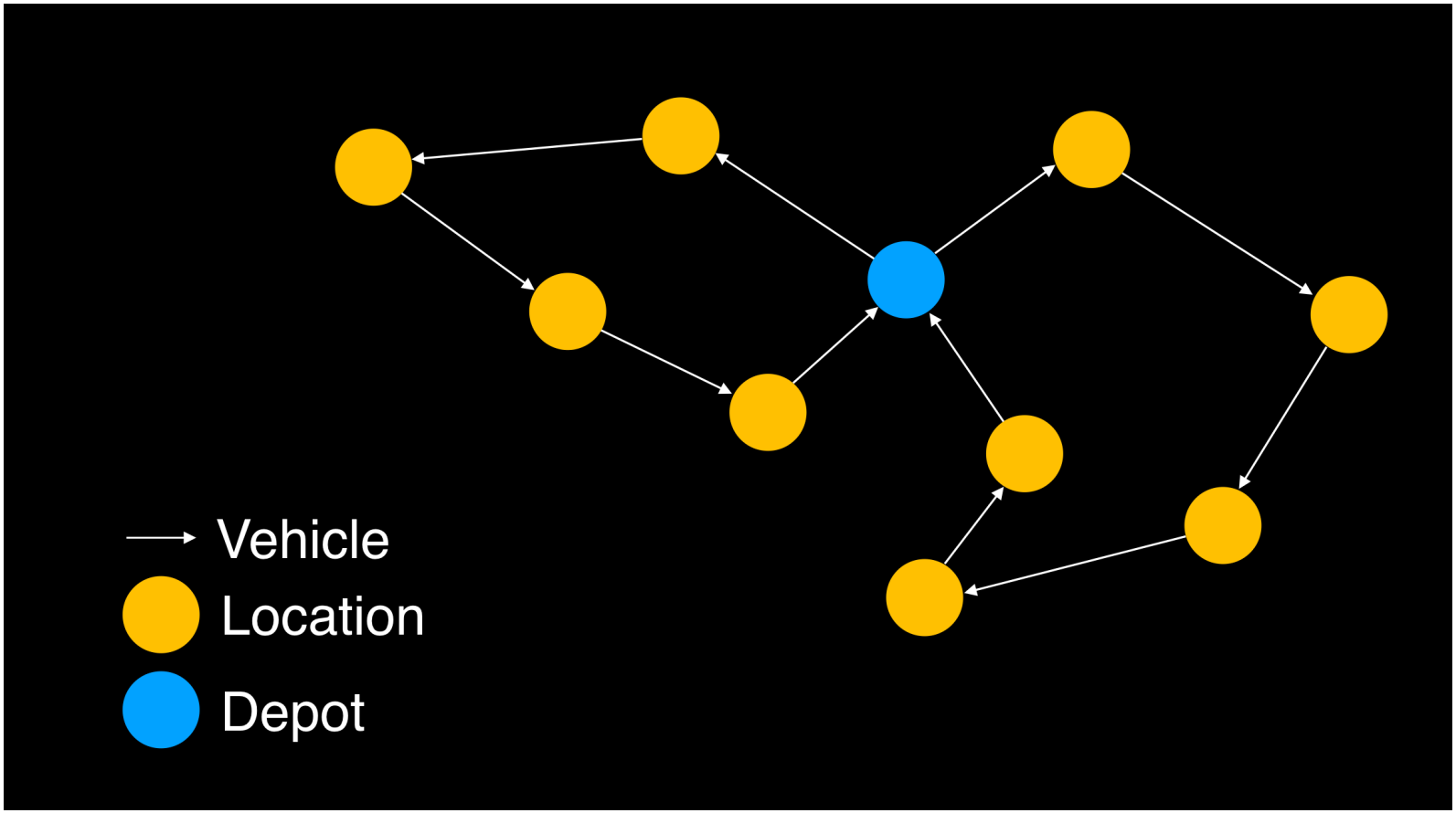
Traditional VRP

AI Institute for Advances in Optimization



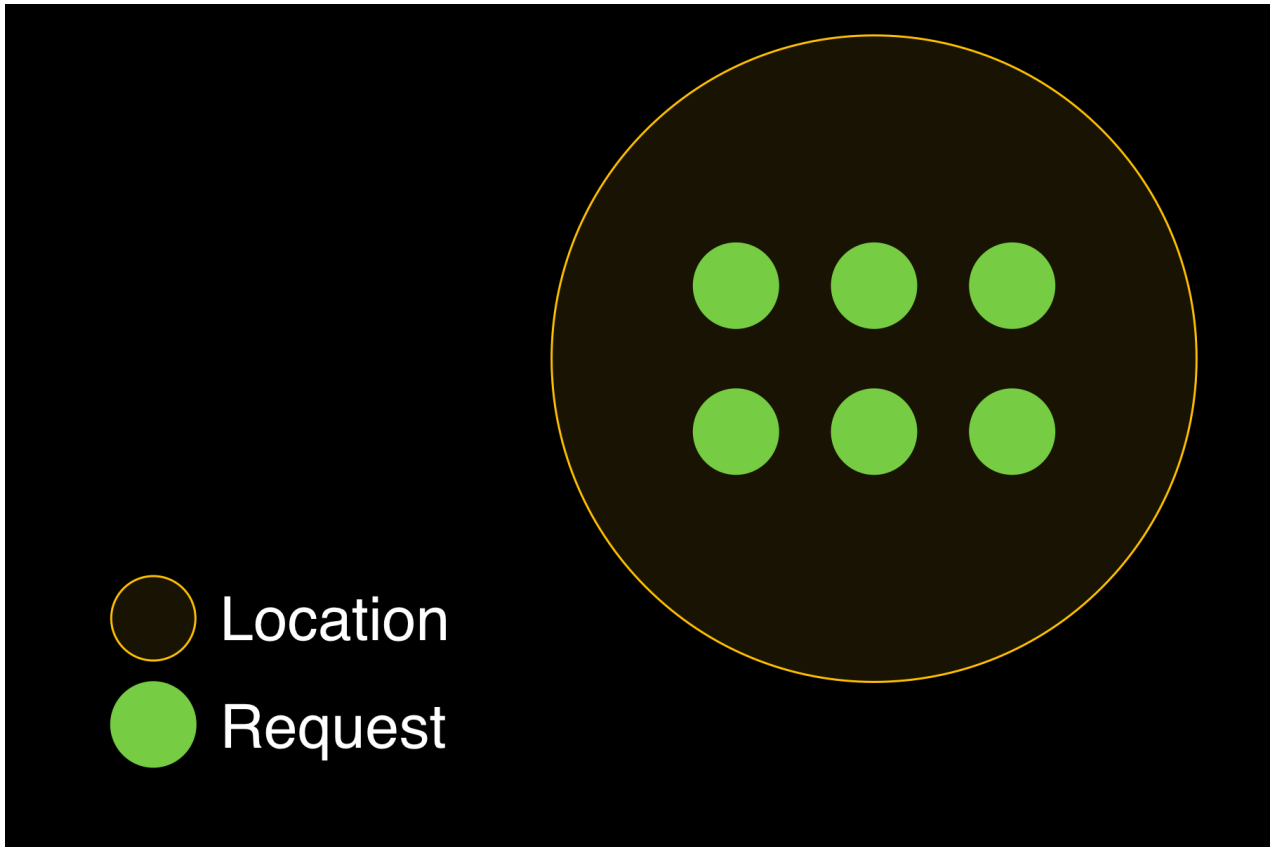
Locations

AI Institute for Advances in Optimization



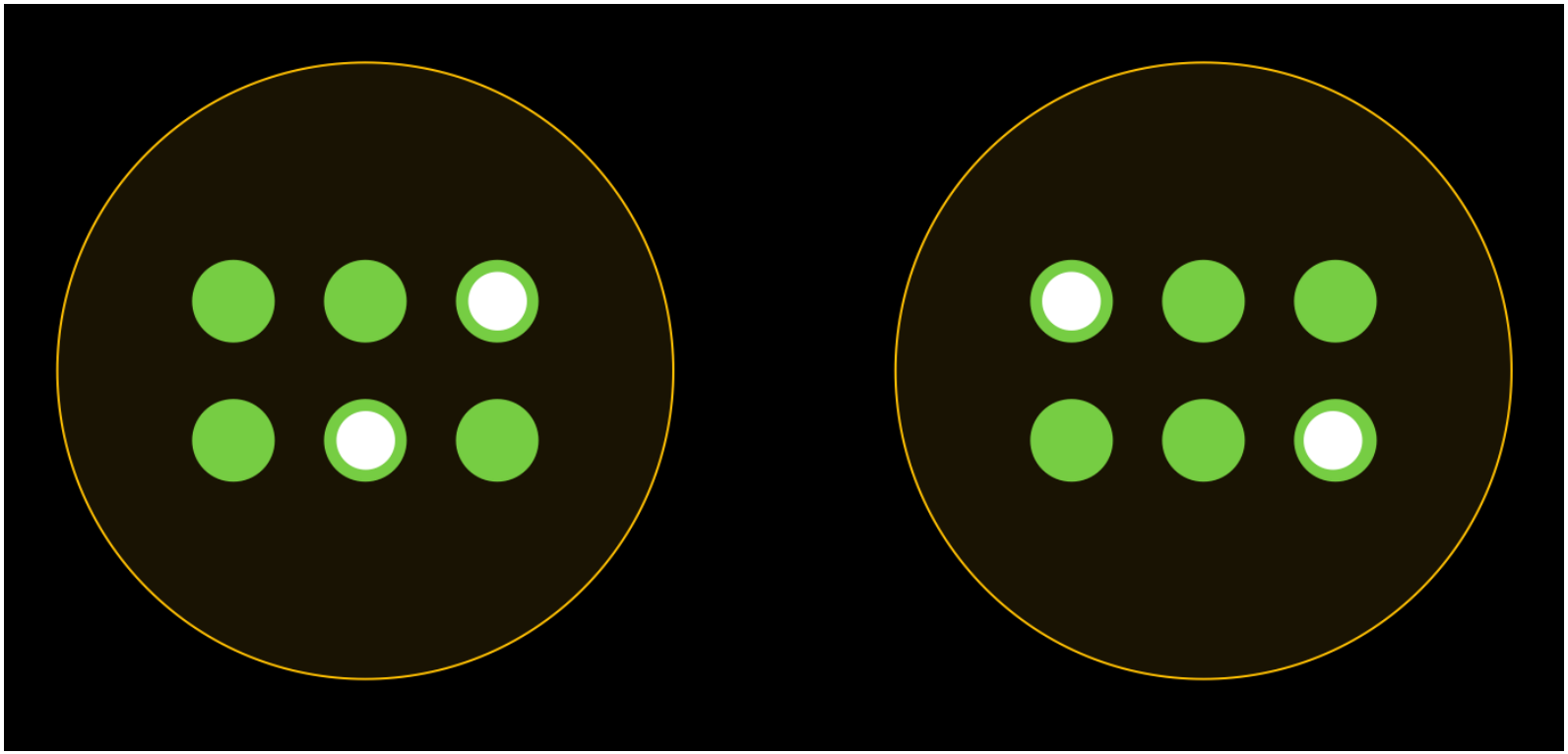
Locations and Requests

AI Institute for Advances in Optimization



Presence Constraints

AI Institute for Advances in Optimization

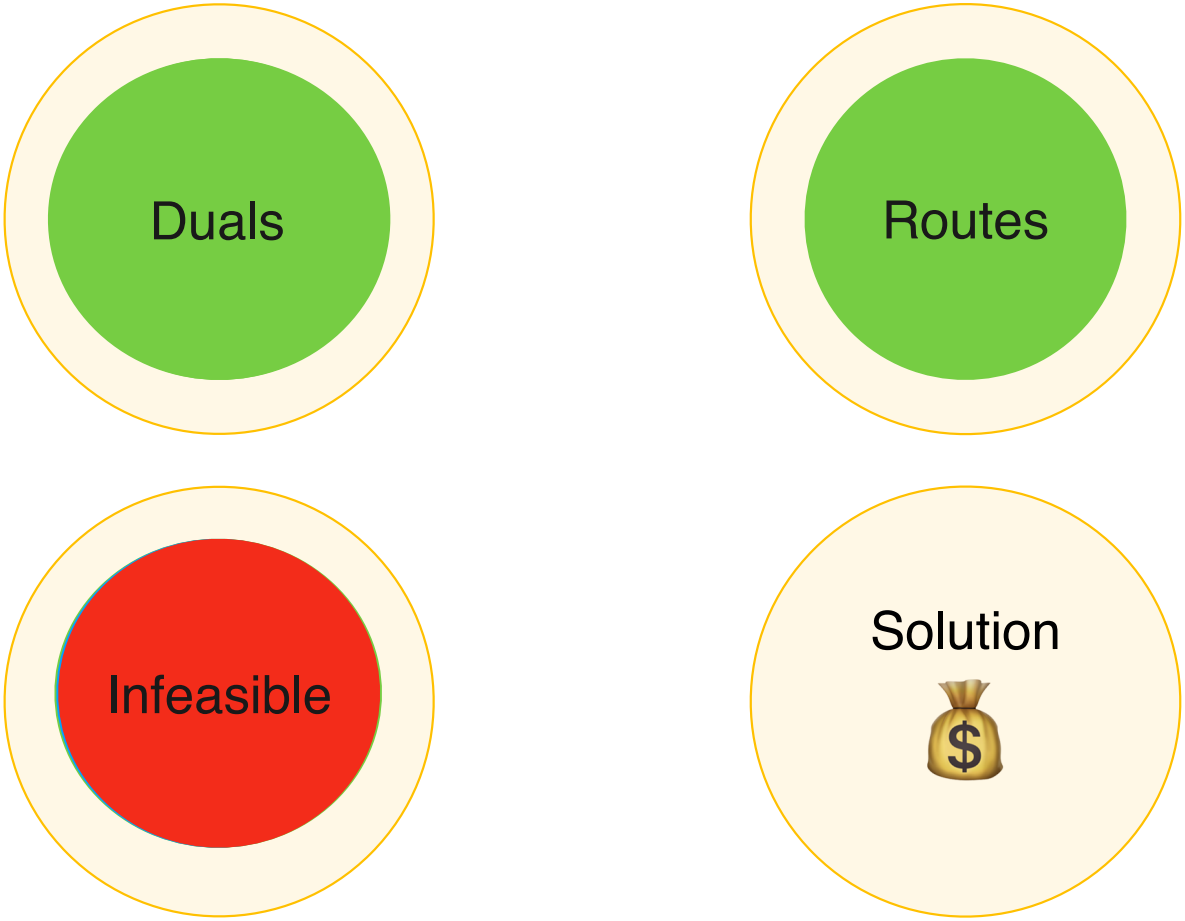


Motivating Applications

- Humanitarian and military logistics
  - number of possible landings in a base
  - number of parking spots available at a base
  - fuel available at an airport
- City logistics
  - parking at hubs
  - number of possible spots at transit centers

Branch and Price and Check

AI Institute for Advances in Optimization



The Master Problem

r is a route

every pickup is covered by a single route

$$\min \sum_{r \in \Omega} c_r x_r$$

subject to

$$\sum_{r \in \Omega} a_{i,r} x_r = 1, \qquad \forall i \in \mathcal{P},$$

$$x_r \in [0,1], \qquad \forall r \in \Omega.$$

The Pricing Problem

AI Institute for Advances in Optimization

- Standard labeling algorithm
  - solving a resource-constrained shortest path with
    - time windows
    - pickup and delivery constraints
    - capacity constraints on the load of the vehicles

- Reduced costs

$$\overline{d}_{ij} = \begin{cases} d_{ij} - \pi_i + \sum_{B \in \mathcal{B}} 1_{Bij} \mu_B, & \forall i \in \mathcal{P}, j \in \mathcal{N}, \\ d_{ij} \qquad\quad + \sum_{B \in \mathcal{B}} 1_{Bij} \mu_B, & \forall i \in \mathcal{N} \setminus \mathcal{P}, j \in \mathcal{N}, \end{cases}$$

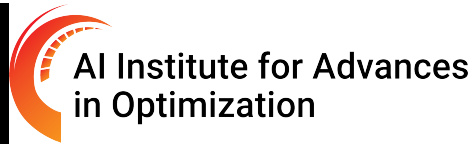
The Separation Problem

AI Institute for Advances in Optimization

- The subproblem is heavily combinatorial
  - no simple Benders composition

$$
\begin{aligned}
&\text{arr}(i) \leq \text{serv}(i), && \forall i \in \mathcal{R},\\
&\text{serv}(i) + t(i) \leq \text{dep}(i), && \forall i \in \mathcal{R},\\
&\text{arr}(i) = \text{serv}(i) = \text{dep}(i), && \forall i \in \mathcal{S} \cup \mathcal{E},\\
&\text{dep}(i) + d(i, \text{succ}(i)) = \text{arr}(\text{succ}(i)), && \forall i \in \mathcal{R} \cup \mathcal{S},\\
&\textsc{Cumulative}(\{\text{serv}(i) : i \in \mathcal{R}_l\}, \{t(i) : i \in \mathcal{R}_l\}, \mathbf{1}, C_l), && \forall l \in \mathcal{L}.
\end{aligned}
$$

The Separation problem



- How do we exclude incompatible routes?
  - in other words, what are the combinatorial Benders cuts?

Combinatorial Benders Cuts

AI Institute for Advances in Optimization

$$\sum_{r \in \Omega} B_r x_r \leq |B| - 1, \qquad \forall B \in \mathcal{B},$$

Numbers of arcs in B for a route r

Arcs in the infeasible routes

Branch and Price and Check

AI Institute for Advances in Optimization

1. Master
   - solve the master problem
2. Separation
   - check if the “integer” routes satisfy the cumulative constraints; add the cut if infeasible & go to (1)
3. Feasibility
   - if all routes are integral, a solution has been found; go to (6)
4. Pricing
   - generate new routes; if any, go to (1)
5. Branching
   - consider a route and branch on the prefixes
6. Node selection
   - take an open node and go to (1)

VRP with Location Constraints

AI Institute for Advances in Optimization

- MIP
  - cumulative constraints implemented with logical constraints (no time-indexed formulation)
- CP
  - VRP + cumulative constraints
- Branch and Price and Check
  - branch and price
  - checking cumulative constraints with CP

Experimental Results

AI Institute for Advances in Optimization

			Instance Set 1								Instance Set 2								Instance Set 3							
			TS	MIP			CP	BPC			TS	MIP			CP	BPC			TS	MIP			CP	BPC		
$\|\mathcal{L}\|$	$\|\mathcal{P}\|$	C_l	UB	UB	LB	Gap	UB	UB	LB	Gap	NG	UB	UB	LB	Gap	UB	UB	LB	Gap	NG	UB	UB	LB	Gap	UB	UB	LB	Gap	NG
8	80	1	–	–	0	–	×	–	746	–	3,171	–	–	0	–	×	–	1,176	–	28,124	–	–	0	–	×	–	390	–	2,865
		2	–	–	0	–	×	–	746	–	3,169	–	–	0	–	×	–	1,176	–	27,082	–	–	0	–	–	–	390	–	17
		3	–	–	0	–	–	–	746	–	13,897	–	–	0	–	×	–	1,176	–	28,660	–	–	0	–	–	–	391	–	2,730
		4	–	–	0	–	–	845	755	10.7%	2,971	–	–	0	–	–	–	1,176	–	28,040	393	–	0	–	–	**393**	393	0.0%	111
		5	794	–	0	–	–	799	755	5.5%	1,903	–	–	0	–	–	1,370	1,182	13.7%	20,805	393	–	0	–	–	**393**	393	0.0%	0
		6	794	–	0	–	–	795	761	4.3%	149	1,195	–	0	–	–	1,195	1,182	1.1%	2,565	393	–	0	–	–	**393**	393	0.0%	0
		7	794	–	0	–	–	794	762	4.0%	0	1,195	–	0	–	–	1,194	1,184	0.8%	1,729	393	–	0	–	–	**393**	393	0.0%	0
		8	794	–	0	–	–	794	762	4.0%	0	1,195	–	0	–	–	1,195	1,184	0.9%	2,242	393	–	0	–	–	**393**	393	0.0%	0
		9	794	–	0	–	–	794	762	4.0%	0	1,195	–	0	–	–	1,195	1,186	0.8%	0	393	–	0	–	–	**393**	393	0.0%	0
		10	794	–	0	–	–	794	762	4.0%	0	1,195	–	0	–	–	1,195	1,186	0.8%	0	393	–	0	–	–	**393**	393	0.0%	0
		11	794	–	0	–	–	794	762	4.0%	0	1,195	–	0	–	–	1,195	1,186	0.8%	0	393	–	0	–	–	**393**	393	0.0%	0
		12	794	–	0	–	–	794	762	4.0%	0	1,195	–	0	–	–	1,195	1,186	0.8%	0	393	–	0	–	–	**393**	393	0.0%	0
		13	794	–	0	–	–	794	762	4.0%	0	1,195	–	0	–	–	1,195	1,186	0.8%	0	393	–	0	–	–	**393**	393	0.0%	0
		14	794	–	0	–	–	794	762	4.0%	0	1,195	–	0	–	–	1,195	1,186	0.8%	0	393	–	0	–	–	**393**	393	0.0%	0
		15	794	–	0	–	–	793	763	3.8%	0	1,195	–	0	–	–	1,195	1,186	0.8%	0	393	–	0	–	–	**393**	393	0.0%	0

			Instance Set 1									Instance Set 2									Instance Set 3								
			TS	MIP			CP	BPC				TS	MIP			CP	BPC				TS	MIP			CP	BPC			
$\|\mathcal{L}\|$	$\|\mathcal{P}\|$	C_l	UB	UB	LB	Gap	UB	UB	LB	Gap	NG	UB	UB	LB	Gap	UB	UB	LB	Gap	NG	UB	UB	LB	Gap	UB	UB	LB	Gap	NG
11	80	1	–	–	4	–	×	–	876	–	8,763	–	–	0	–	×	–	1,260	–	24,022	–	–	0	–	×	–	411	–	10,388
		2	–	–	4	–	×	–	872	–	12,877	–	–	0	–	×	–	1,260	–	25,102	–	–	0	–	–	–	407	–	12
		3	–	–	4	–	–	–	875	–	9,282	–	–	0	–	×	–	1,260	–	24,733	–	–	0	–	–	–	410	–	780
		4	–	–	4	–	–	–	881	–	914	–	–	0	–	–	–	1,260	–	25,974	413	–	0	–	–	413	411	0.5%	184
		5	883	–	4	–	–	**883**	883	0.0%	32	–	–	0	–	–	1,340	1,260	6.0%	11,902	413	–	0	–	–	413	412	0.2%	0
		6	883	–	4	–	–	**883**	883	0.0%	0	–	–	0	–	–	1,319	1,266	4.0%	5,011	413	–	0	–	–	413	412	0.2%	0
		7	883	–	4	–	–	**883**	883	0.0%	0	–	–	0	–	–	1,312	1,269	3.3%	2,083	413	–	0	–	–	413	412	0.2%	0
		8	883	–	4	–	–	**883**	883	0.0%	0	1,280	–	0	–	–	1,280	1,270	0.8%	2,165	413	–	0	–	–	413	412	0.2%	0
		9	883	–	4	–	–	**883**	883	0.0%	0	1,280	–	0	–	–	1,280	1,272	0.6%	0	413	–	0	–	–	413	412	0.2%	0
		10	883	–	4	–	–	**883**	883	0.0%	0	1,280	–	0	–	–	1,280	1,272	0.6%	0	413	–	0	–	–	413	412	0.2%	0
		11	883	–	4	–	–	**883**	883	0.0%	0	1,280	–	0	–	–	1,280	1,272	0.6%	0	413	–	0	–	–	413	412	0.2%	0
		12	883	–	4	–	–	**883**	883	0.0%	0	1,280	–	0	–	–	1,280	1,272	0.6%	0	413	–	0	–	–	413	412	0.2%	0
		13	883	–	4	–	–	**883**	883	0.0%	0	1,280	–	0	–	–	1,280	1,272	0.6%	0	413	–	0	–	–	413	412	0.2%	0
		14	883	–	4	–	–	**883**	883	0.0%	0	1,280	–	0	–	–	1,280	1,272	0.6%	0	413	–	0	–	–	413	411	0.5%	0
		15	883	–	4	–	–	**883**	883	0.0%	0	1,280	–	0	–	–	1,280	1,272	0.6%	0	413	–	0	–	–	413	412	0.2%	0

Outline

AI Institute for Advances in Optimization

- Brief Overview to constraint programming
- Scheduling with constraint programming
- Hybridizing constraint programming and MIP
- Learning-based constraint programming
- Learning Optimization Proxies

AI Institute for Advances in Optimization

Constraints (2011) 16:250–282
DOI 10.1007/s10601-010-9103-2

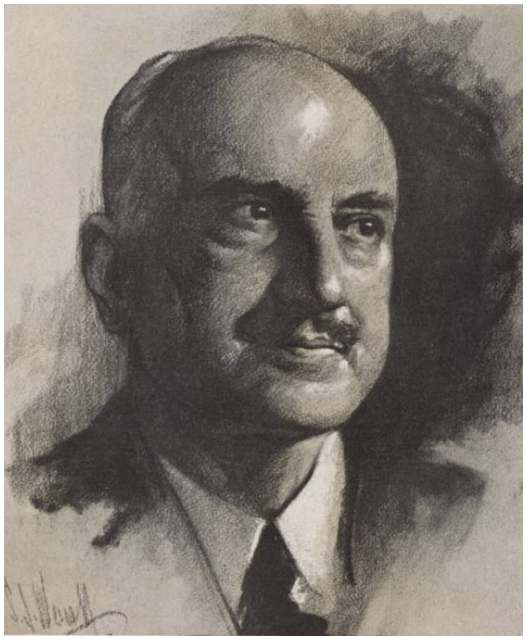
Explaining the `cumulative` propagator

Andreas Schutt · Thibaut Feydy · Peter J. Stuckey · Mark G. Wallace

Learning

AI Institute for Advances in Optimization

- Those who forget the past are doomed to repeat it

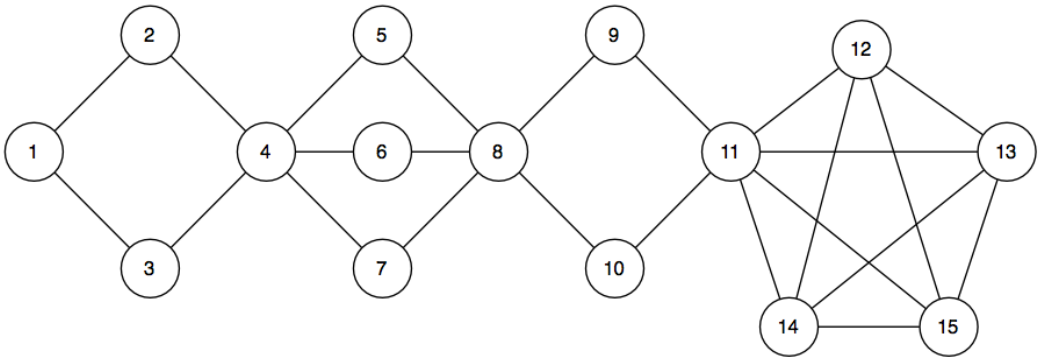


George Santayana

How Much Search is Repeated?

AI Institute for Advances in Optimization

- Color the following graph with 4 colors

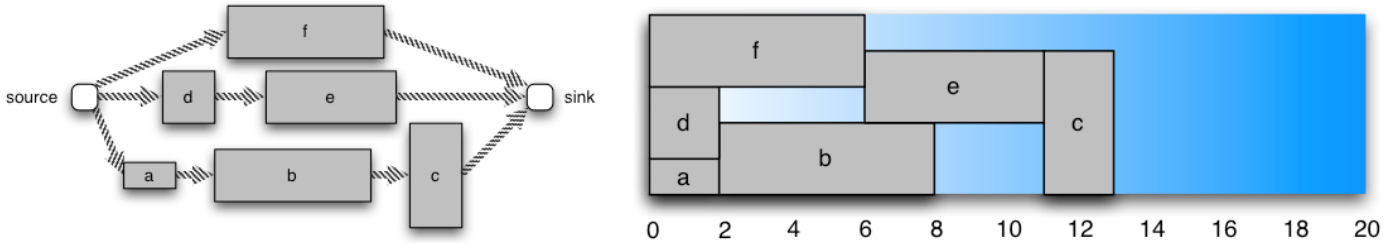


- Traditional search: 462672 failures
- Constraint Programming with Learning
  - 18 failures

How Much Search is Repeated?

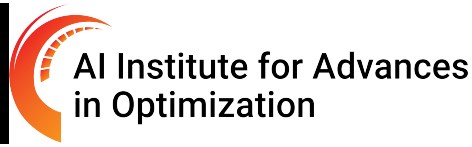
AI Institute for Advances in Optimization

- Resource Constrained Project Scheduling
  - BL instance (20 tasks)



- Input order: 934,535 failures
  - With learning: 931 failures
- Smallest start time order: 296,567 failures
  - With learning: 551 failures
- Activity-based search: > 2,000,000 failures
  - With learning: 1144 failures

Conflict-Based Learning in CP



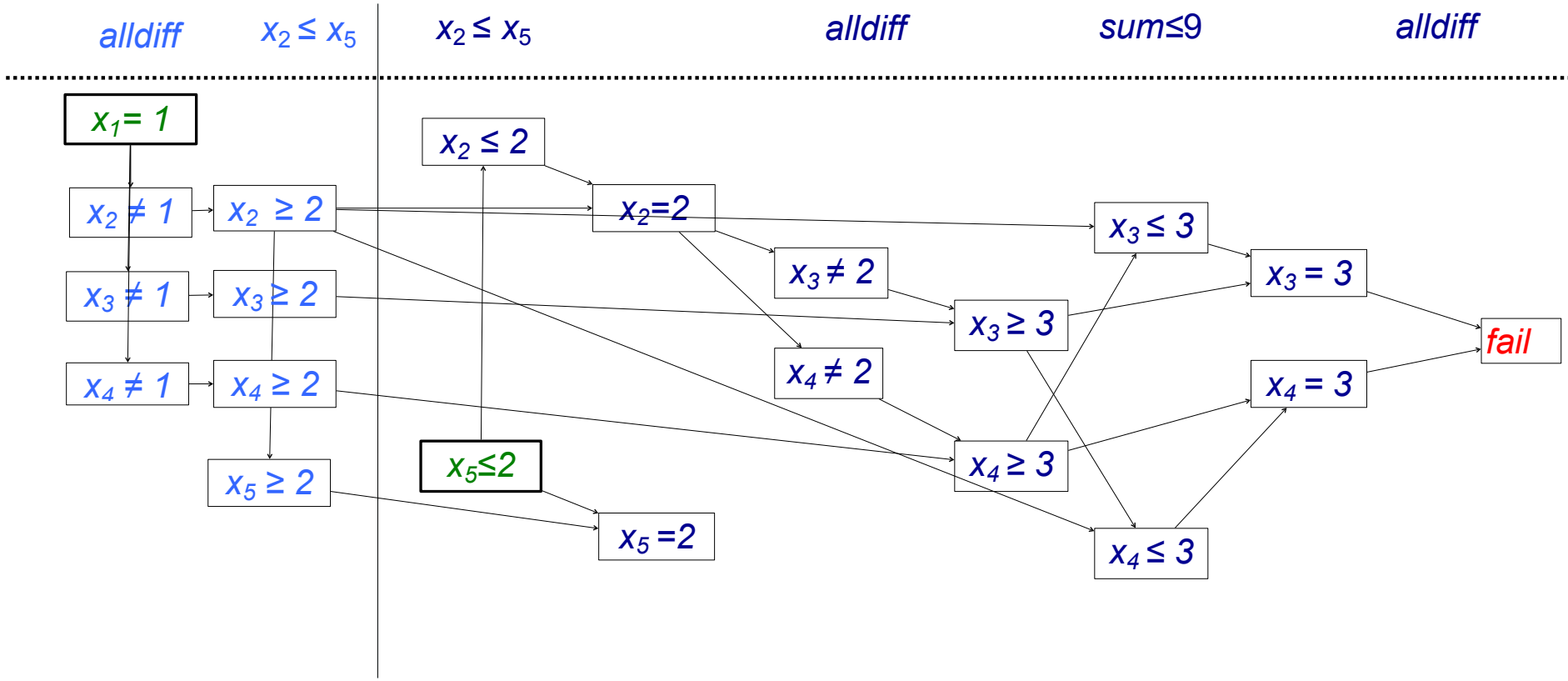
- Can we learn from failures?
  - A long history (from at least 1986)
  - but it has not really worked until recently
- Conflict-based learning
  - learn a new clause (1UIP) when encountering a failure

Inference Graph

- What does pruning do in CP?
  - $x = v$, $x \neq v$, $x \geq v$, $x \leq v$ (in its simplest form)
    - domain events
  - denote them as $[x = v]$, $[x \neq v]$, $[x \geq v]$, $[x \leq v]$
- The graph captures the CP Inferences
  - e.g. $[x \leq 2]$ and $x \geq y$ implies that $[y \leq 2]$
  - inference: $[x \leq 2] \rightarrow [y \leq 2]$

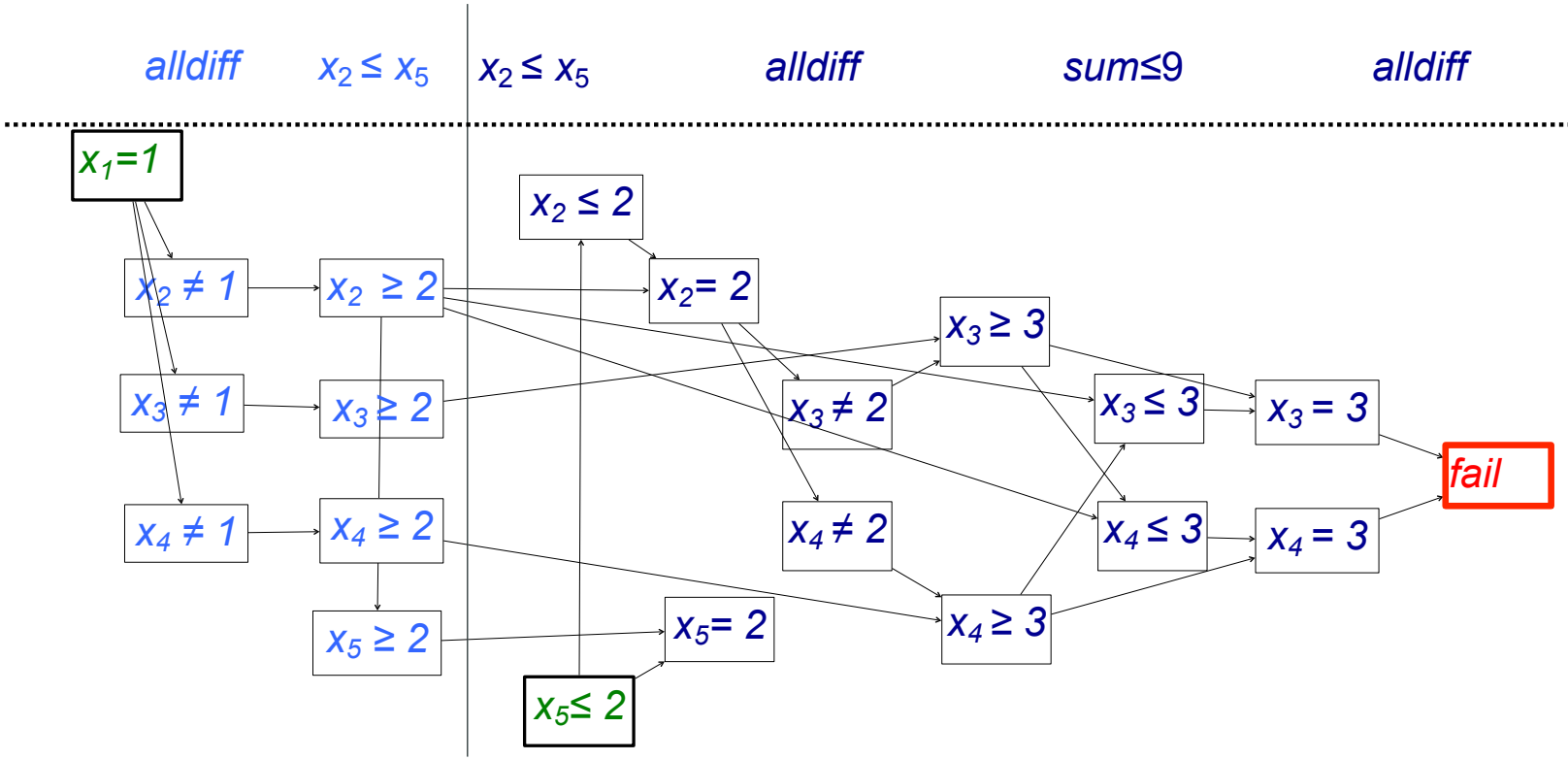
Inference Graph

AI Institute for Advances in Optimization



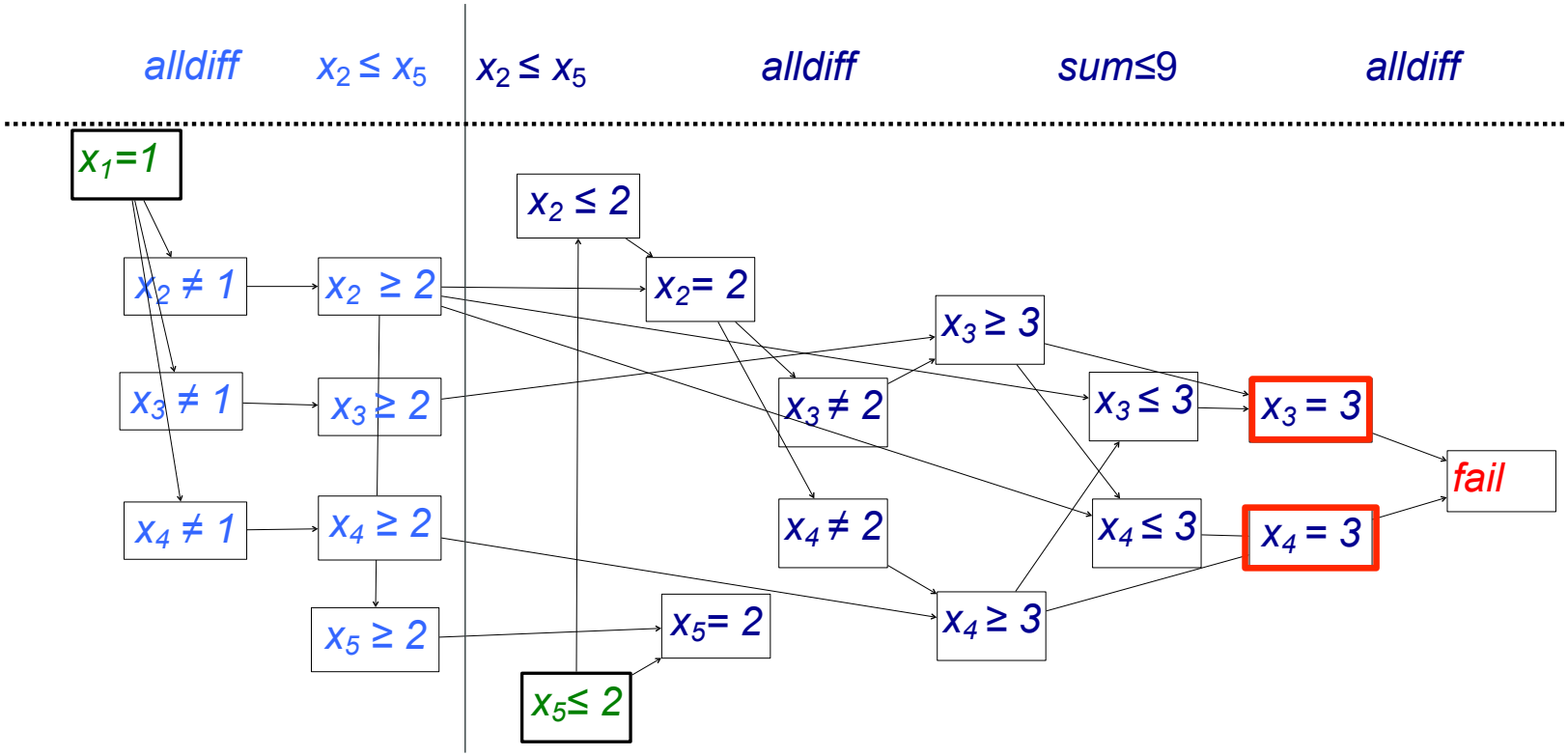
Nogood Learning

AI Institute for Advances in Optimization



Nogood Learning

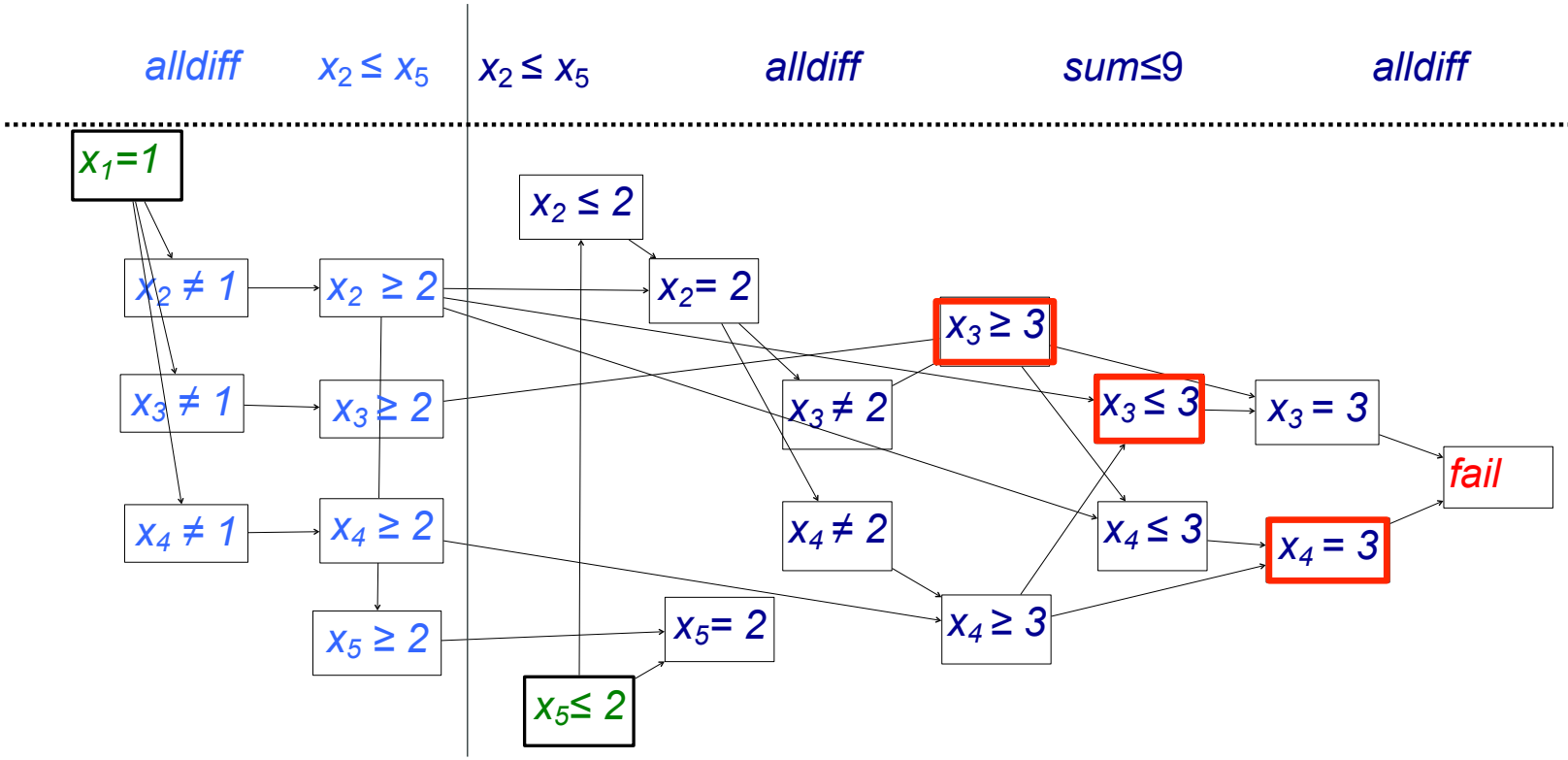
AI Institute for Advances in Optimization



$x_3=3 \wedge x_4=3 \rightarrow$ *false*

Nogood Learning

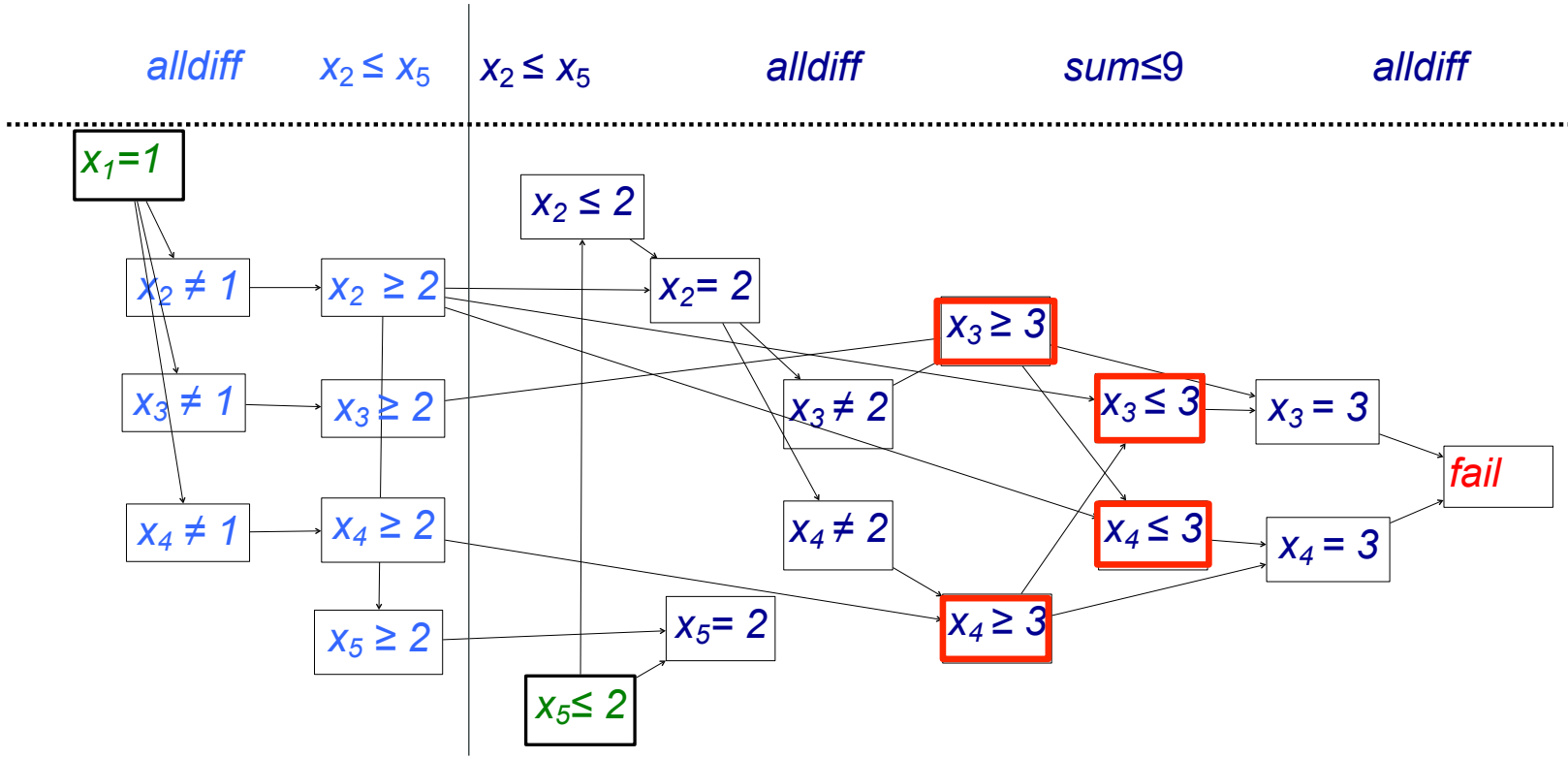
AI Institute for Advances in Optimization



$x_3 \ge 3 \wedge x_3 \le 3 \wedge x_4=3 \rightarrow$ *false*

Nogood Learning

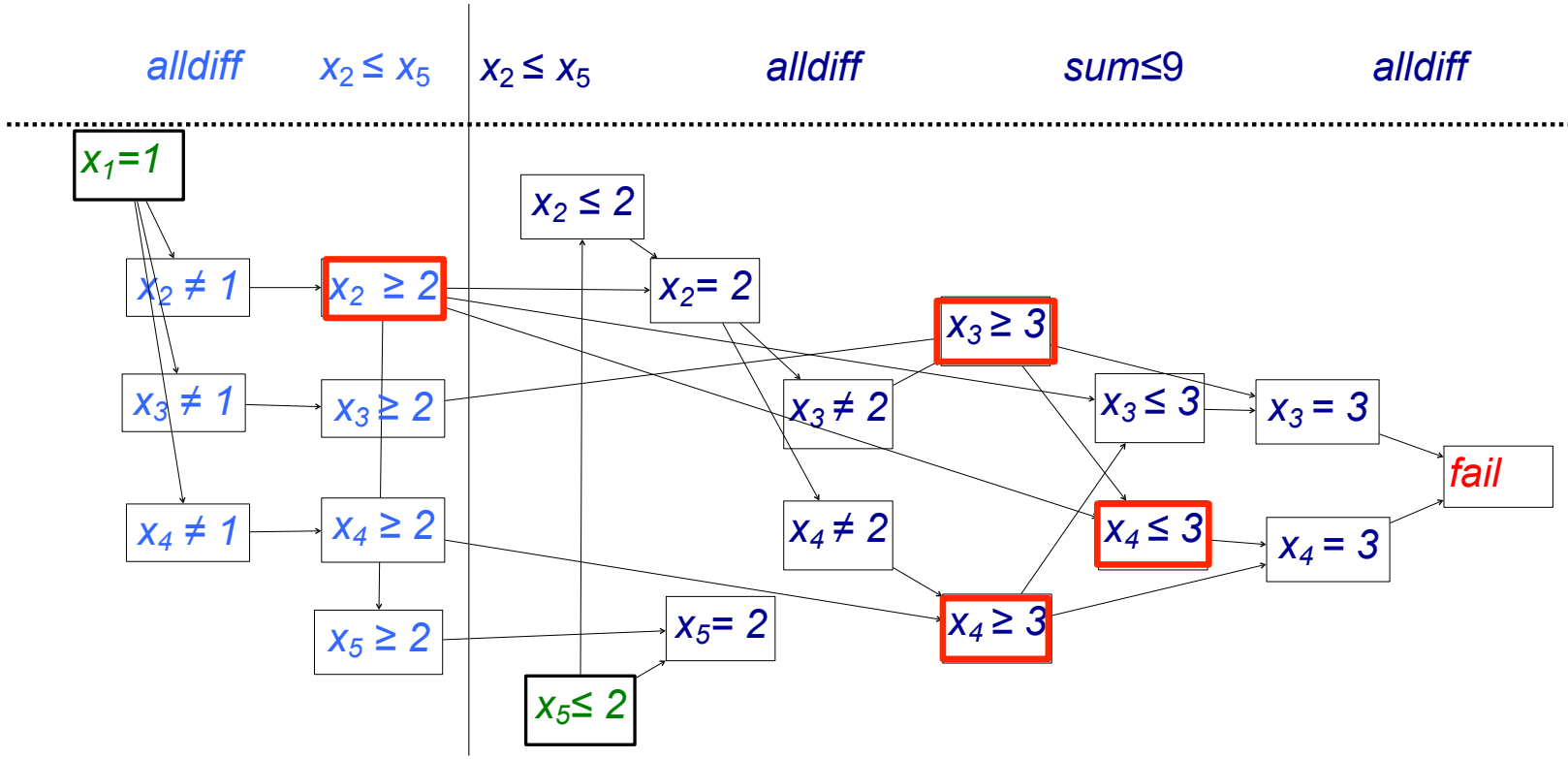
AI Institute for Advances in Optimization



$\{x_3 \ge 3, x_4 \ge 3, x_3 \le 3, x_4 \le 3\} \rightarrow$ *false*

Nogood Learning

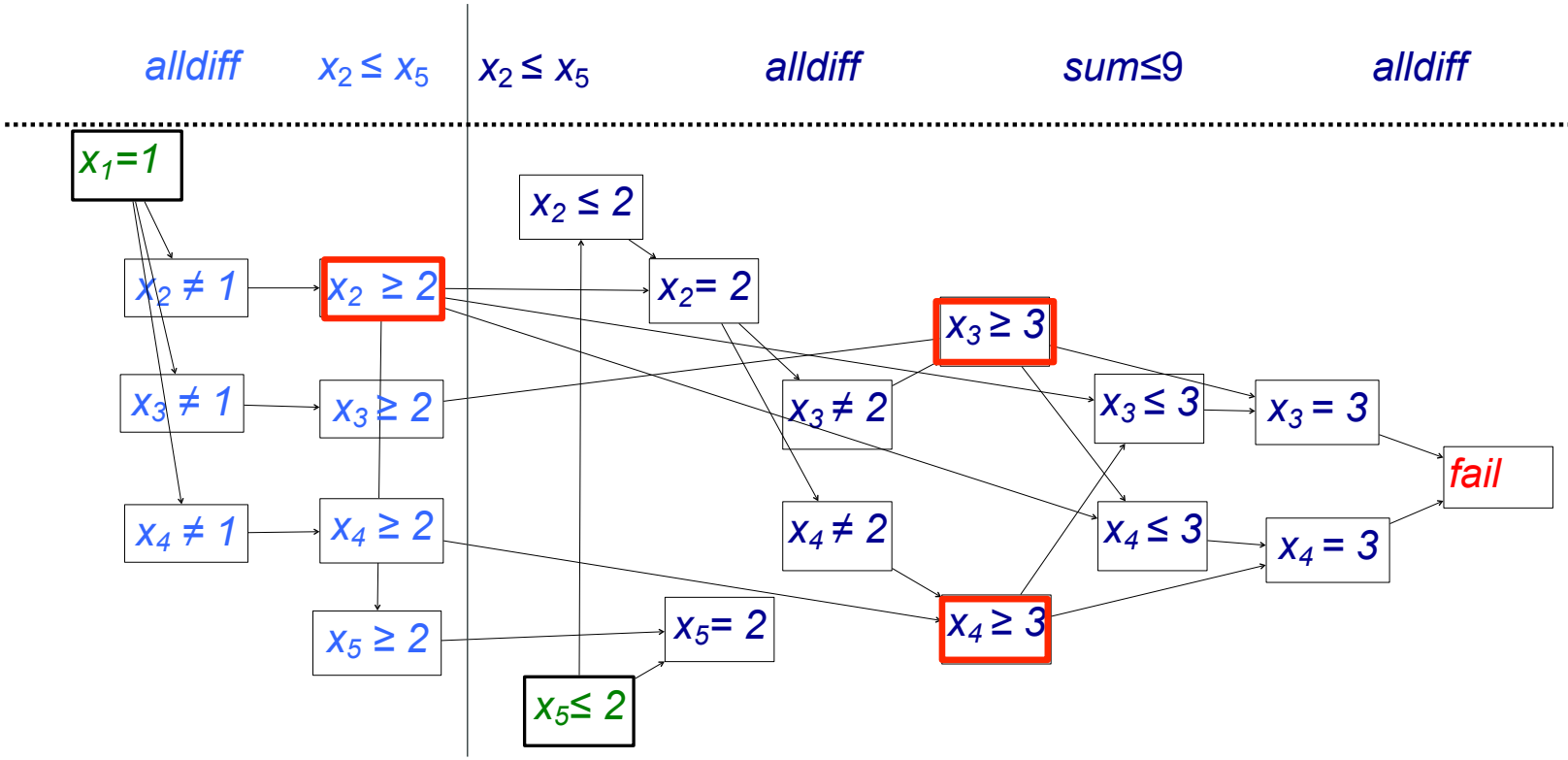
AI Institute for Advances in Optimization



$\{x_2 \ge 2, x_3 \ge 3, x_4 \ge 3, x_3 \le 3\}$ ➔ *false*

Nogood Learning

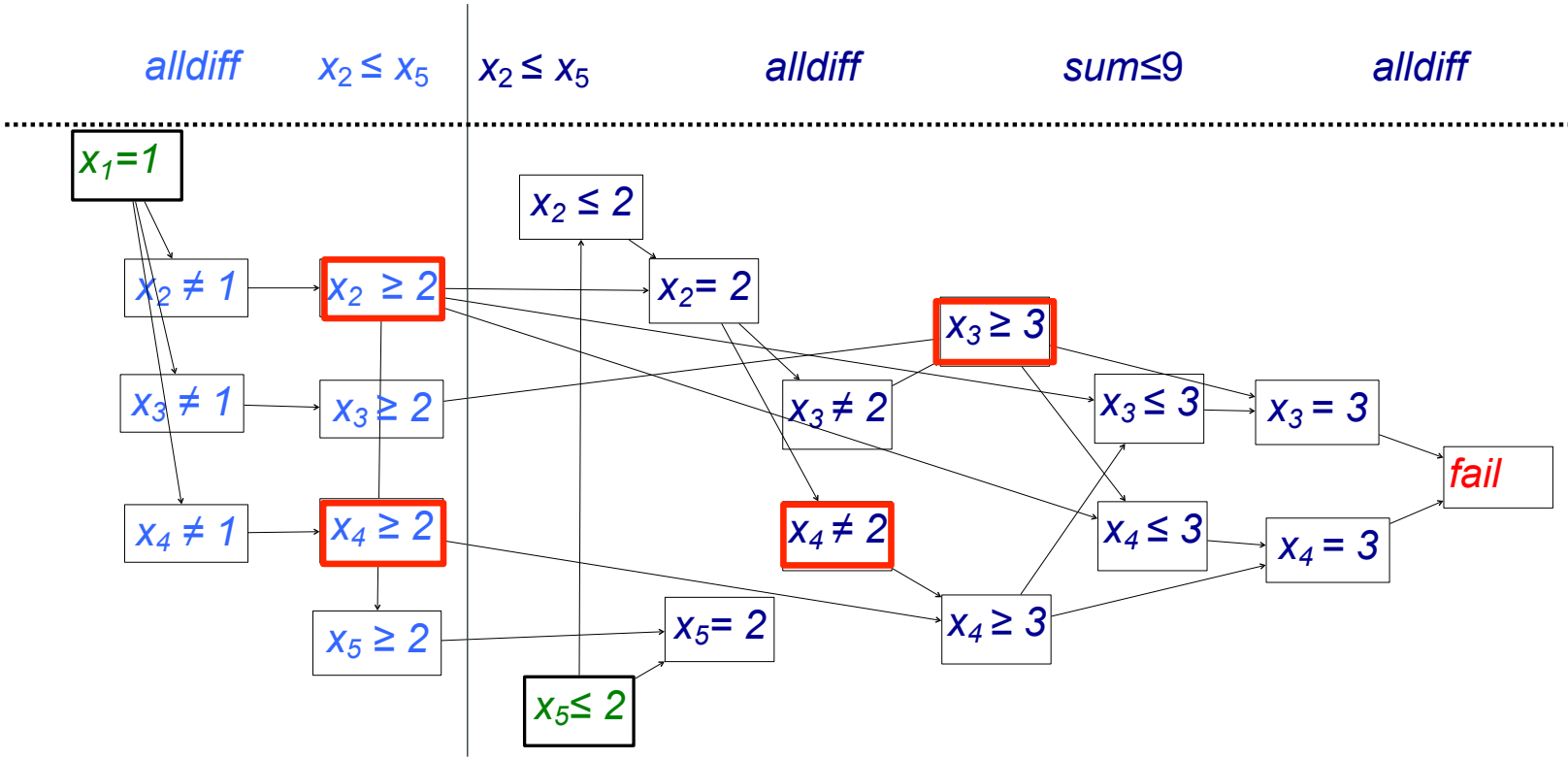
AI Institute for Advances in Optimization



$\{x_2 \ge 2, x_3 \ge 3, x_4 \ge 3\}$ ➔ *false*

Nogood Learning

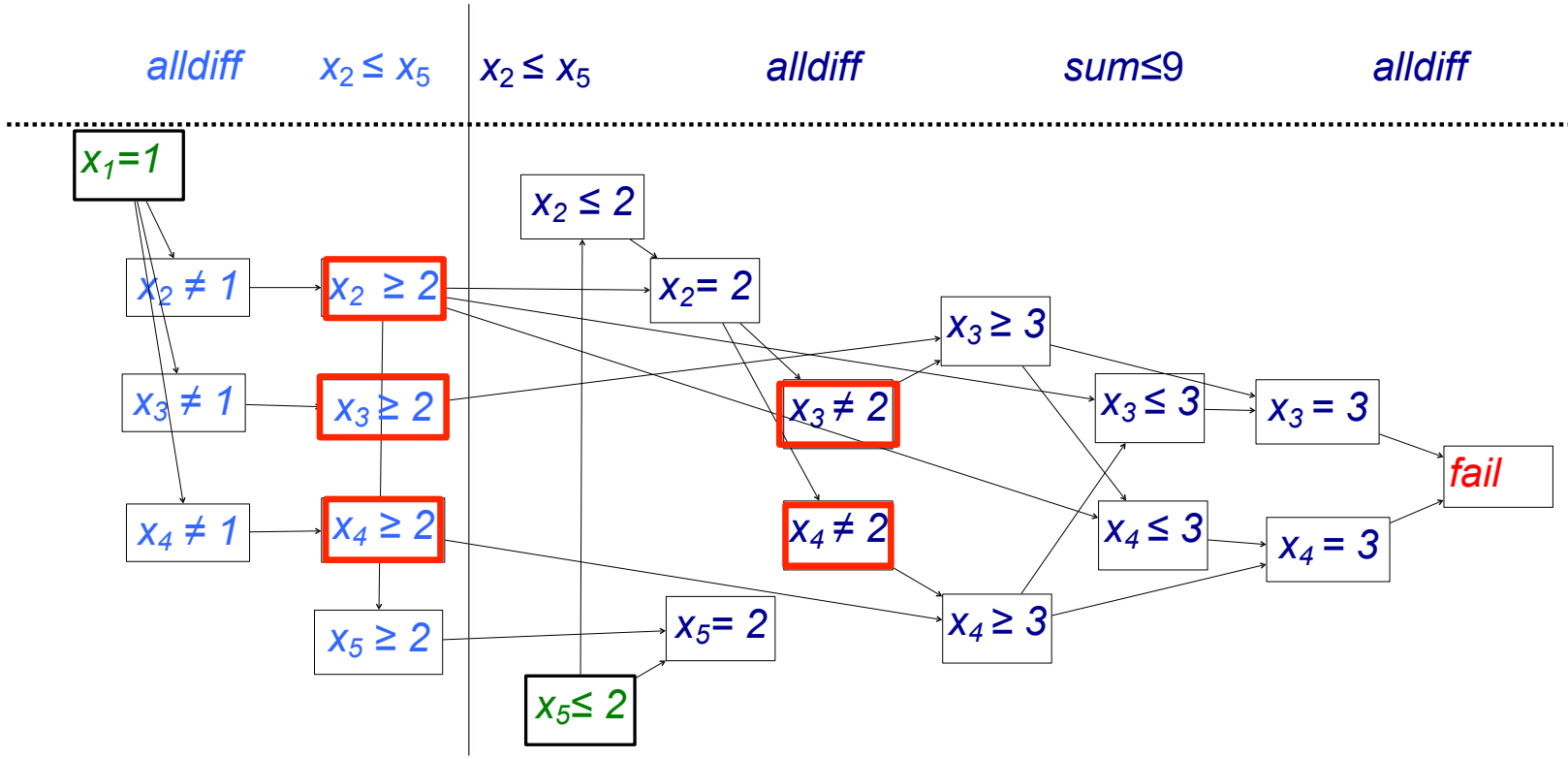
AI Institute for Advances in Optimization



$\{x_2 \ge 2, x_4 \ge 2, x_4 \neq 2, x_3 \ge 3\}$ ➔ *false*

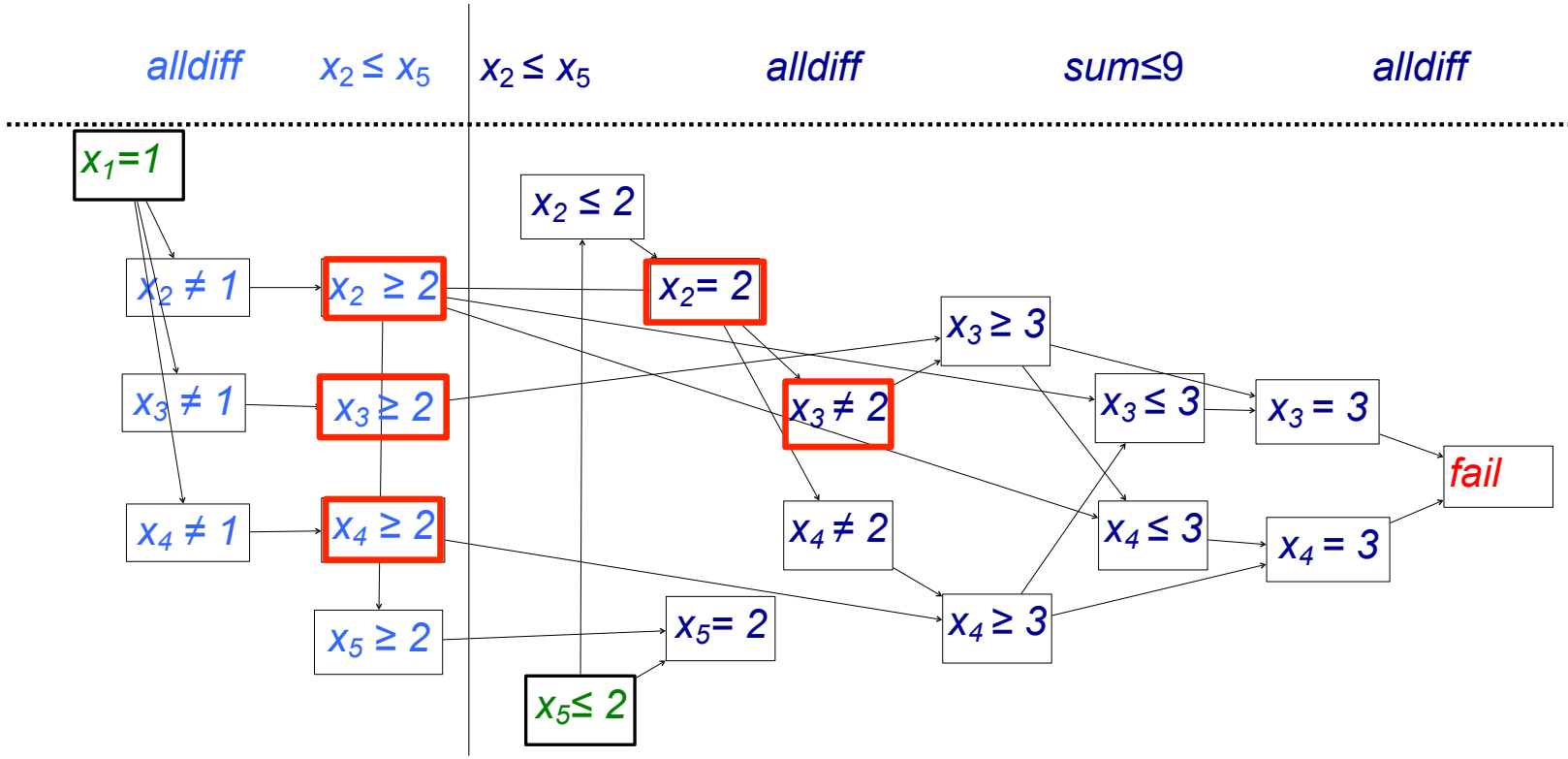
Nogood Learning

AI Institute for Advances in Optimization



$\{x_2 \ge 2, x_3 \ge 2, x_4 \ge 2, x_3 \neq 2, x_4 \neq 2\}$ ➔ *false*

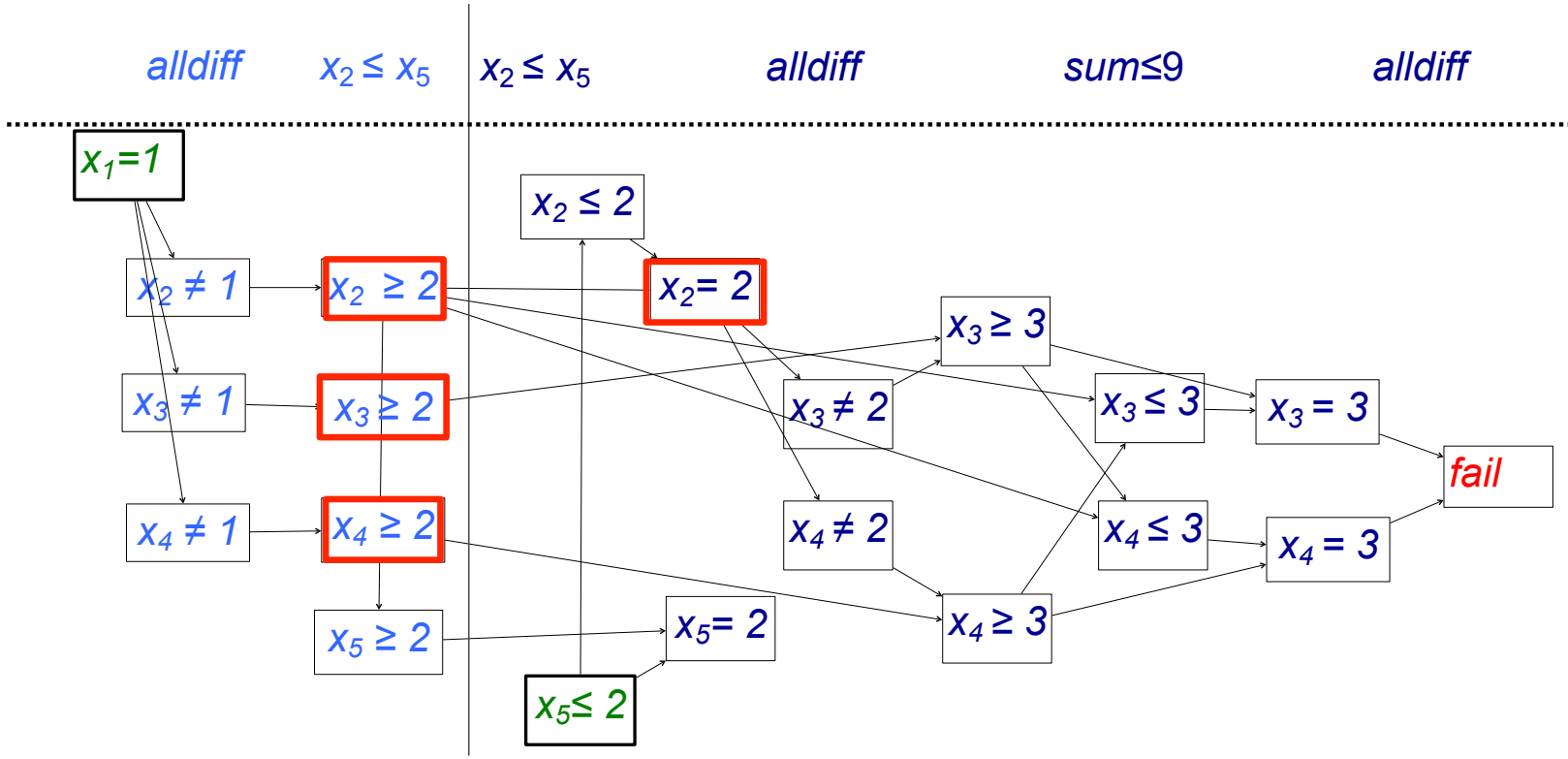
Nogood Learning



$\{x_2 \geq 2, x_3 \geq 2, x_4 \geq 2, x_2 = 2, x_3 \neq 2\}$ → *false*

Nogood Learning

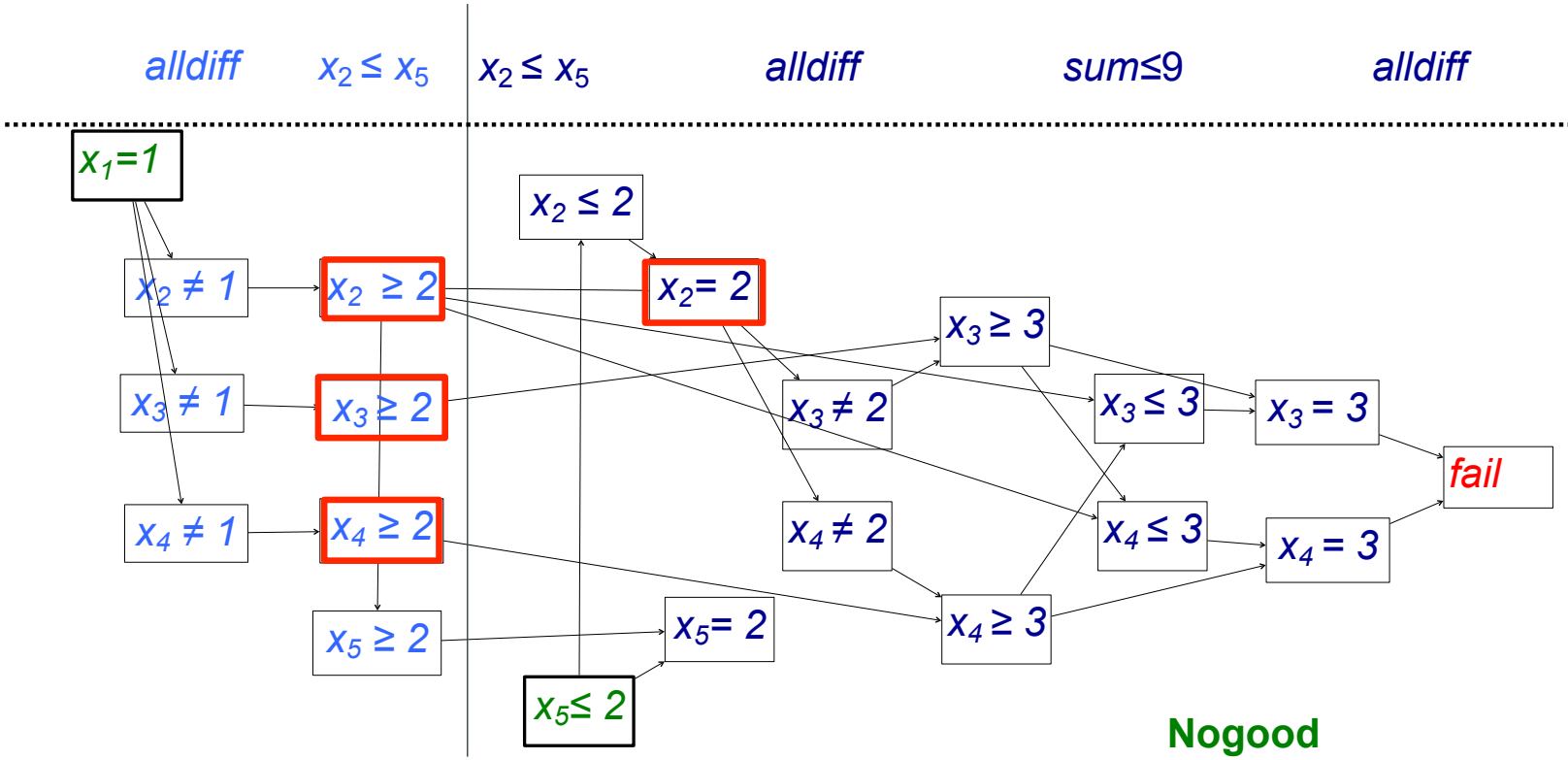
AI Institute for Advances in Optimization



$\{x_2 \ge 2, x_3 \ge 2, x_4 \ge 2, x_2 = 2\}$ ➔ *false*

Nogood Learning

AI Institute for Advances in Optimization



Nogood

$\{x_2 \ge 2, x_3 \ge 2, x_4 \ge 2, x_2 = 2\} \rightarrow$ *false*

$x_3 \le 1 \vee x_4 \le 1 \vee x_2 \neq 2$

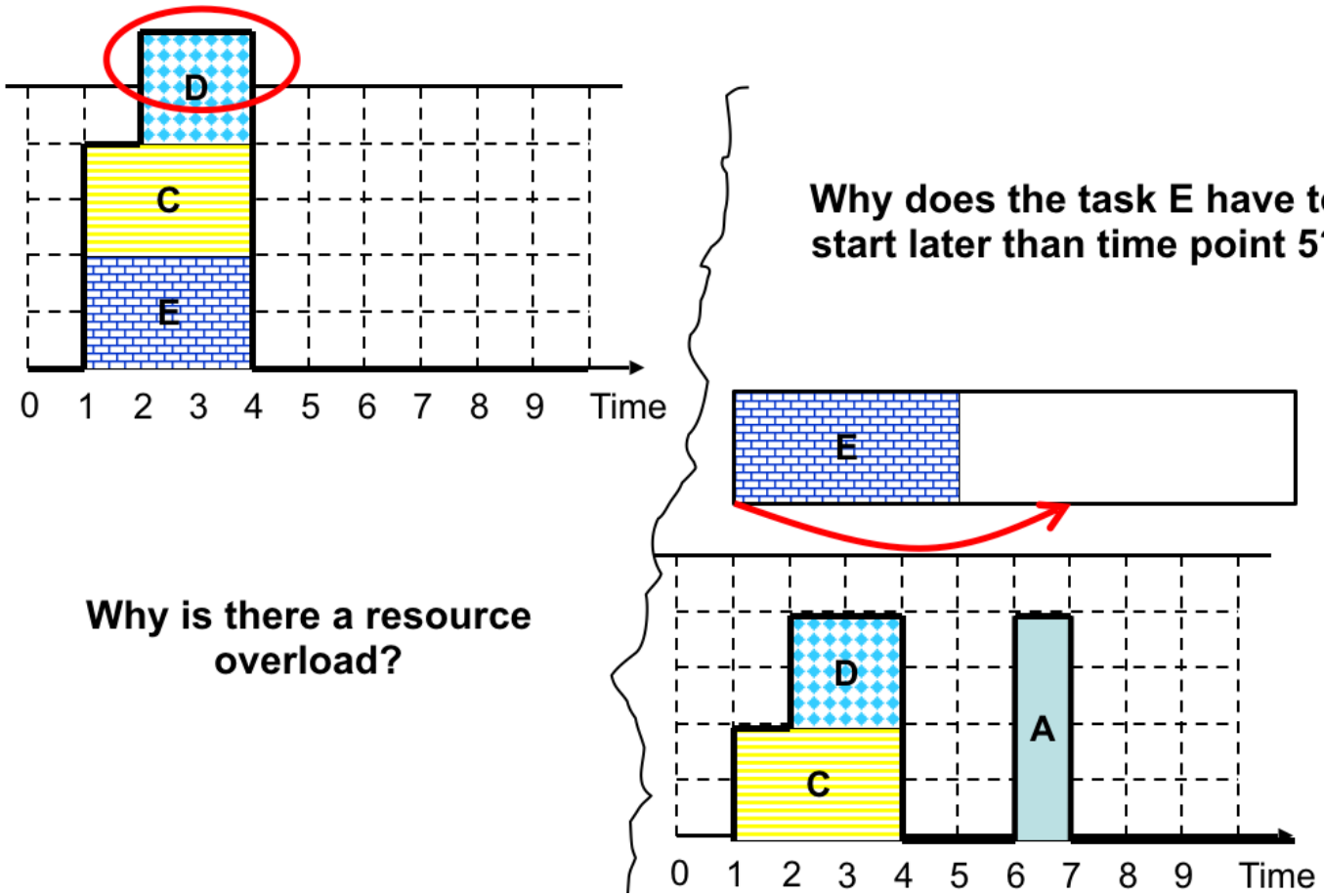
The Role of Constraints

AI Institute for Advances in Optimization

- Traditionally
  - feasibility checking
  - domain filtering
- With learning
  - explaining its failures
  - explaining its inferences
- Key goal in learning
  - make the explanations as general/reusable as possible

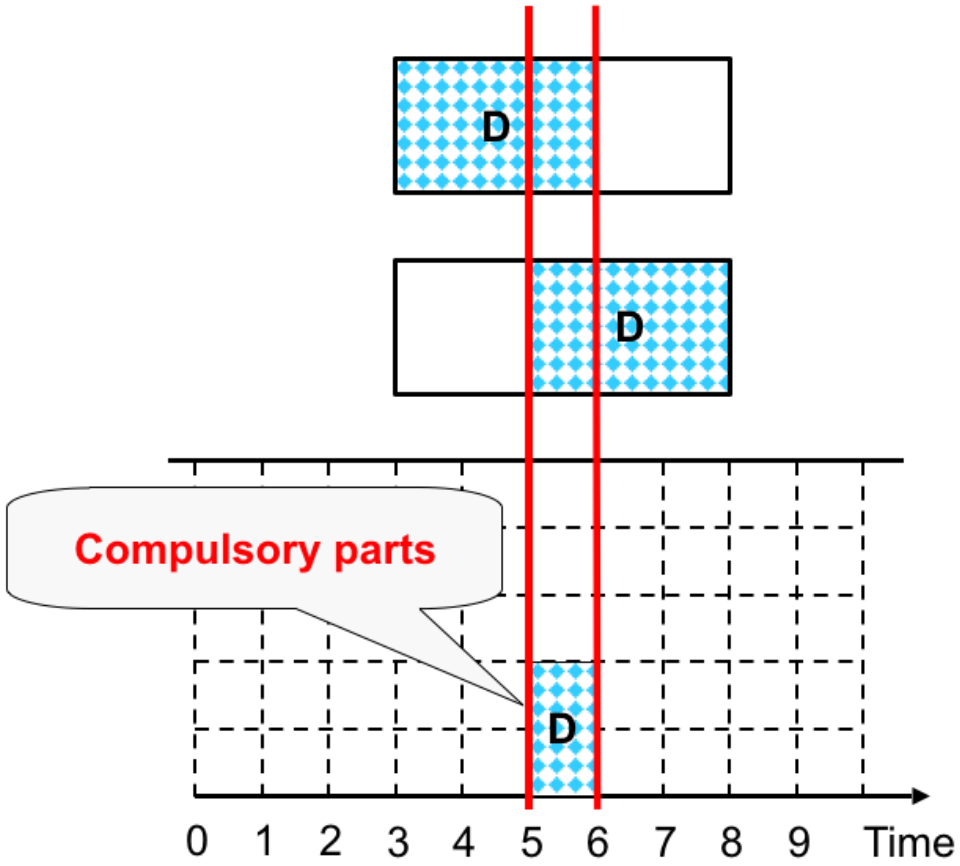
Explaining the Cumulative Constraint

AI Institute for Advances in Optimization



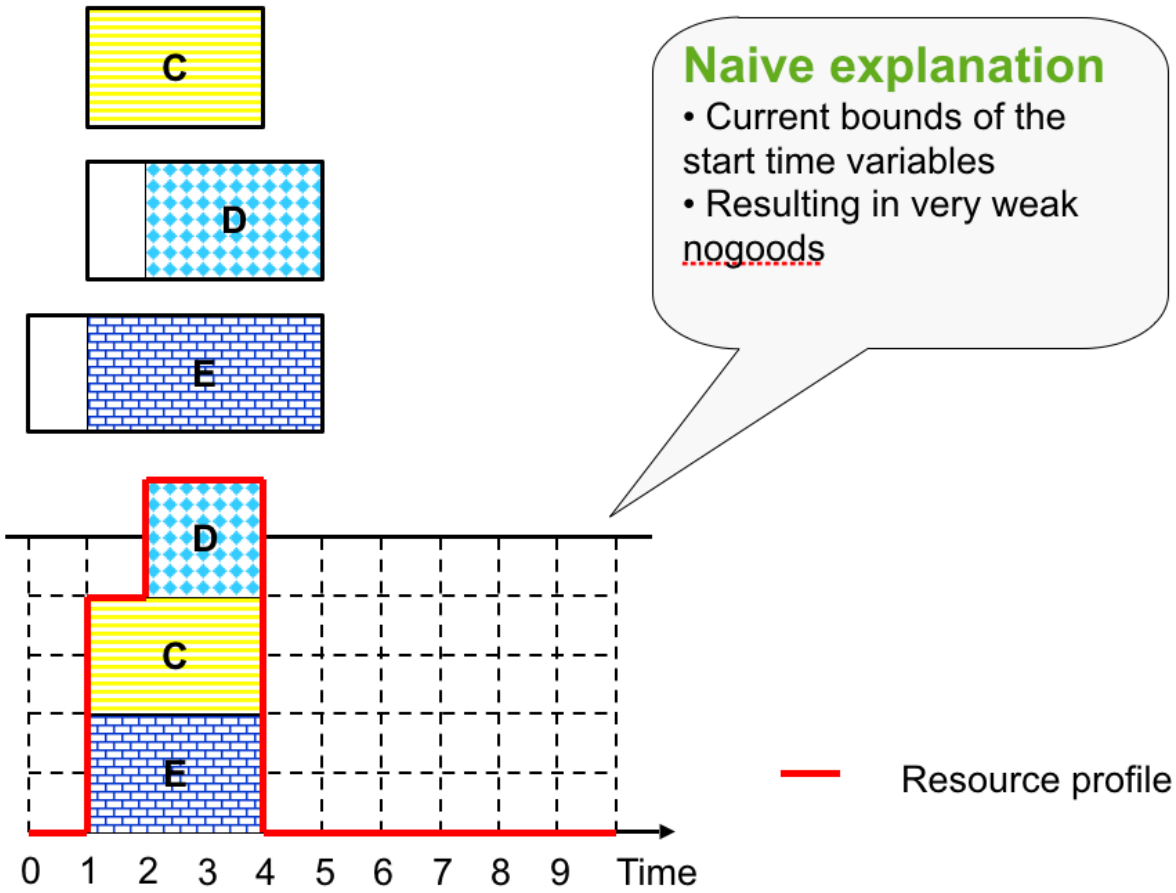
Explaining the Cumulative Constraint

AI Institute for Advances in Optimization



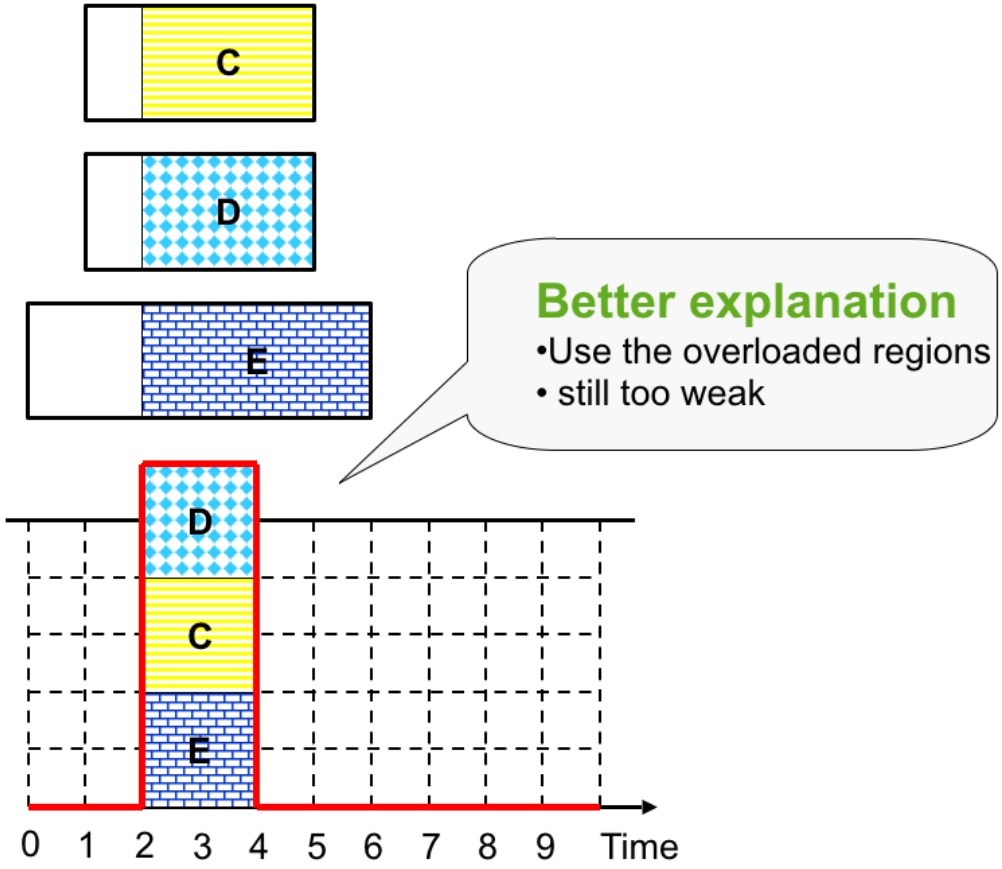
Explaining Failures

AI Institute for Advances in Optimization



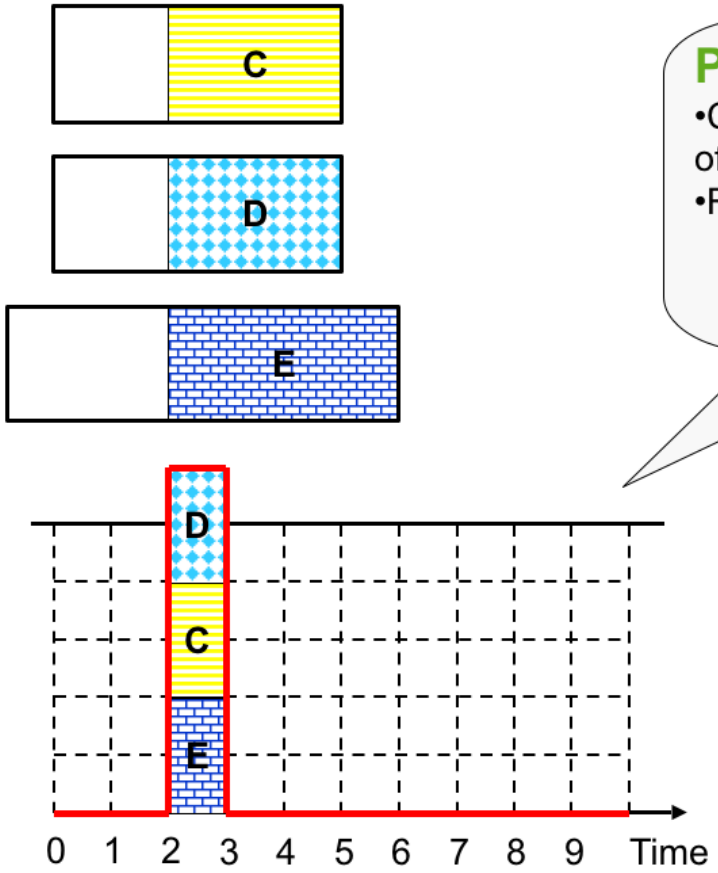
Explaining Failures

AI Institute for Advances in Optimization



Explaining Failures

AI Institute for Advances in Optimization

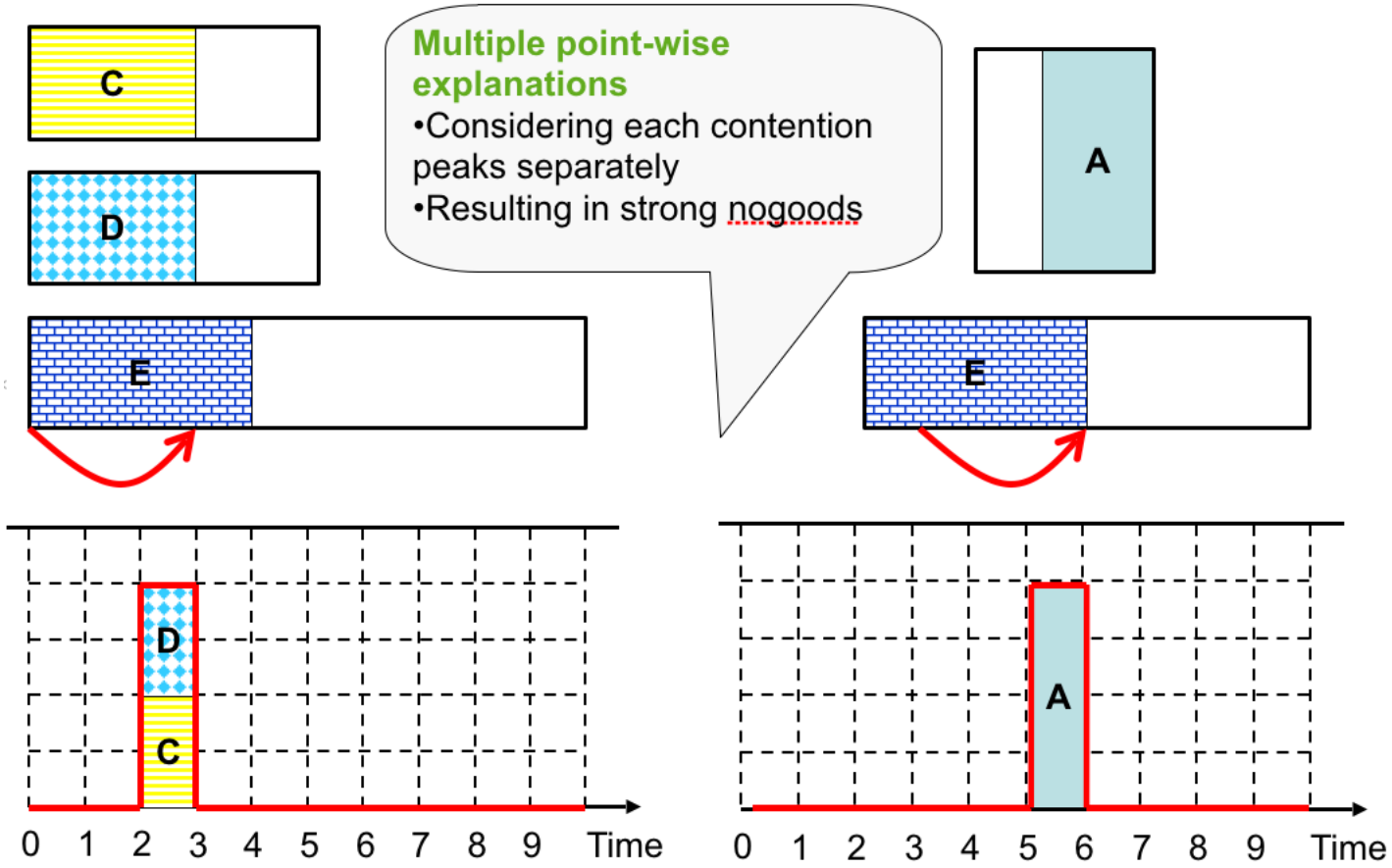


Point-wise explanation

- Consideration only single units of time in the resource overload
- Resulting in strong nogoods

Explaining Inferences

AI Institute for Advances in Optimization



The Value of Learning

AI Institute for Advances in Optimization

- Scheduling
  - Resource Constrained Project Scheduling Problems (RCPSP)
    - (probably) the most studied scheduling problems
    - closed 71 open problems
    - Solves more problems in 18s then previous state of the art in 1800s
  - RCPSP/Max (more complex precedence constraints)
    - closed 578 open instances of 631
    - recreates or betters all best known solutions by any method on 2340 instances except 3
  - RCPSP/DC (discounted cashflow)
    - Always finds solution on 19440 instances, optimal in all but 152 (versus 832 in previous state of the art)
    - the state of the art complete method for this problem

Scheduling in the Last Decade

AI Institute for Advances in Optimization


What's
NEXT

Outline

AI Institute for Advances in Optimization

- Brief Overview to constraint programming
- Scheduling with constraint programming
- Hybridizing constraint programming and MIP
- Learning-based constraint programming
- Learning Optimization Proxies

AI Institute for Advances in Optimization

PRELIMINARY PREPRINT VERSION: DO NOT CITE
The AAAI Digital Library will contain the published
version some time after the conference.

Fast Approximations for Job Shop Scheduling: A Lagrangian Dual Deep Learning Method

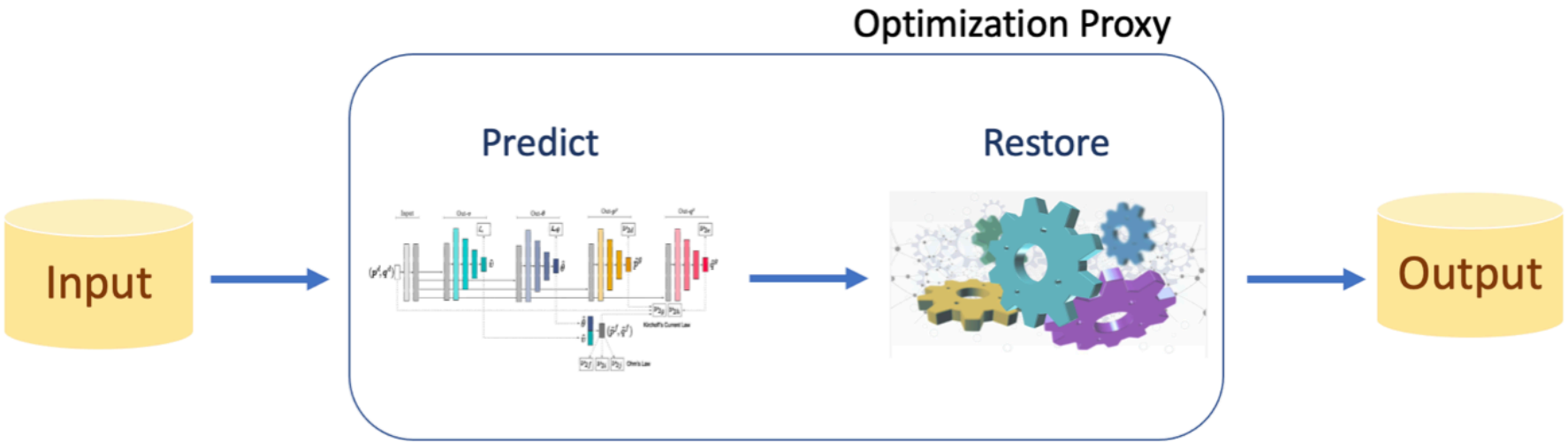
James Kotary,[1] **Ferdinando Fioretto,** [1] **Pascal Van Hentenryck** [2]

[1] Syracuse University
[2] Georgia Institute of Technology
jkotary@syr.edu, ffiorett@syr.edu, pvh@isye.gatech.edu

Optimization Proxies

AI Institute for Advances in Optimization

- Role of Optimization Proxies
  - learning the input/output relationship of an optimization problem



Optimal Power Flow

AI Institute for Advances in Optimization

Model 1 (AC Optimal Power Flow (AC-OPF)).

$$\mathcal{O}(\dot{\boldsymbol{p}}^d, \dot{\boldsymbol{q}}^d) = \arg\min_{\boldsymbol{p}^g,\, \boldsymbol{v}} \sum_{i\in\mathcal{N}} \text{cost}(p_i^g) \tag{1}$$

subject to

$$\dot{v}_i^{\min} \leqslant v_i \leqslant \dot{v}_i^{\max} \qquad \forall i \in \mathcal{N}, \tag{2a}$$

$$-\dot{\theta}_{ij}^{\Delta} \leqslant \theta_i - \theta_j \leqslant \dot{\theta}_{ij}^{\Delta} \qquad \forall (ij) \in \mathcal{E}; \tag{2b}$$

$$\dot{p}_i^{g\min} \leqslant p_i^g \leqslant \dot{p}_i^{g\max} \qquad \forall i \in \mathcal{N}, \tag{3a}$$

$$\dot{q}_i^{g\min} \leqslant q_i^g \leqslant \dot{q}_i^{g\max} \qquad \forall i \in \mathcal{N}; \tag{3b}$$

$$(p_{ij}^f)^2 + (q_{ij}^f)^2 \leqslant \dot{S}_{ij}^{f\max} \qquad \forall (ij) \in \mathcal{E}; \tag{4}$$

$$p_{ij}^f = \dot{g}_{ij} v_i^2 - v_i v_j (\dot{b}_{ij} \sin(\theta_i - \theta_j) + \dot{g}_{ij} \cos(\theta_i - \theta_j)) \qquad \forall (ij) \in \mathcal{E}; \tag{5a}$$

$$q_{ij}^f = -\dot{b}_{ij} v_i^2 - v_i v_j (\dot{g}_{ij} \sin(\theta_i - \theta_j) - \dot{b}_{ij} \cos(\theta_i - \theta_j)) \qquad \forall (ij) \in \mathcal{E}; \tag{5b}$$

$$p_i^g - \dot{p}_i^d = \sum_{(ij)\in\mathcal{E}} p_{ij}^f \qquad \forall i \in \mathcal{N}; \tag{6a}$$

$$q_i^g - \dot{q}_i^d = \sum_{(ij)\in\mathcal{E}} q_{ij}^f \qquad \forall i \in \mathcal{N}; \tag{6b}$$

output: $(\boldsymbol{p}^g, \boldsymbol{v})$ – The system operational parameters.

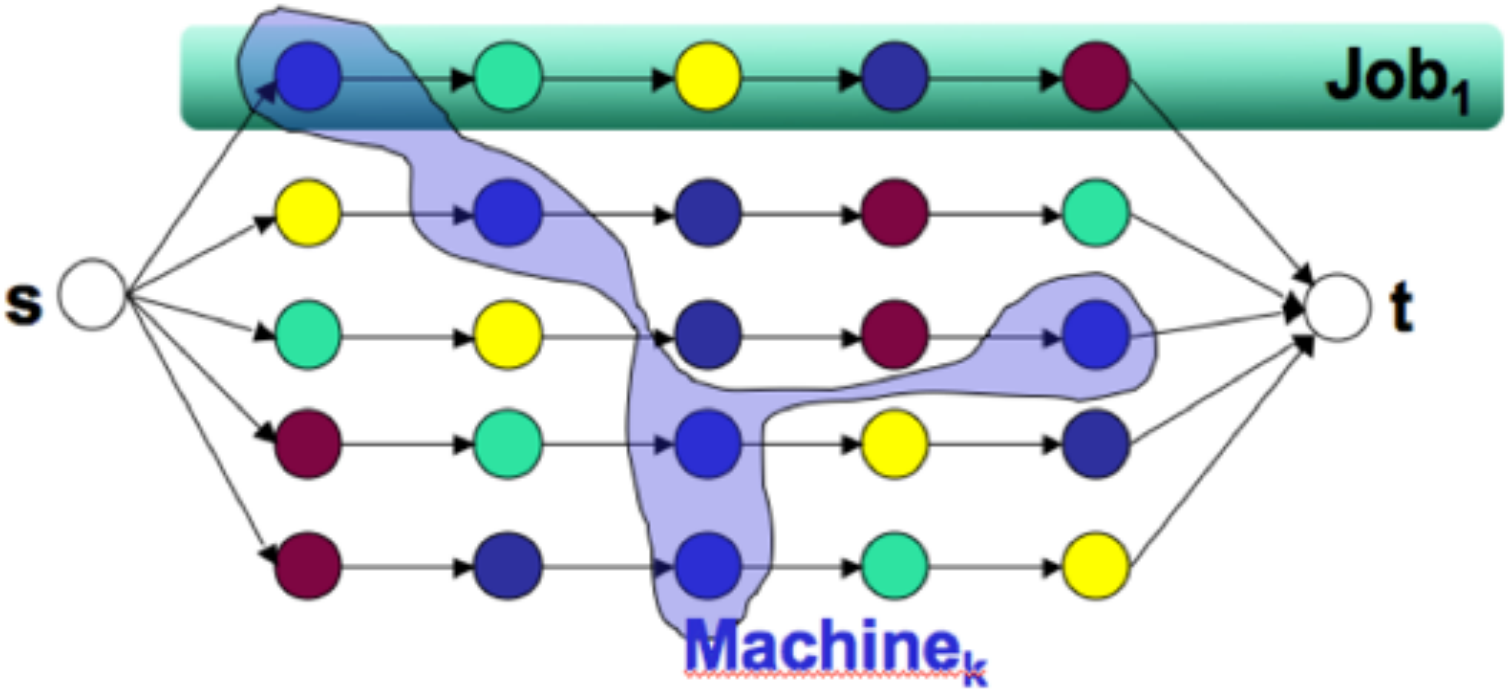
Learning Optimization Proxies

AI Institute for Advances in Optimization

- Challenges
  - physical, engineering, and business constraints
- Challenging learning problems
  - empire risk minimization under constraints
- Data augmentation
  - data is not an issue since we have an optimization models
- Nice synergies
  - between optimization and learning

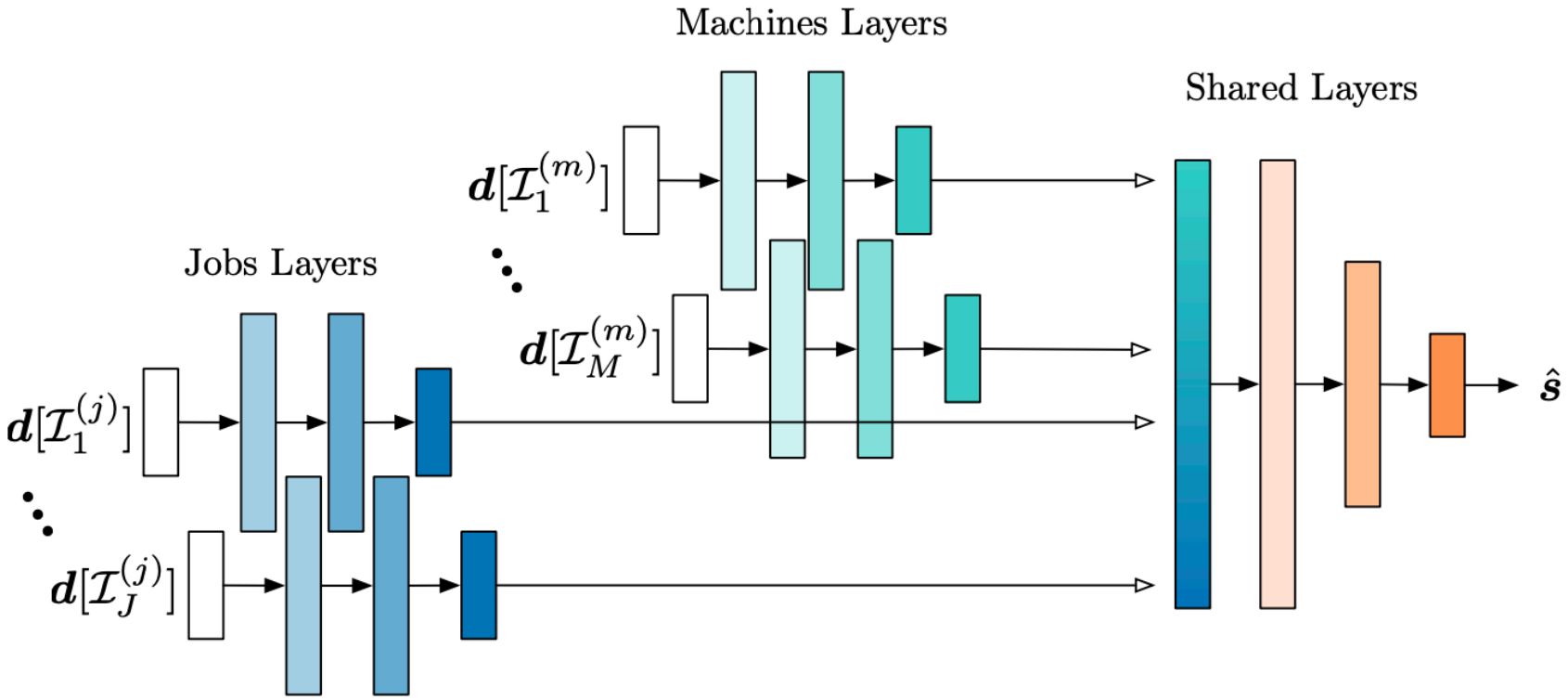
Scheduling Proxies

AI Institute for Advances in Optimization



Predict

AI Institute for Advances in Optimization



Repair

AI Institute for Advances in Optimization

Model 2: Recovering a Feasible Solution to the JSP.

$$\Pi(\boldsymbol{s}) = \text{argmin}_{\boldsymbol{s}} \; u$$

subject to: (2a), (2b)

$$s_t^j \geq s_{t'}^{j'} + d_{t'}^{j'} \quad \forall j, j' \in [J], t, t' \in [T] \text{ s.t. } (j,t) \preceq_{\hat{\boldsymbol{s}}} (j',t') \tag{8a}$$

$$s_t^j \geq 0 \quad \forall j \in [J], t \in [T] \tag{8b}$$

Algorithm 2: *The JSP Greedy Recovery.*

Input: $\{\hat{s}_t^j\}_{t\in[T],j\in[J]}$: predicted start times

$Q \leftarrow$ enqueue$((j,1)), \quad \forall j \in [J]$
while Q *is empty* **do**
  $(j,t) \leftarrow$ dequeue(Q)
  Schedule task (j,t) with start time $\hat{s}_t^j$
  if $t < T$ **then**
    $\hat{s}_{t+1}^j \leftarrow \max(\hat{s}_{t+1}^j, \hat{s}_t^j + d_t^j)$
    $Q \leftarrow$ enqueue(j,t)
  end
  foreach (t', j') *s.t.* $t \neq t'$ & $\sigma_{t'}^{j'} = \sigma_t^j$ & $\hat{s}_{t'}^{j'} \geq \hat{s}_t^j$ **do**
    $\hat{s}_{t'}^{j'} \leftarrow \max(\hat{s}_{t'}^{j'}, \hat{s}_t^j + d_t^j)$
  end
end

Experimental Setting

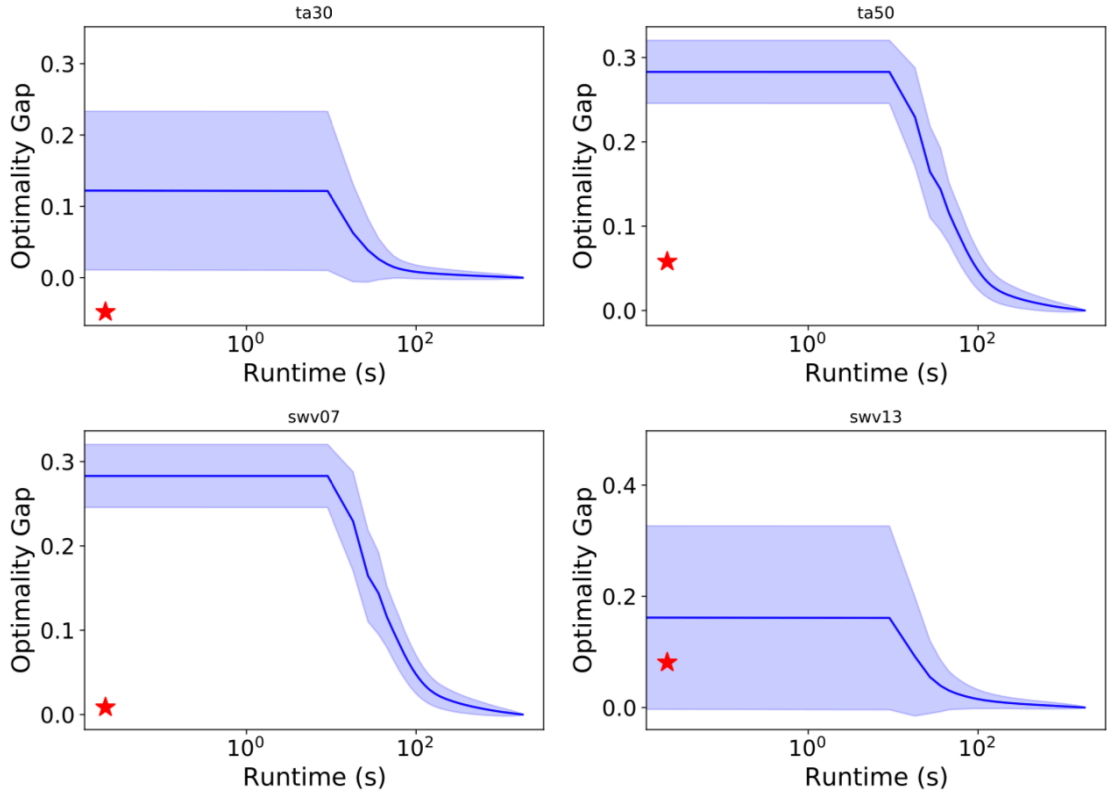
AI Institute for Advances in Optimization

- Slowdown of a machine
  - from 1% to 50%
- Training/Testing instances
  - 5,000
- Optimization solvers
  - CP Optimizer with 1800 seconds
- Heuristics
  - Shortest Processing Time (SPT), Least Work Remaining (LWR), Most Work Remaining (MWR), Least Operations Remaining (LOR), and Most Operations Remaining (MOR)

Learning Scheduling Problems

AI Institute for Advances in Optimization

- Comparing SOTA solvers with deep learning + feasibility restoration



Learning Scheduling Problems

AI Institute for Advances in Optimization

- Comparing SOTA solvers with deep learning + feasibility restoration

Instance	Size	Prediction Err(×10) ↓		Constraint Viol(×10^2) ↓		Opt. Gap Heuristics (%) ↓					Opt. Gap DNNs (%) ↓		Time SoTA Eq. (s) ↑	
	$J \times M$	FC	JSP-DNN	FC	JSP-DNN	SPT	LWR	MWR	LOR	MOR	FC	JSP-DNN	FC	JSP-DNN
yn02	20×20	2.770	**0.138**	1.134	**0.122**	628	837	40	934	40	12.80 ± 5.4	**-0.045 ± 0.9**	10.20	**1800+**
ta25	20×20	1.607	**0.361**	0.631	**0.244**	593	877	59	787	46	13.61 ± 3.13	**-0.143 ± 0.8**	11.02	**1800+**
ta30	30×15	4.338	**1.196**	1.483	**0.357**	558	910	63	856	46	15.01 ± 2.63	**-0.48 ± 5.18**	9.06	**1800+**
ta40	30×20	7.880	**3.341**	1.863	**0.104**	492	794	57	836	25	23.11 ± 7.33	**3.19 ± 1.88**	8.40	**12.04**
ta50	50×10	4.580	**1.322**	1.223	**0.225**	789	789	53	1116	43	18.30 ± 5.22	**5.85± 2.72**	8.02	**90.30**
swv03	20×15	9.473	**2.683**	2.777	**0.850**	203	212	75	190	50	28.61 ± 14.27	**7.62 ± 2.51**	4.04	**36.36**
swv05	20×10	6.586	**2.950**	2.325	**0.626**	183	192	80	177	66	20.78 ± 10.54	**6.34 ± 1.82**	7.24	**18.18**
swv07	20×10	4.587	**0.681**	1.222	**0.223**	299	295	68	352	43	10.69 ± 6.83	**0.01± 4.75**	26.0	**254.5**
swv09	20×15	5.678	**3.462**	2.132	**0.211**	322	270	69	285	75	22.12 ± 8.52	**5.42 ± 1.21**	6.48	**28.32**
swv11	50×10	7.958	**3.244**	2.711	**0.282**	237	231	94	263	73	23.18 ± 2.27	**4.80 ± 4.47**	7.02	**92.00**
swv13	50×10	23.21	**3.557**	1.615	**0.323**	225	203	114	218	79	22.79 ± 16.21	**8.11 ± 4.20**	7.08	**24.08**

Table 2: Accuracy metrics compared between FC and JSP-DNN (left sub-table) and accuracy of simple heuristics vs CP-Optimizer at 1800s (right sub-table). Best results shown in bold.

Outline

AI Institute for Advances in Optimization

- Brief Overview to constraint programming
- Scheduling with constraint programming
- Hybridizing constraint programming and MIP
- Learning-based constraint programming
- Learning Optimization Proxies

Thank you for listening

AI Institute for Advances in Optimization