AAAI-18 Tutorial on:

MULTI-AGENT DISTRIBUTED CONSTRAINED OPTIMIZATION

Ferdinando Fioretto
University of Michigan

William Yeoh
Washington University in St. Louis

Roie Zivan
Ben-Gurion University of the Negev

SCHEDULE

- 11:20am: Preliminaries
- 11:40am: DCOP Algorithms
- 12:20pm: DCOP Extensions
- 12:30pm: Applications
- 12:50pm: Challenges and Open Questions
- 1:00pm: Done! Lunch? :)

LIL' BIT OF SHAMELESS PROMOTION :)

- Tutorial materials are based on our recent JAIR survey paper:

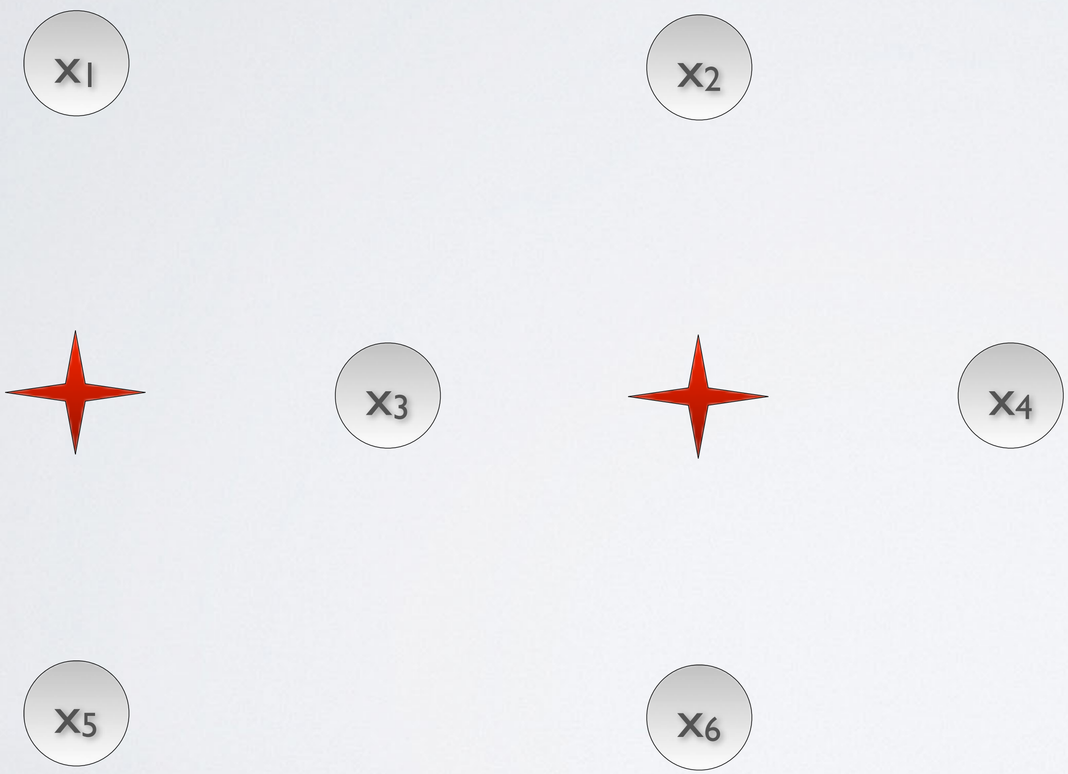
  Ferdinando Fioretto, Enrico Pontelli, and William Yeoh.
  Distributed Constraint Optimization Problems and Applications: A Survey.
  Journal of Artificial Intelligence Research (JAIR), to appear, 2018.

- Includes more models, algorithms, and applications.
- Also available on arXiv.

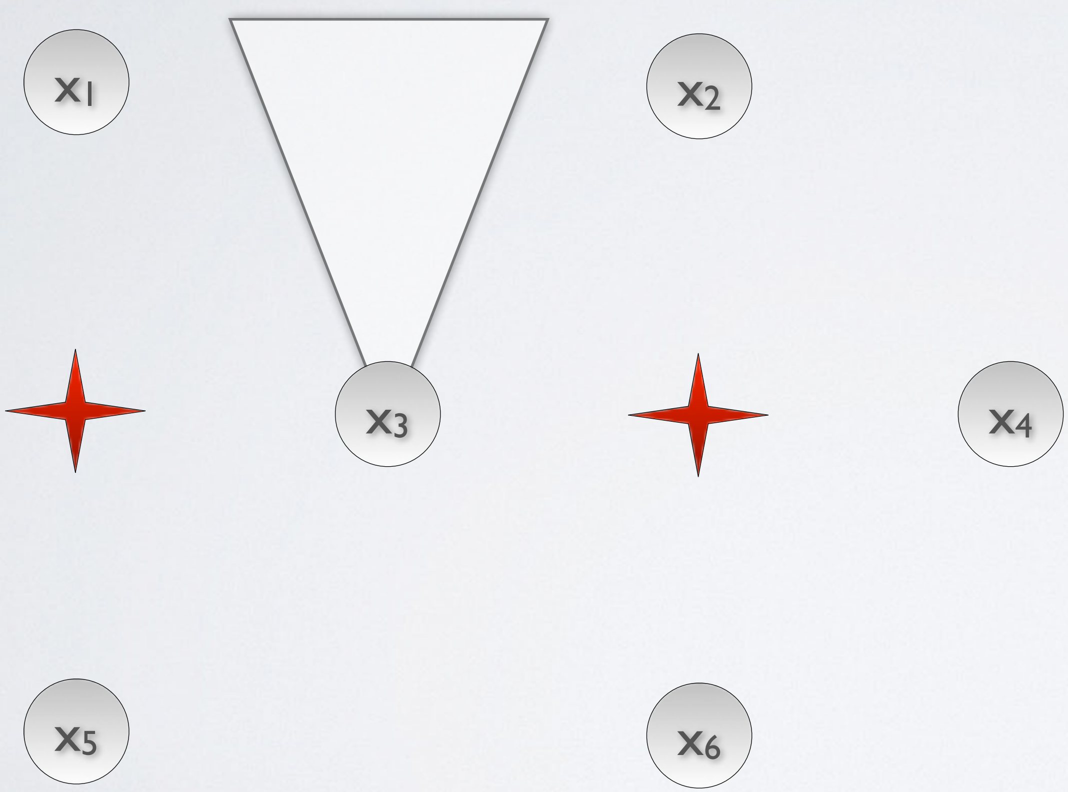
PRELIMINARIES

AAAI-18 Tutorial on
Multi-Agent Distributed Constrained Optimization

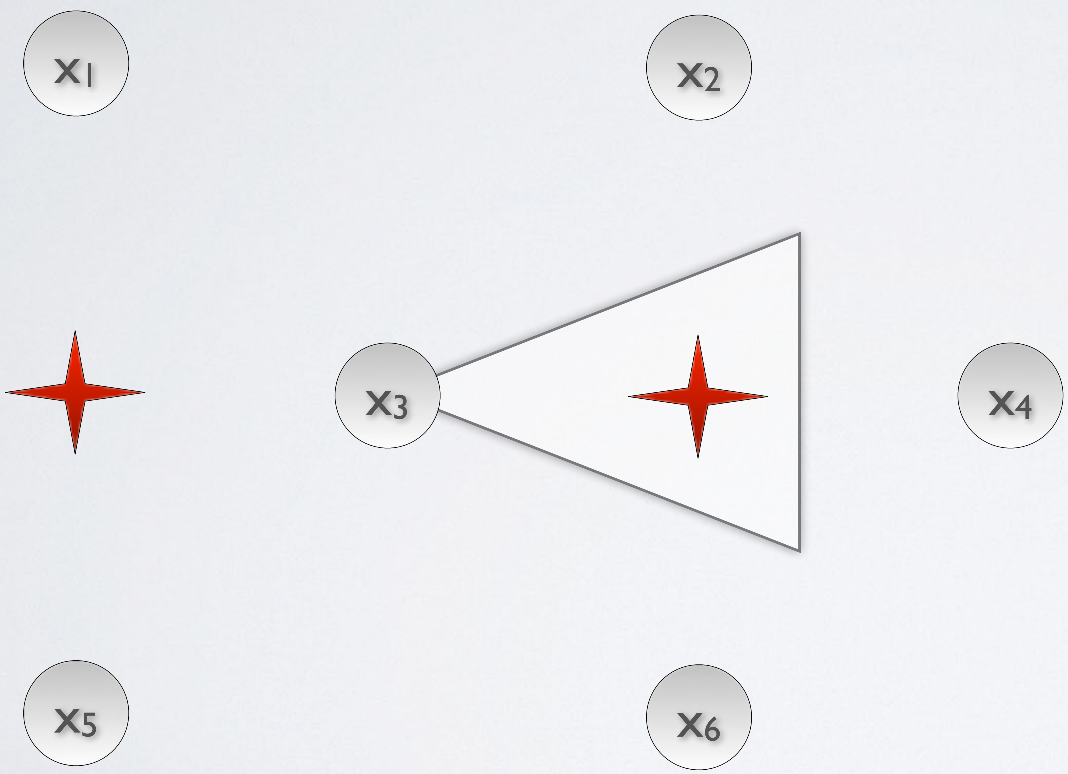
MOTIVATING DOMAIN: SENSOR NETWORK



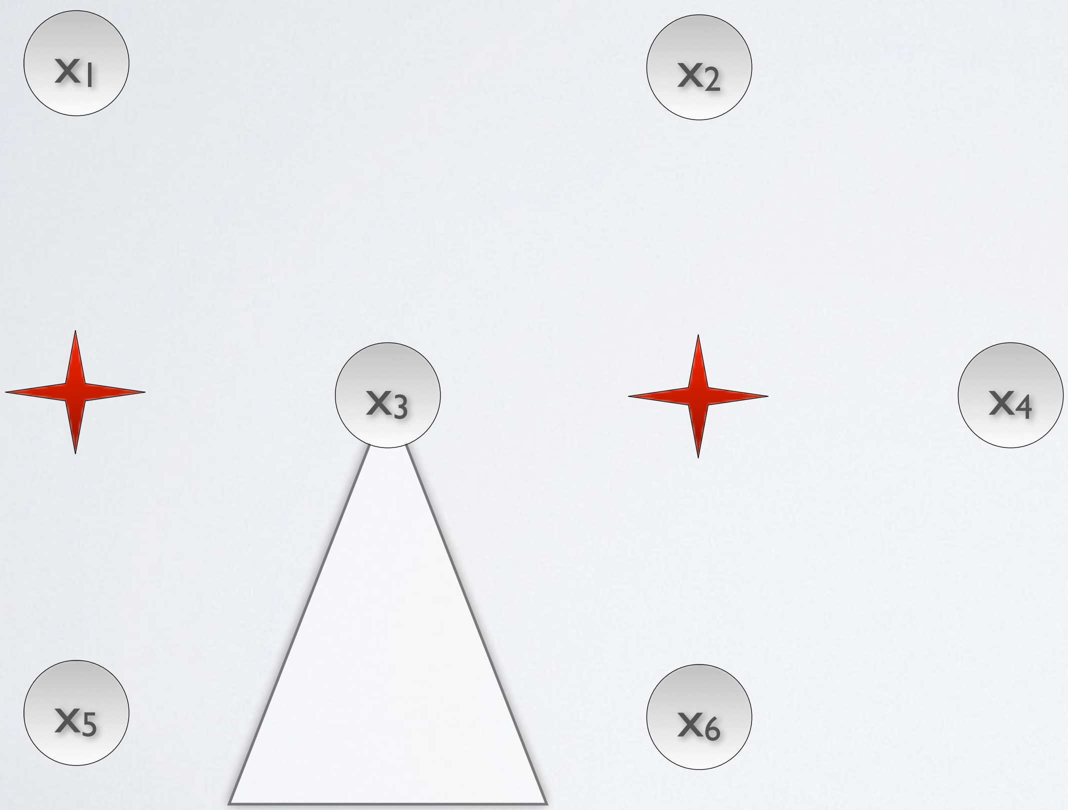
MOTIVATING DOMAIN: SENSOR NETWORK



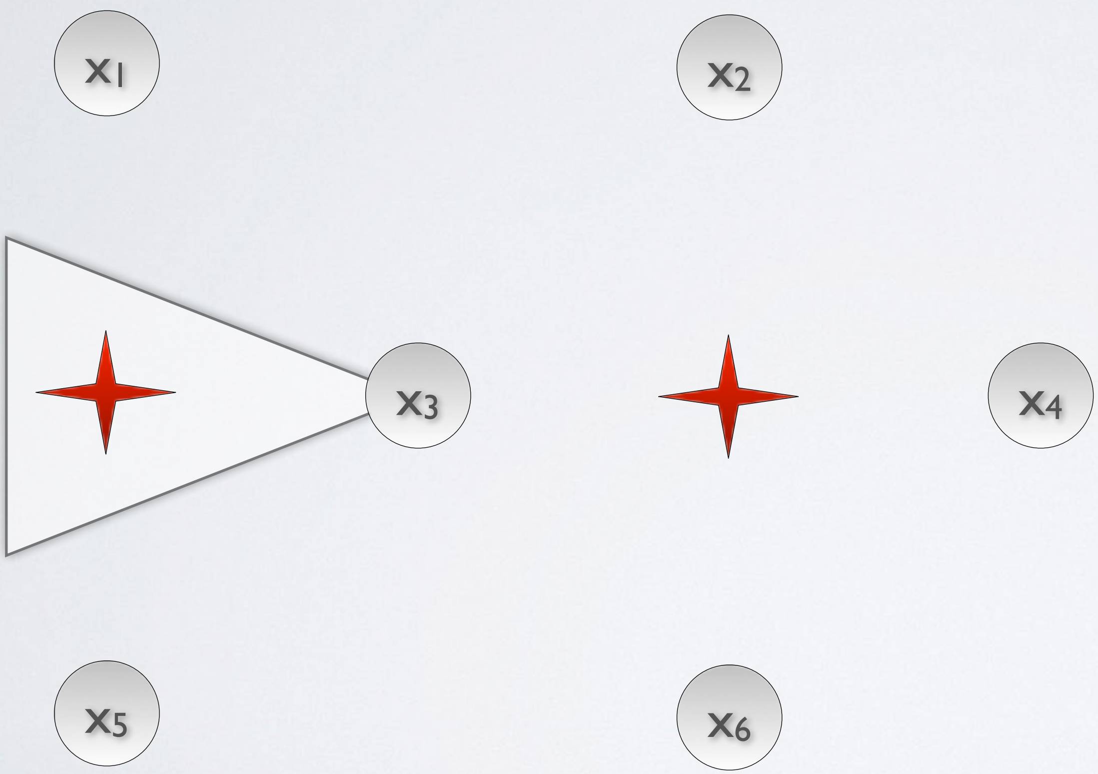
MOTIVATING DOMAIN: SENSOR NETWORK



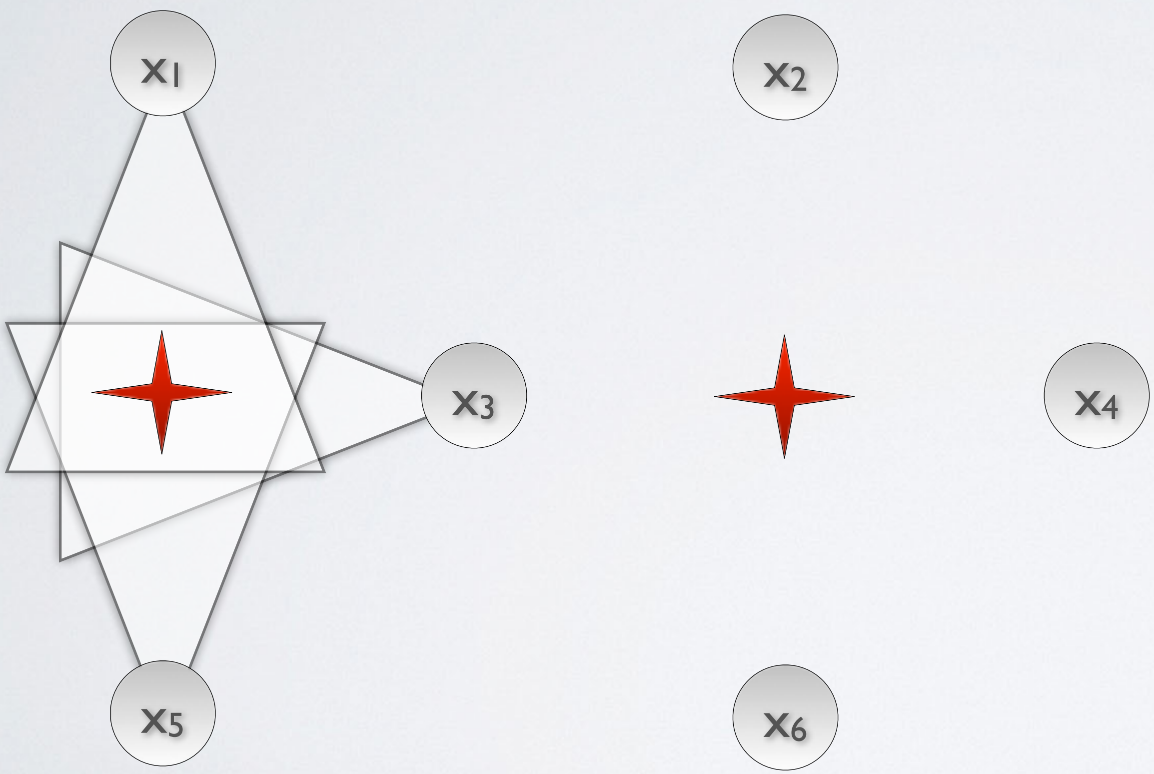
MOTIVATING DOMAIN: SENSOR NETWORK



MOTIVATING DOMAIN: SENSOR NETWORK



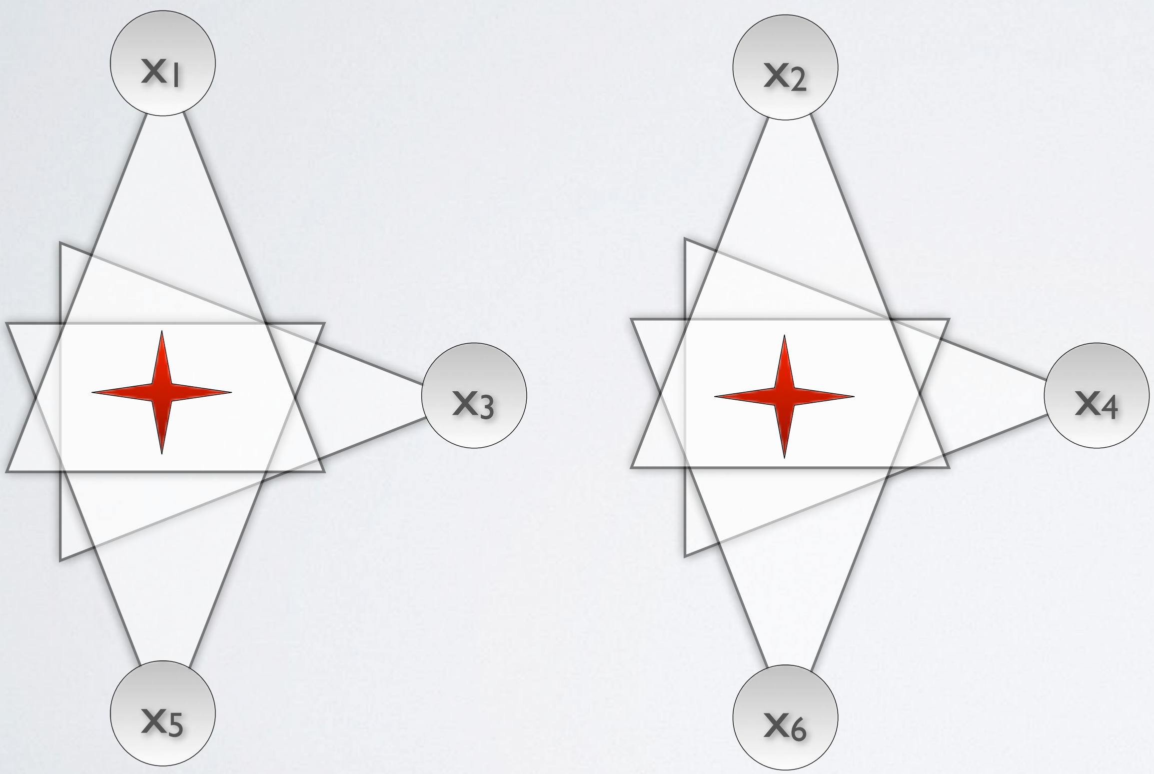
MOTIVATING DOMAIN: SENSOR NETWORK



x_1	x_3	x_5	Sat?
N	N	N	X
N	N	E	X
	…		X
S	**W**	**N**	✓
	…		X
W	W	W	X

Model the problem as a CSP

MOTIVATING DOMAIN: SENSOR NETWORK



x_1	x_3	x_5	Sat?
N	N	N	X
N	N	E	X
	…		X
S	**W**	**N**	✓
	…		X
W	W	W	X

Model the problem as a CSP

CSP

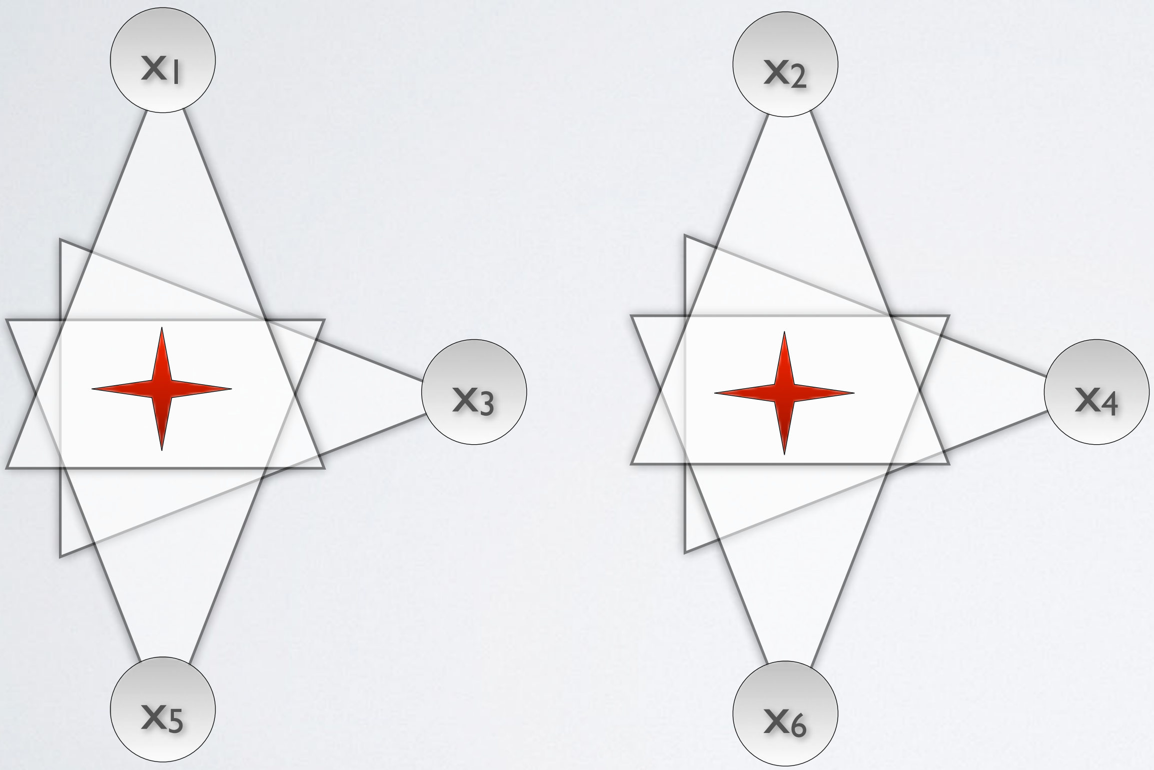
CONSTRAINT SATISFACTION

- Variables $X = \{x_1, \ldots, x_n\}$
- Domains $D = \{D_1, \ldots, D_n\}$
- Constraints $C = \{c_1, \ldots, c_m\}$
  where a constraint $c_i \subseteq D_{i_1} \times D_{i_2} \times \ldots \times D_{i_n}$
  denotes the possible valid joint assignments for the variables $x_{i_1}, x_{i_2}, \ldots, x_{i_n}$ it involves

- **GOAL**: Find an assignment to all variables that satisfies all the constraints

CSP

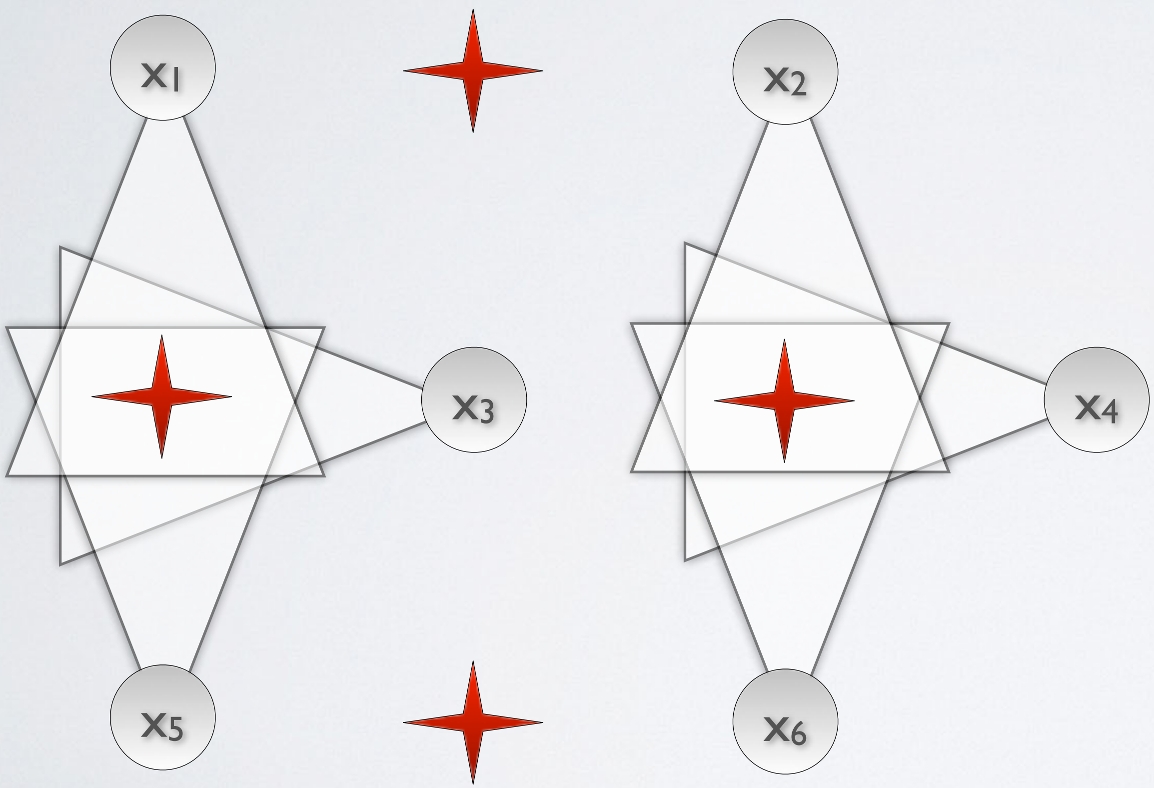
CONSTRAINT SATISFACTION



x_1	x_3	x_5	Sat?
N	N	N	X
N	N	E	X
	…		X
S	**W**	**N**	✓
	…		X
W	W	W	X

Model the problem as a CSP

MAX-CSP
MAX CONSTRAINT SATISFACTION



x_1	x_3	x_5	Sat?
N	*N*	*N*	X
N	*N*	*E*	X
…			X
S	***W***	***N***	✓
…			X
W	*W*	*W*	X

Model the problem as a Max-CSP

MAX-CSP

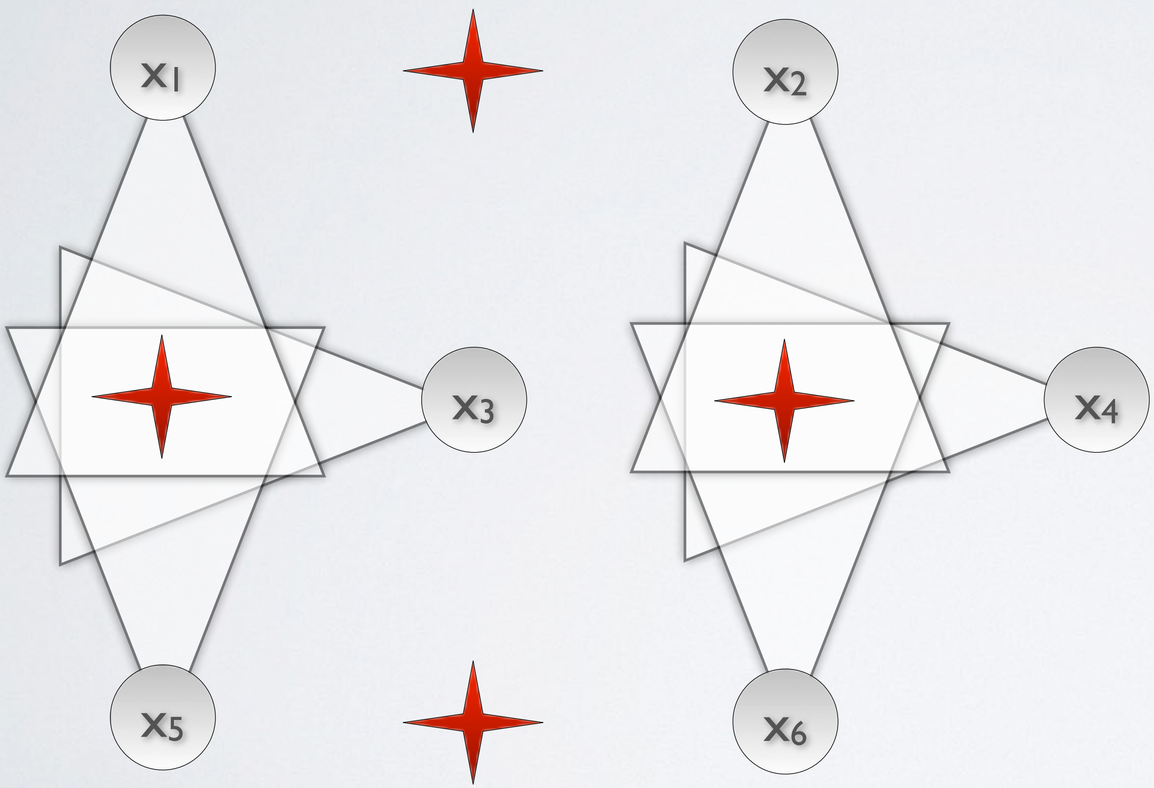
MAX CONSTRAINT SATISFACTION

- Variables $X = \{x_1, \ldots, x_n\}$
- Domains $D = \{D_1, \ldots, D_n\}$
- Constraints $C = \{c_1, \ldots, c_m\}$
  where a constraint $c_i \subseteq D_{i_1} \times D_{i_2} \times \ldots \times D_{i_n}$
  denotes the possible valid joint assignments for the variables $x_{i_1}, x_{i_2}, \ldots, x_{i_n}$ it involves

- **GOAL**: Find an assignment to all variables that satisfies a maximum number of constraints

MAX-CSP

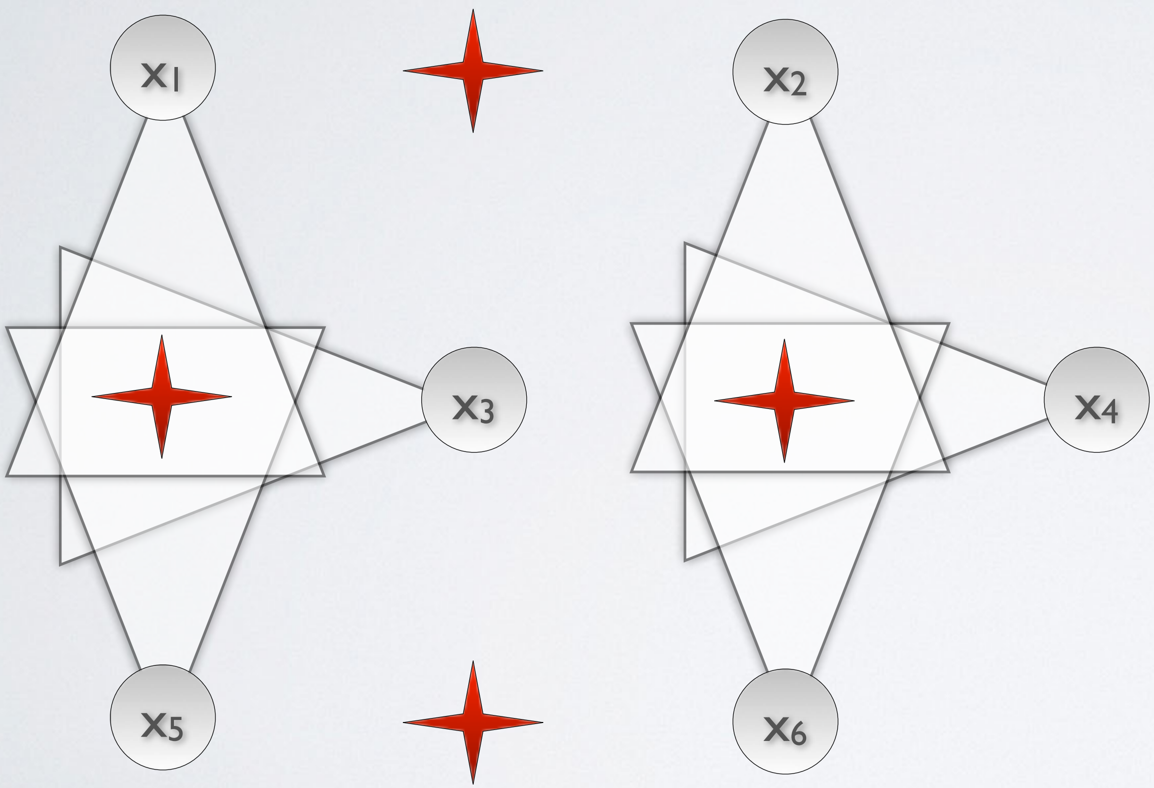
MAX CONSTRAINT SATISFACTION



x_1	x_3	x_5	Sat?
N	*N*	*N*	X
N	*N*	*E*	X
	...		X
S	***W***	***N***	✓
	...		X
W	*W*	*W*	X

Model the problem as a Max-CSP

WCSP (COP)
CONSTRAINT OPTIMIZATION



x1	x3	x5	Cost
N	*N*	*N*	∞
N	*N*	*E*	∞
	…		∞
S	**W**	**N**	**10**
	…		∞
W	*W*	*W*	∞

Model the problem as a COP

WCSP (COP)
CONSTRAINT OPTIMIZATION

- Variables $X = \{x_1, \ldots, x_n\}$
- Domains $D = \{D_1, \ldots, D_n\}$
- Constraints $C = \{c_1, \ldots, c_m\}$
  where a constraint $c_i : D_{i_1} \times \ldots \times D_{i_n} \rightarrow \mathbb{R}_+ \cup \{\infty\}$
  expresses the degree of constraint violation

- **GOAL**: Find an assignment that minimizes the sum of the costs of all the constraints

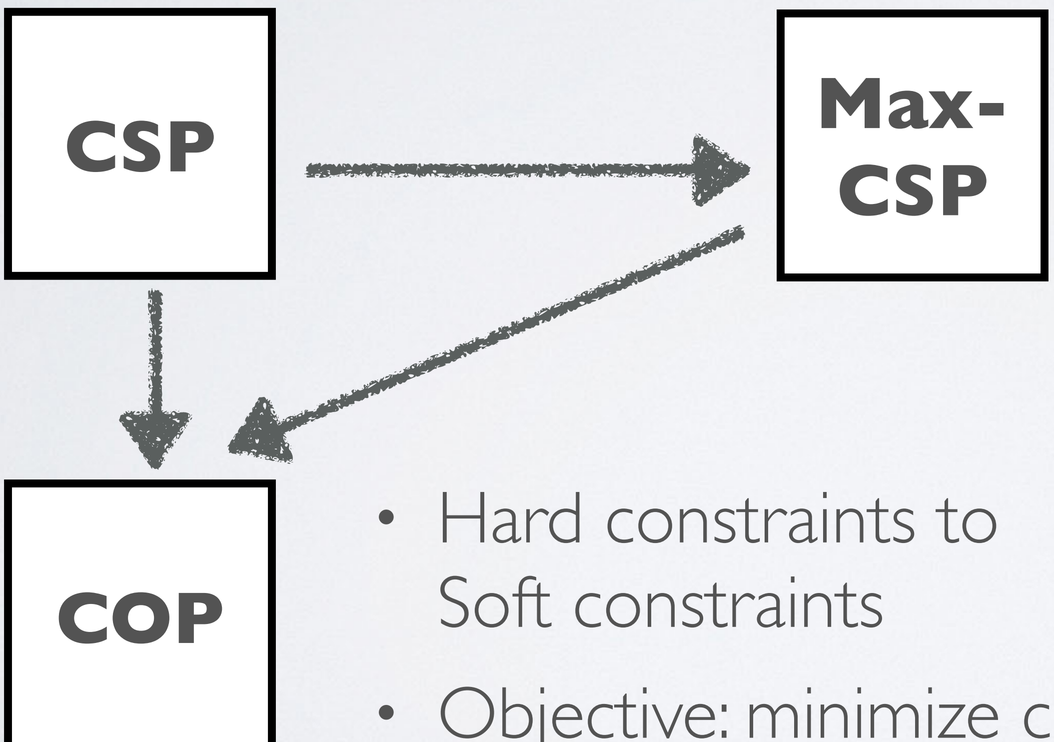
WCSP (COP)

CONSTRAINT OPTIMIZATION

CSP → **Max-CSP**

- Objective: maximize #constraints satisfied

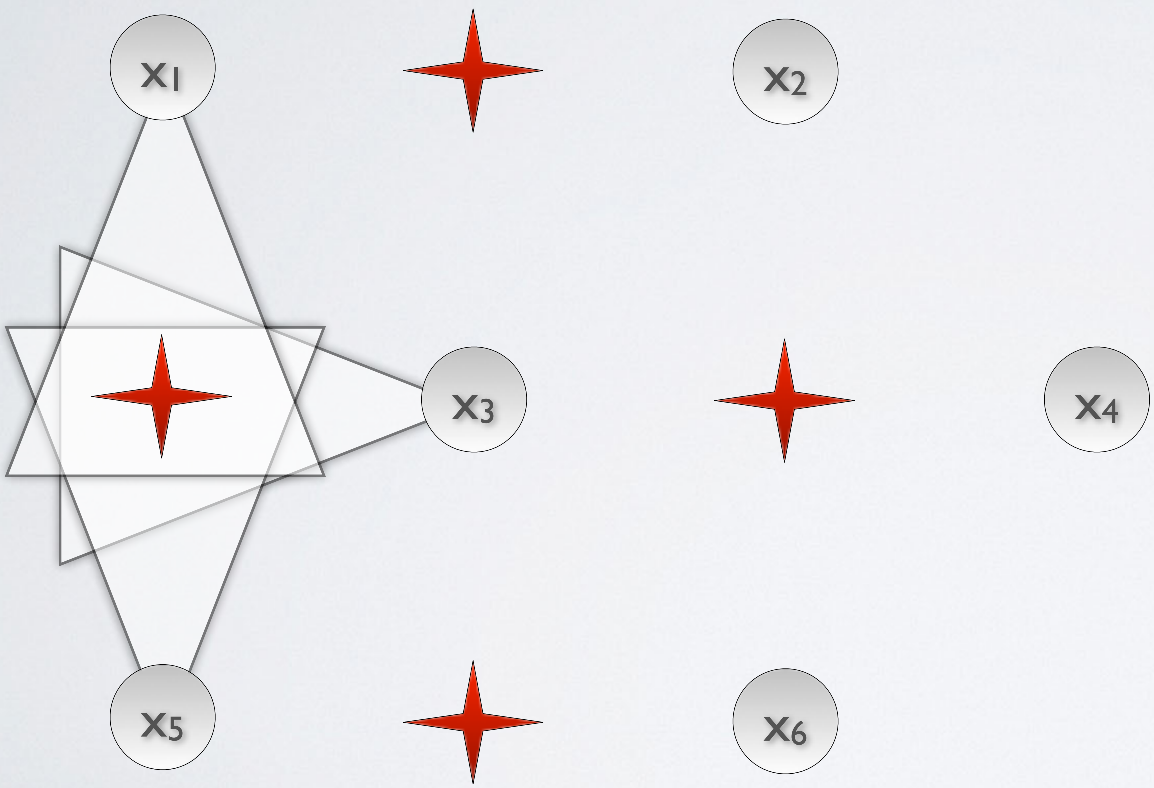
WCSP (COP)
CONSTRAINT OPTIMIZATION



- Objective: maximize #constraints satisfied

- Hard constraints to Soft constraints
- Objective: minimize cost

WCSP (COP)
CONSTRAINT OPTIMIZATION



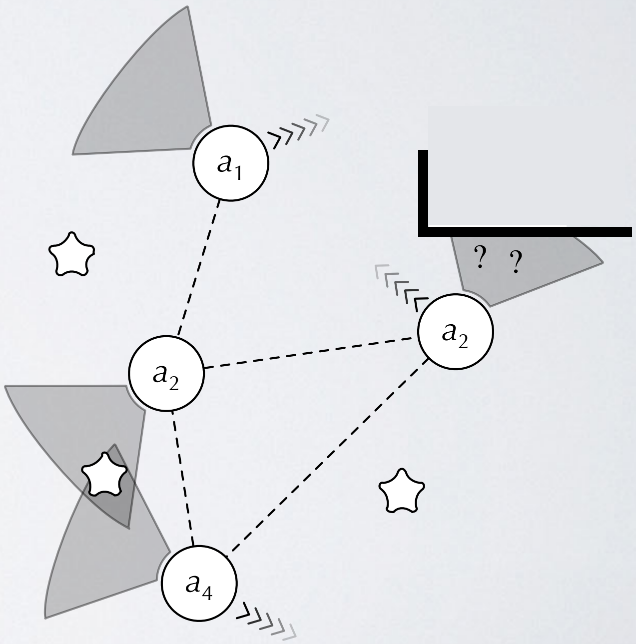
Imagine that each sensor is an autonomous agent.

How should this problem be modeled and solved in a decentralized manner?

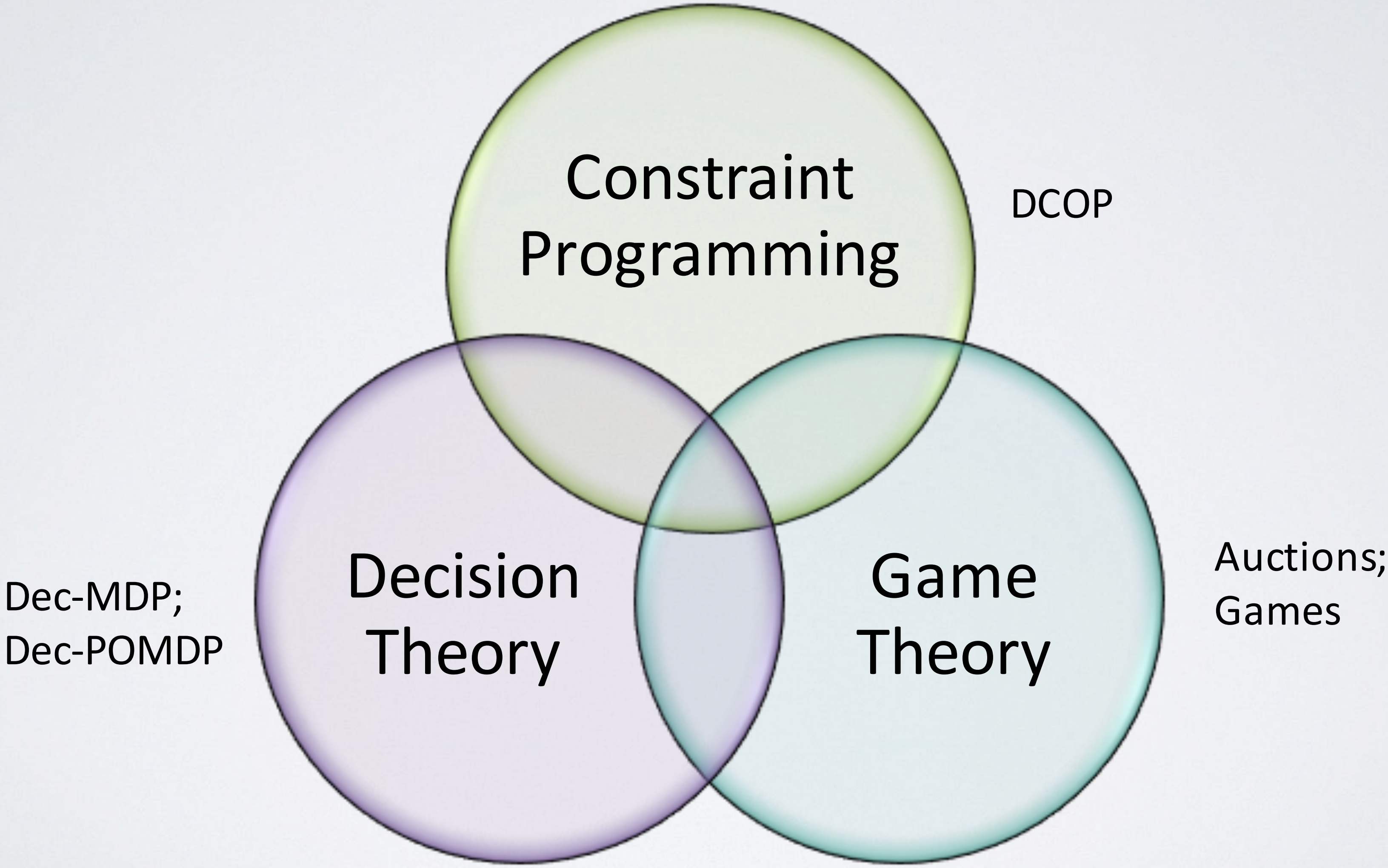
MULTI-AGENT SYSTEMS

- **Agent**: An entity that behaves autonomously in the pursuit of goals
- **Multi-agent system**: A system of multiple interacting agents

- An agent is:
  - *Autonomous*: Is of full control of itself
  - *Interactive*: May communicate with other agents
  - *Reactive*: Responds to changes in the environment or requests by other agents
  - *Proactive*: Takes initiatives to achieve its goals

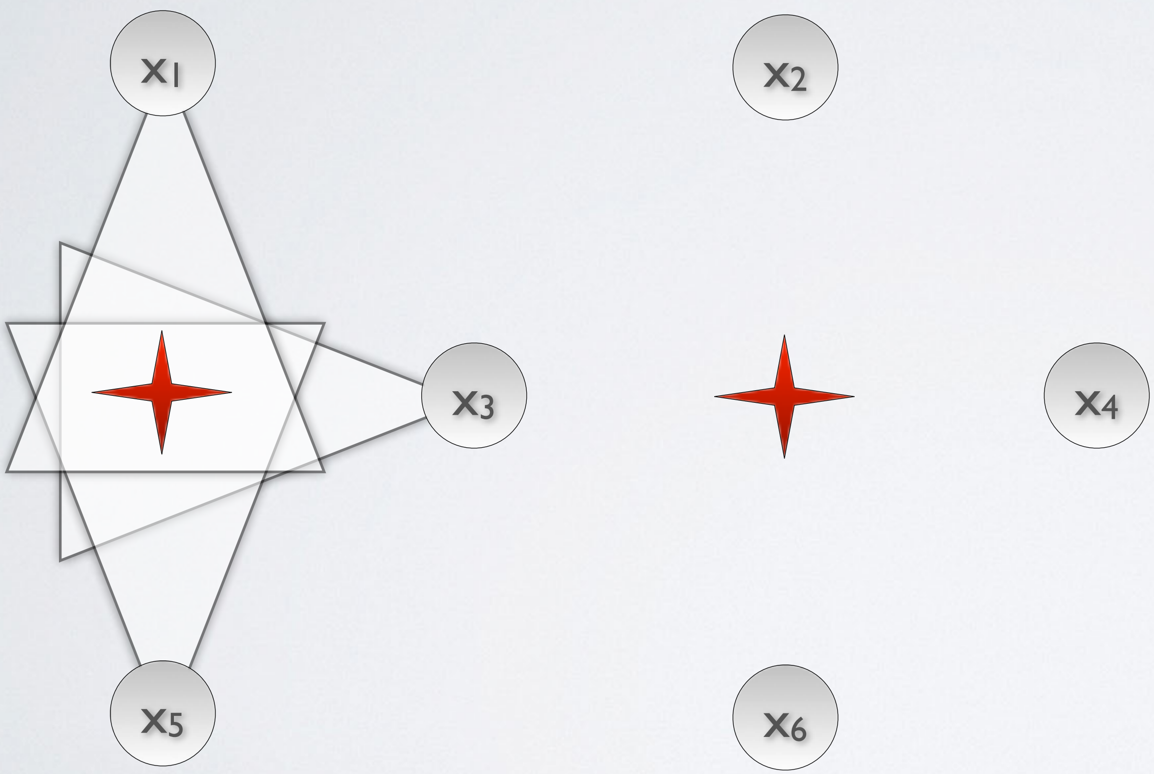


MULTI-AGENT SYSTEMS



DCOP

DISTRIBUTED CONSTRAINT OPTIMIZATION

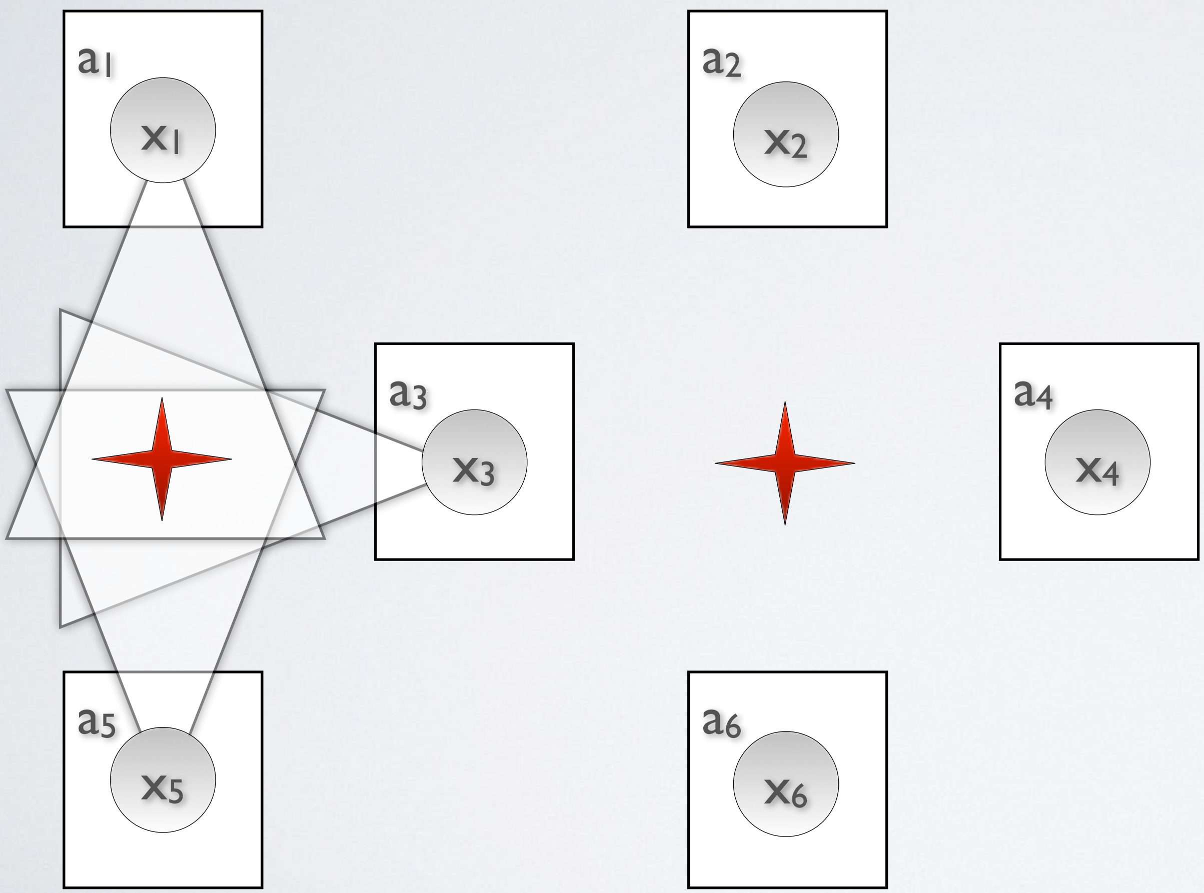


Imagine that each sensor is an autonomous agent.

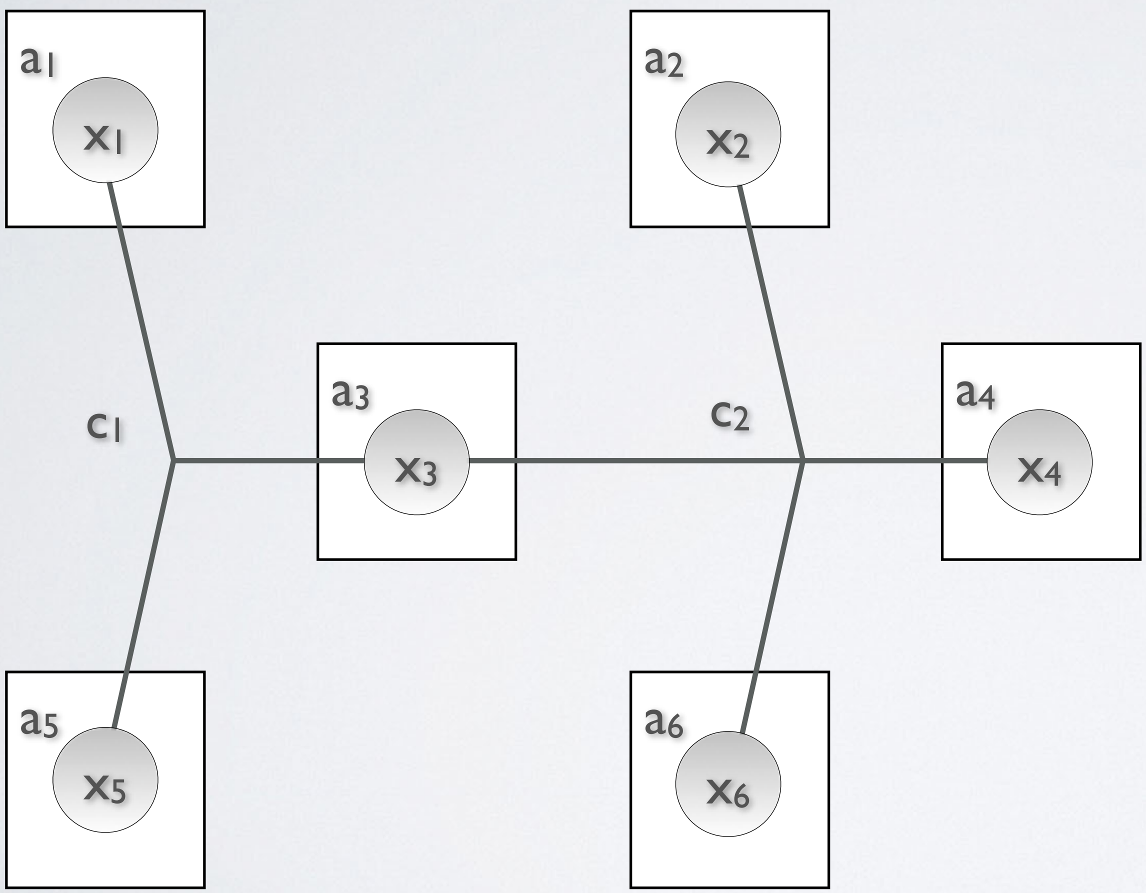
How should this problem be modeled and solved in a decentralized manner?

DCOP

DISTRIBUTED CONSTRAINT OPTIMIZATION



DCOP
DISTRIBUTED CONSTRAINT OPTIMIZATION



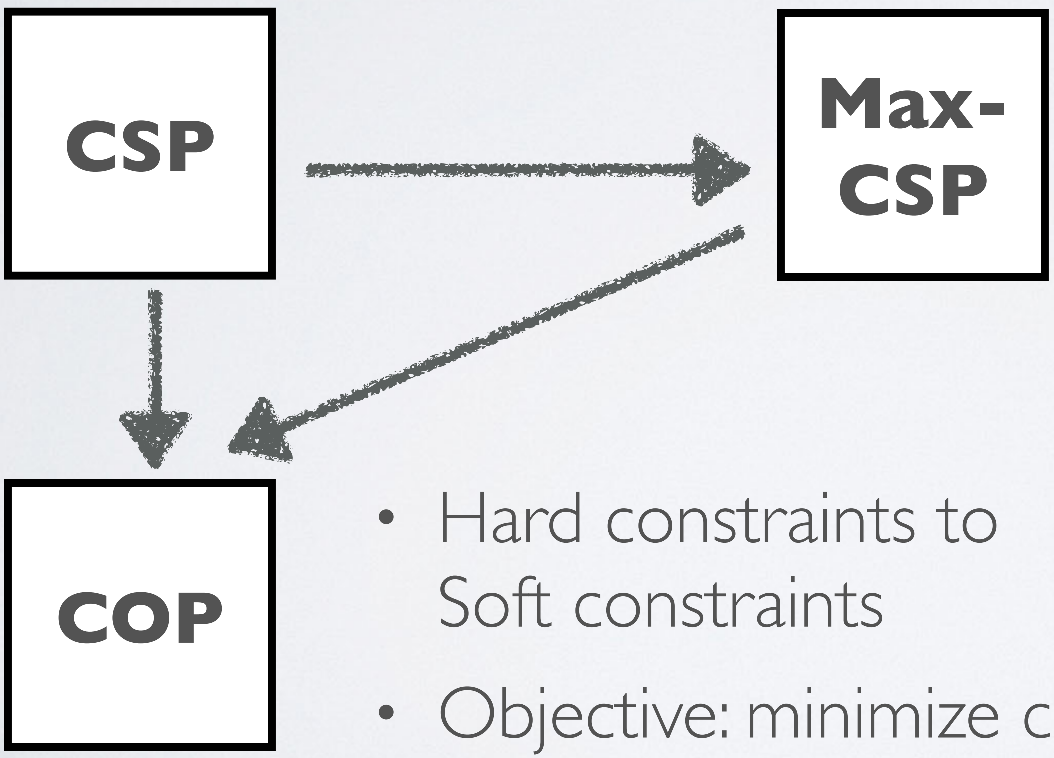
DCOP

DISTRIBUTED CONSTRAINT OPTIMIZATION

- Agents $A = \{a_i, \ldots, a_n\}$
- Variables $X = \{x_1, \ldots, x_n\}$
- Domains $D = \{D_1, \ldots, D_n\}$
- Constraints $C = \{c_1, \ldots, c_m\}$
- Mapping of variables to agents

- **GOAL**: Find an assignment that minimizes the sum of the costs of all the constraints

DCOP
DISTRIBUTED CONSTRAINT OPTIMIZATION

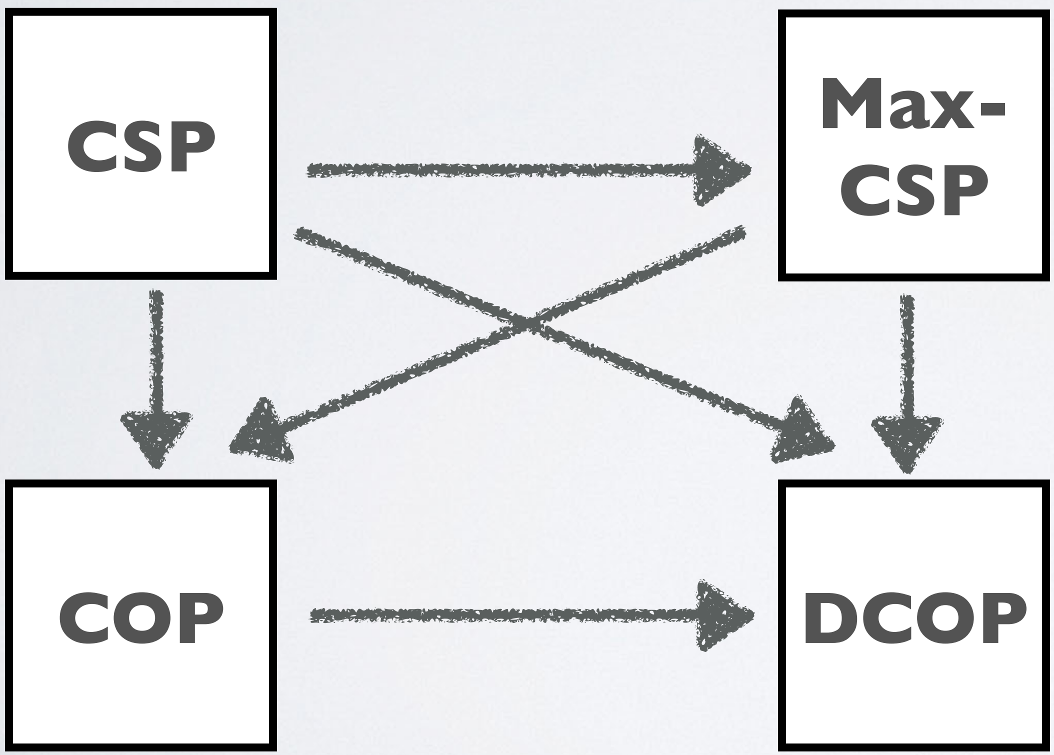


- Objective: maximize #constraints satisfied

- Hard constraints to Soft constraints
- Objective: minimize cost

DCOP

DISTRIBUTED CONSTRAINT OPTIMIZATION



- Variables are controlled by agents
- Communication model
- Local agents' knowledge

DCOP

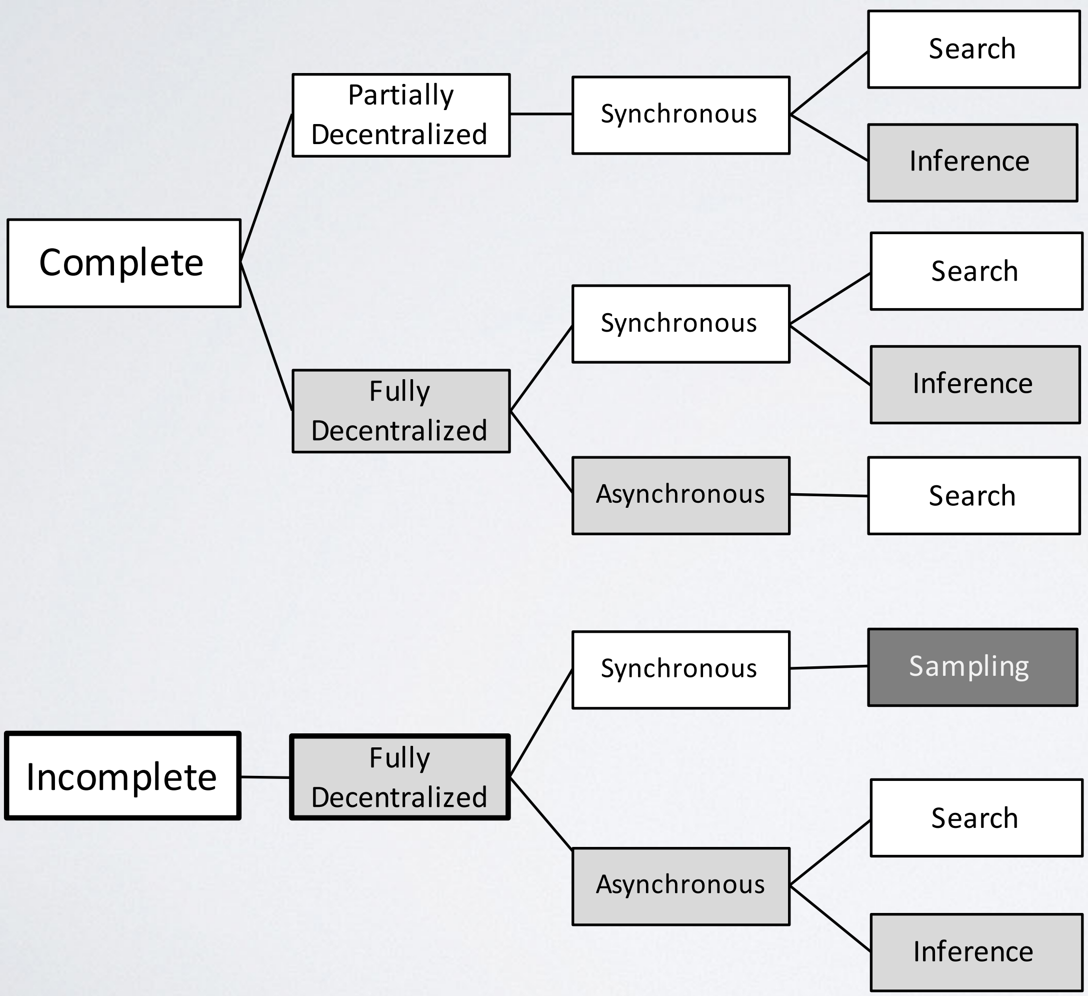
DISTRIBUTED CONSTRAINT OPTIMIZATION

- Why distributed models?
  - Natural mapping for multi-agent systems
  - Potentially faster by exploiting parallelism
  - Potentially more robust: no single point of failure, no single network bottleneck
  - Maintains more private information
  - ...

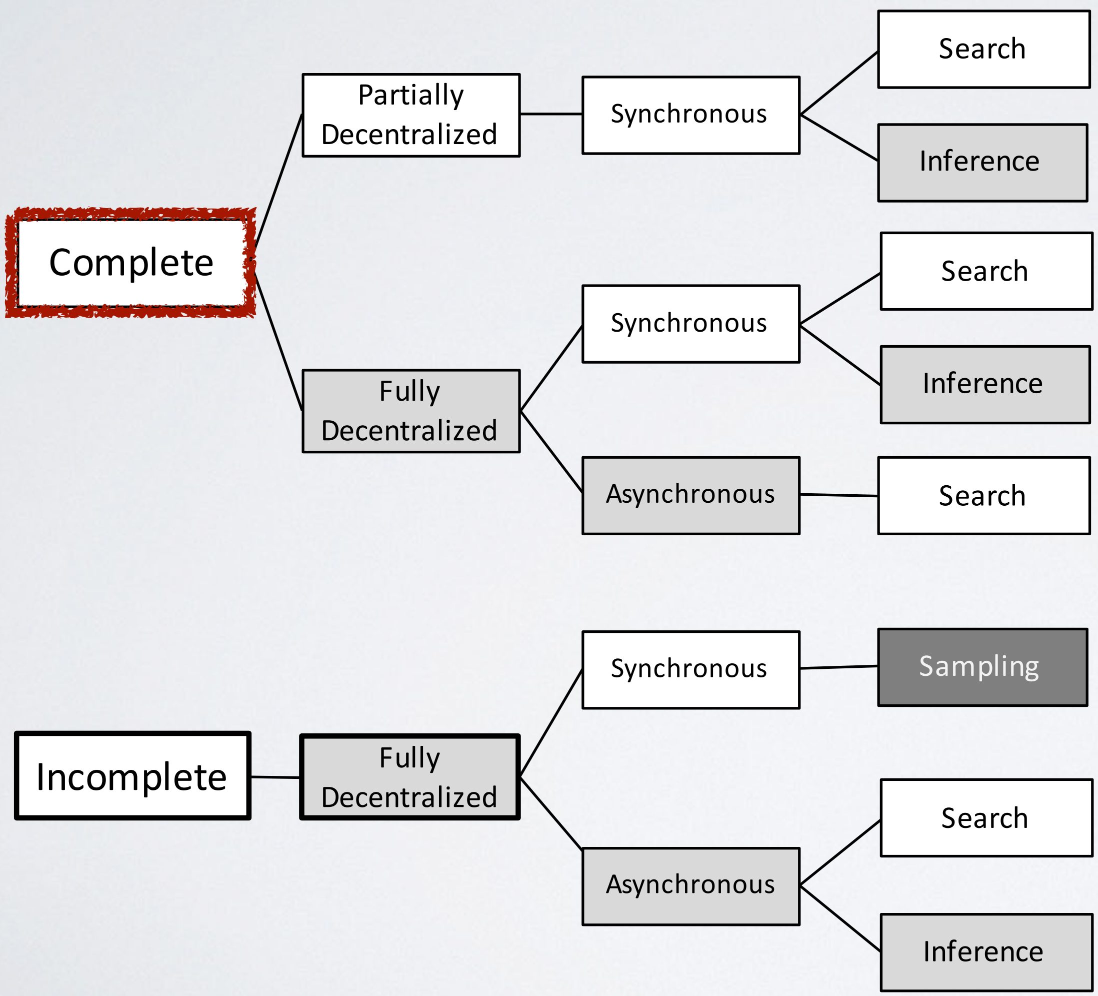
DCOP ALGORITHMS

AAAI-18 Tutorial on
Multi-Agent Distributed Constrained Optimization

DCOP ALGORITHMS

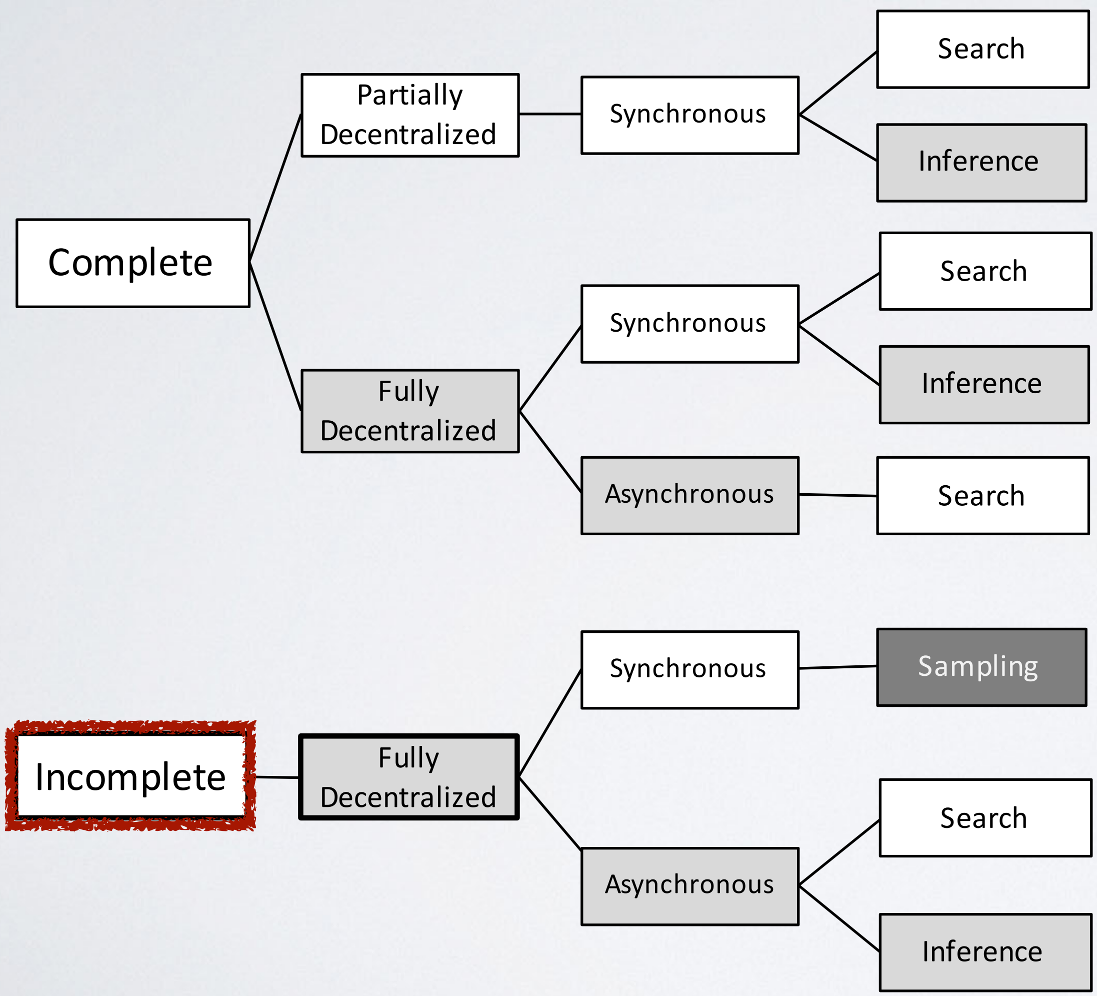


DCOP ALGORITHMS



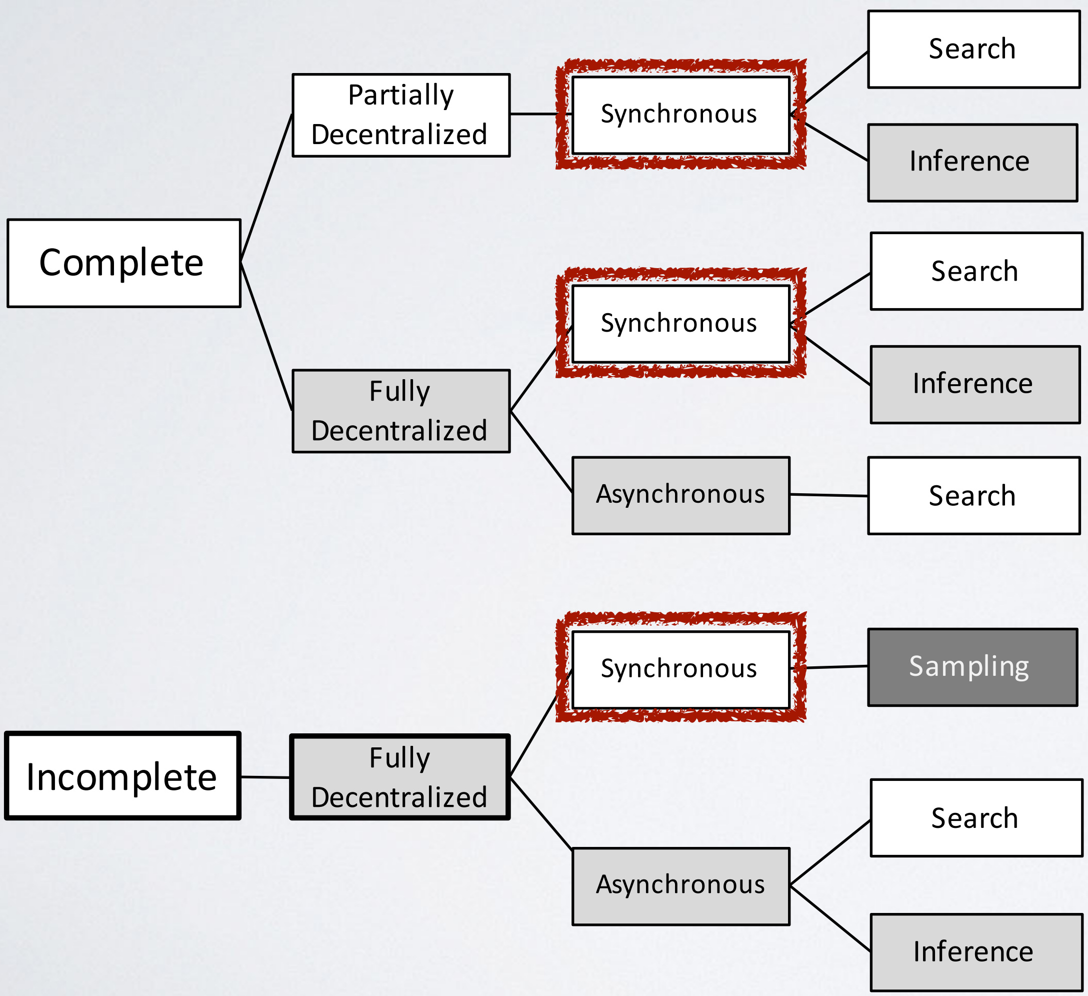
- Important Metrics:
  - Agent complexity
  - Network loads
  - Message size

DCOP ALGORITHMS



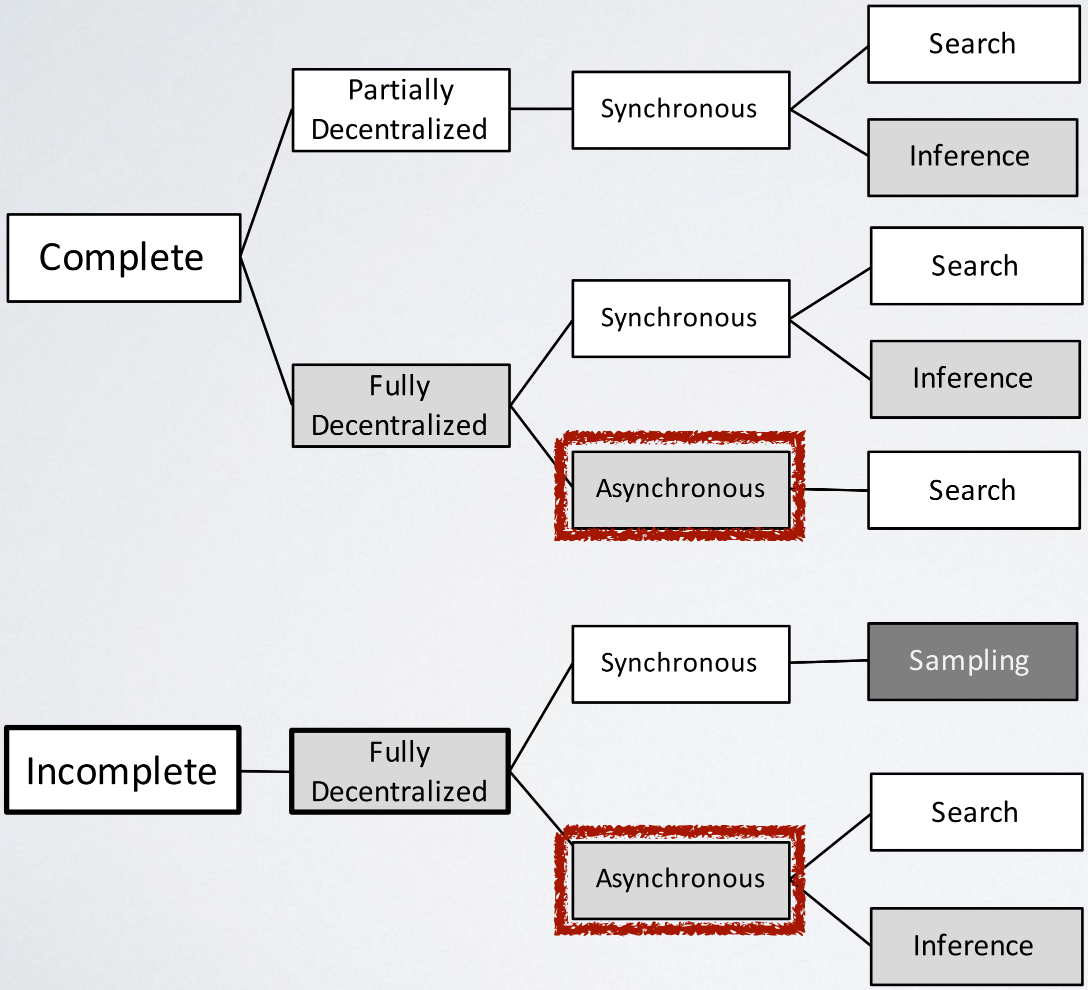
- Important Metrics:
  - Agent complexity
  - Network loads
  - Message size
  - Anytime
  - Quality guarantees
  - Execution time vs. solution quality

DCOP ALGORITHMS



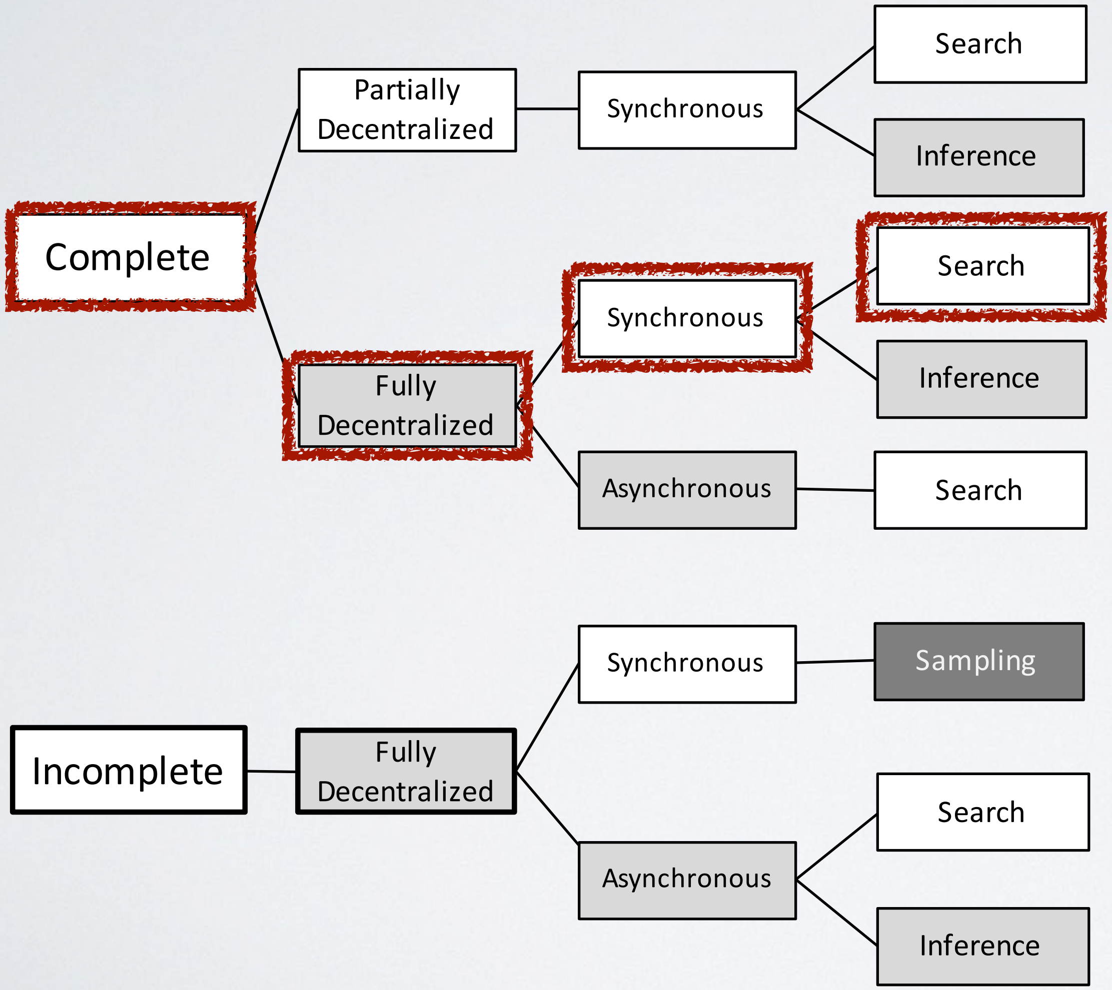
- Systematic process, divided in steps.
- Each agent waits for particular messages before acting
- Consistent view of the search process
- Typically, increases idle-time

DCOP ALGORITHMS



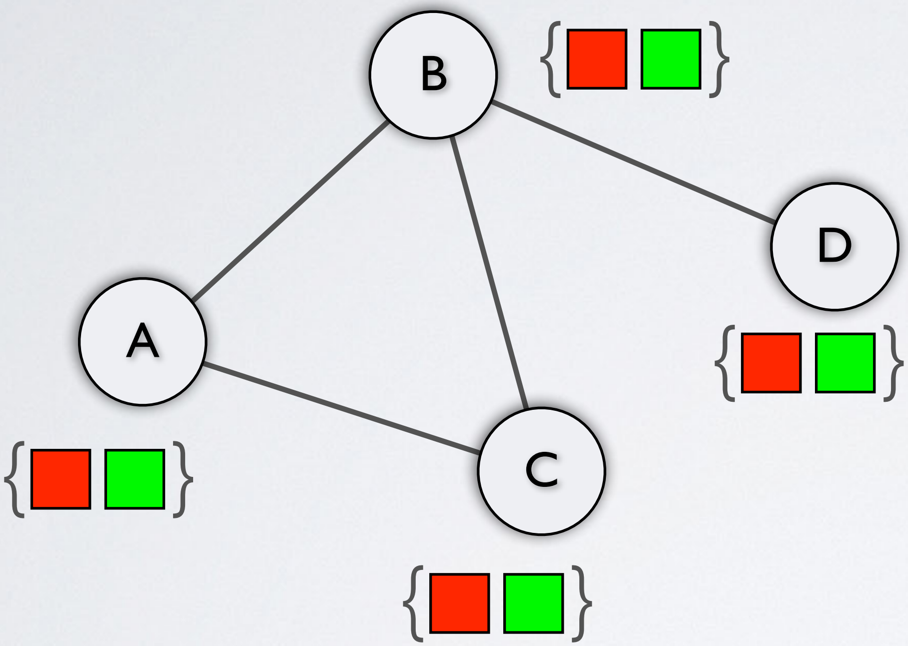
- Decision based on agents' local state
- Agents' actions do not depend on sequence of received messages
- Minimizes idle-time
- No guarantees on validity of local views

DCOP ALGORITHMS



Synchronous Branch and Bound (SBB)

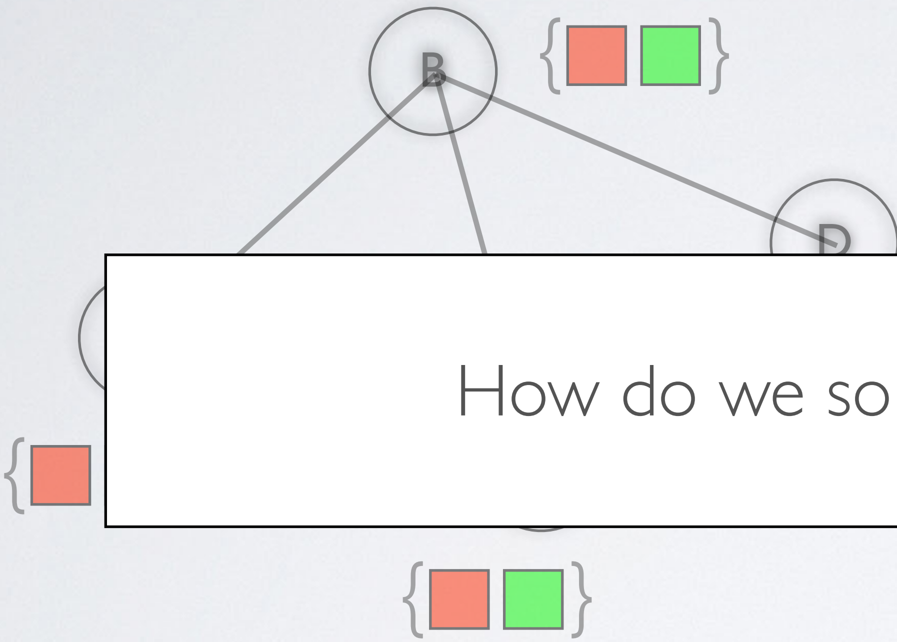
SBB



x_i	x_j	Cost (A,B)	Cost (A,C)	Cost (B,C)	Cost (B,D)
		5	5	5	3
		8	10	4	8
		20	20	3	10
		3	3	3	3

Katsutoshi Hirayama, Makoto Yokoo: Distributed Partial Constraint Satisfaction Problem. CP 1997: 222-236

SBB



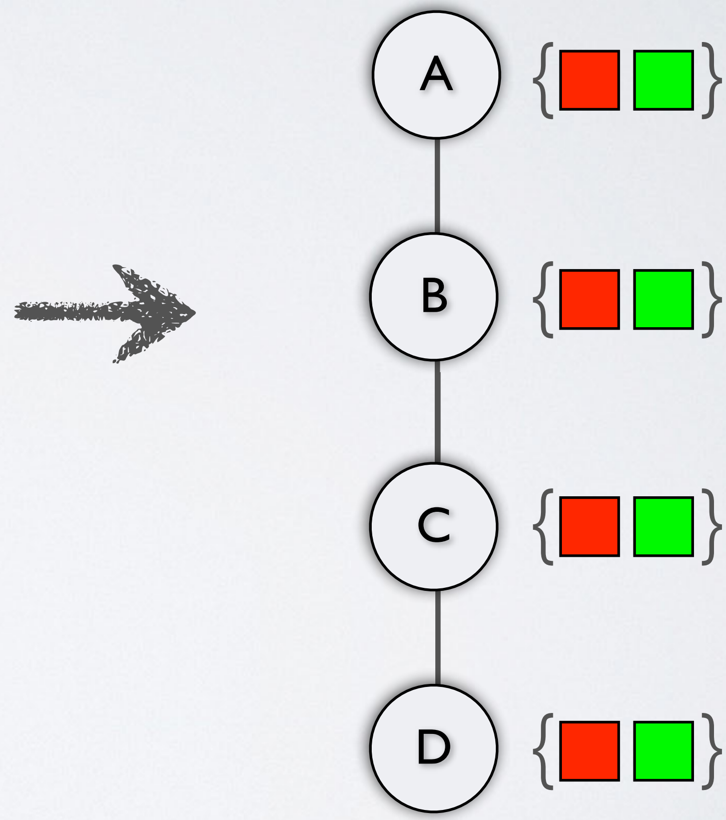
x_i	x_j	Cost (A,B)	Cost (A,C)	Cost (B,C)	Cost (B,D)
		5	5	5	3
					8
					10
		3	3	3	3

How do we solve this distributedly?

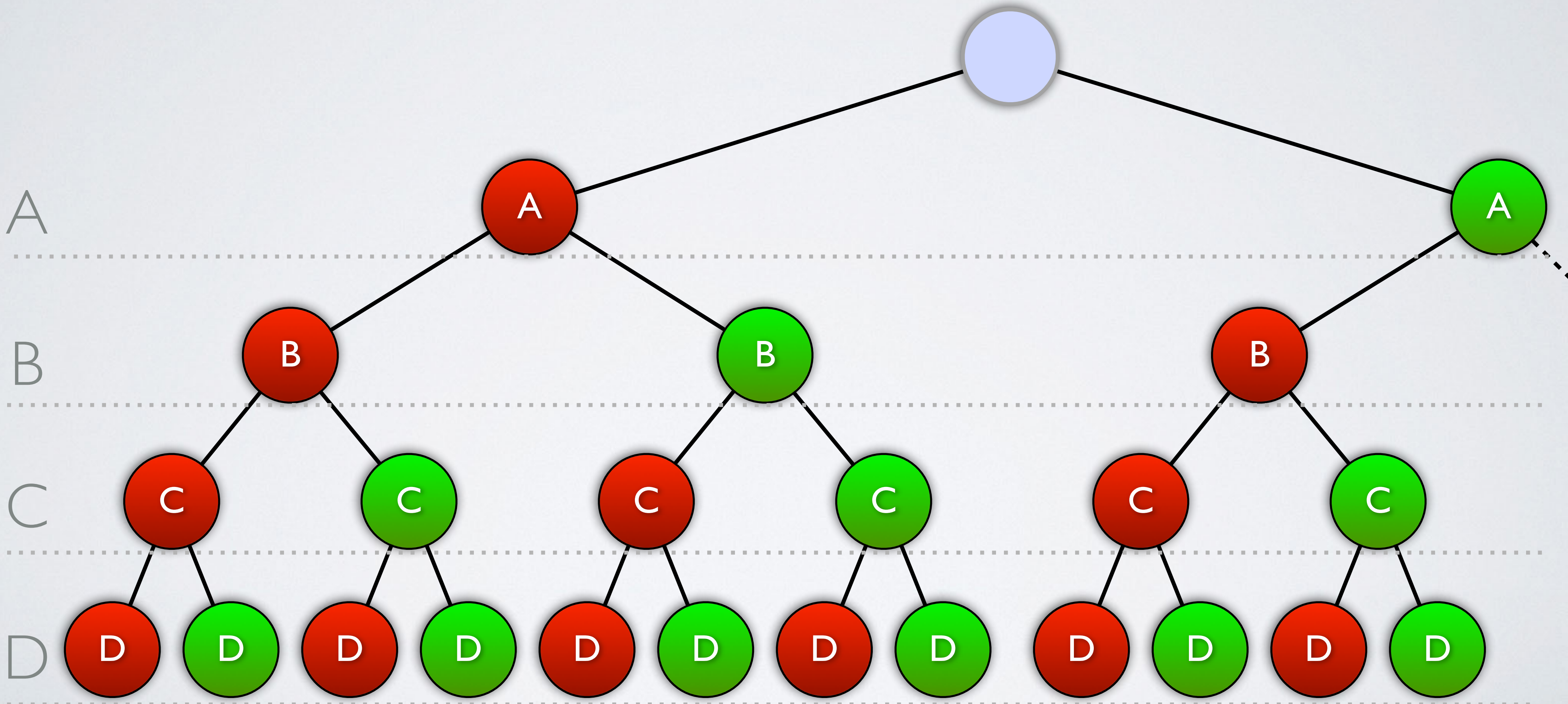
SBB

- Agents operate on a complete ordering
- Agents exchange CPA messages containing partial assignments.
- When a solution is found, its solution cost as an UB is broadcasted to all agents.
- The UB is used for branch pruning.

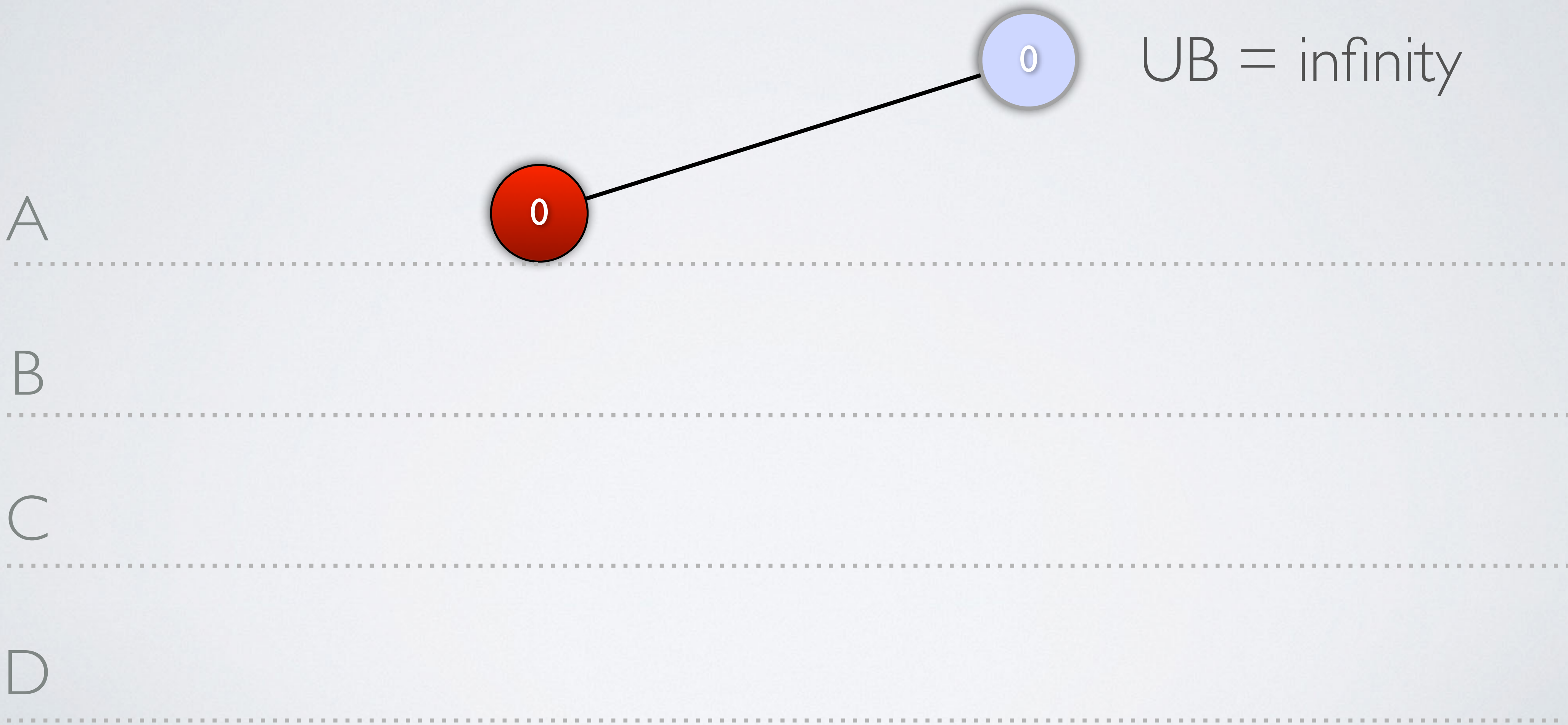
Complete Ordering



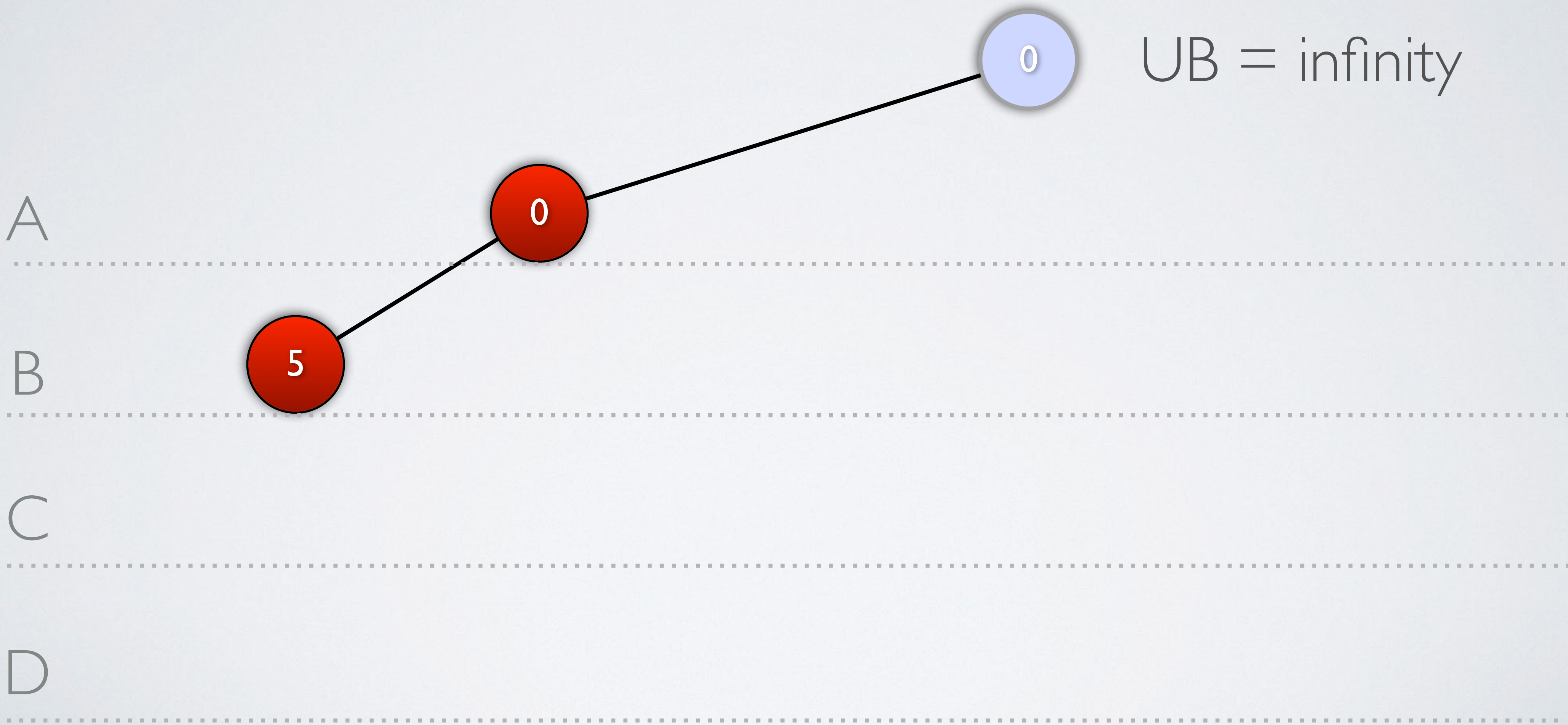
SBB



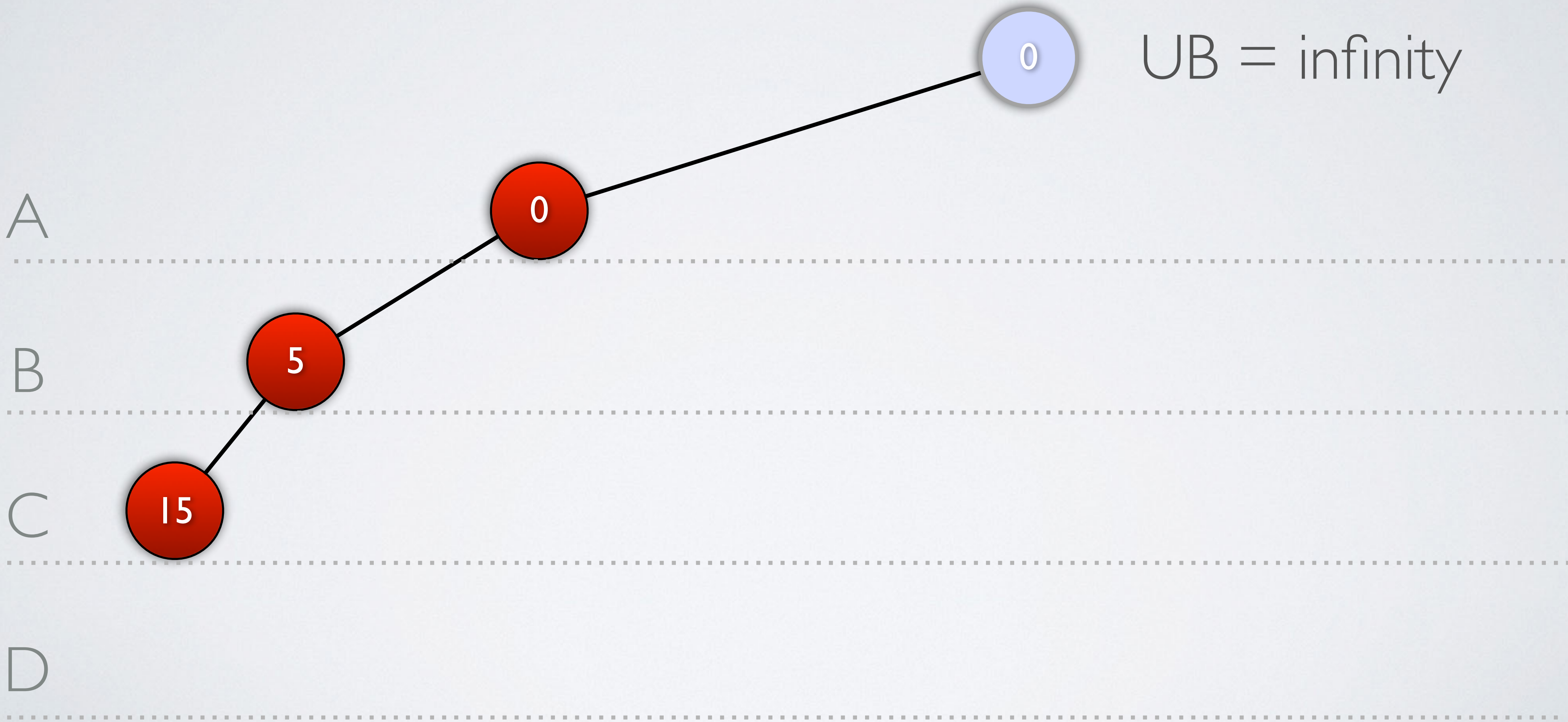
SBB



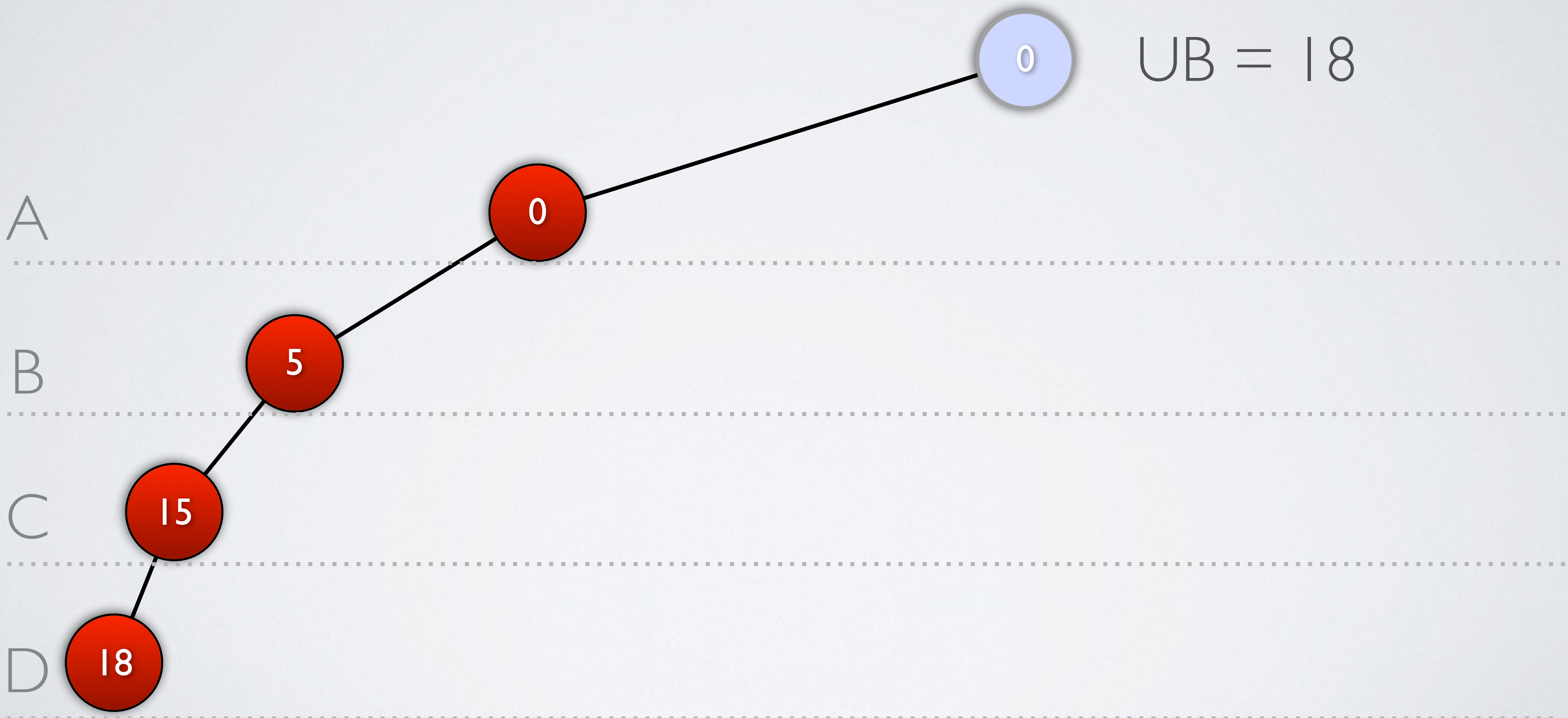
SBB



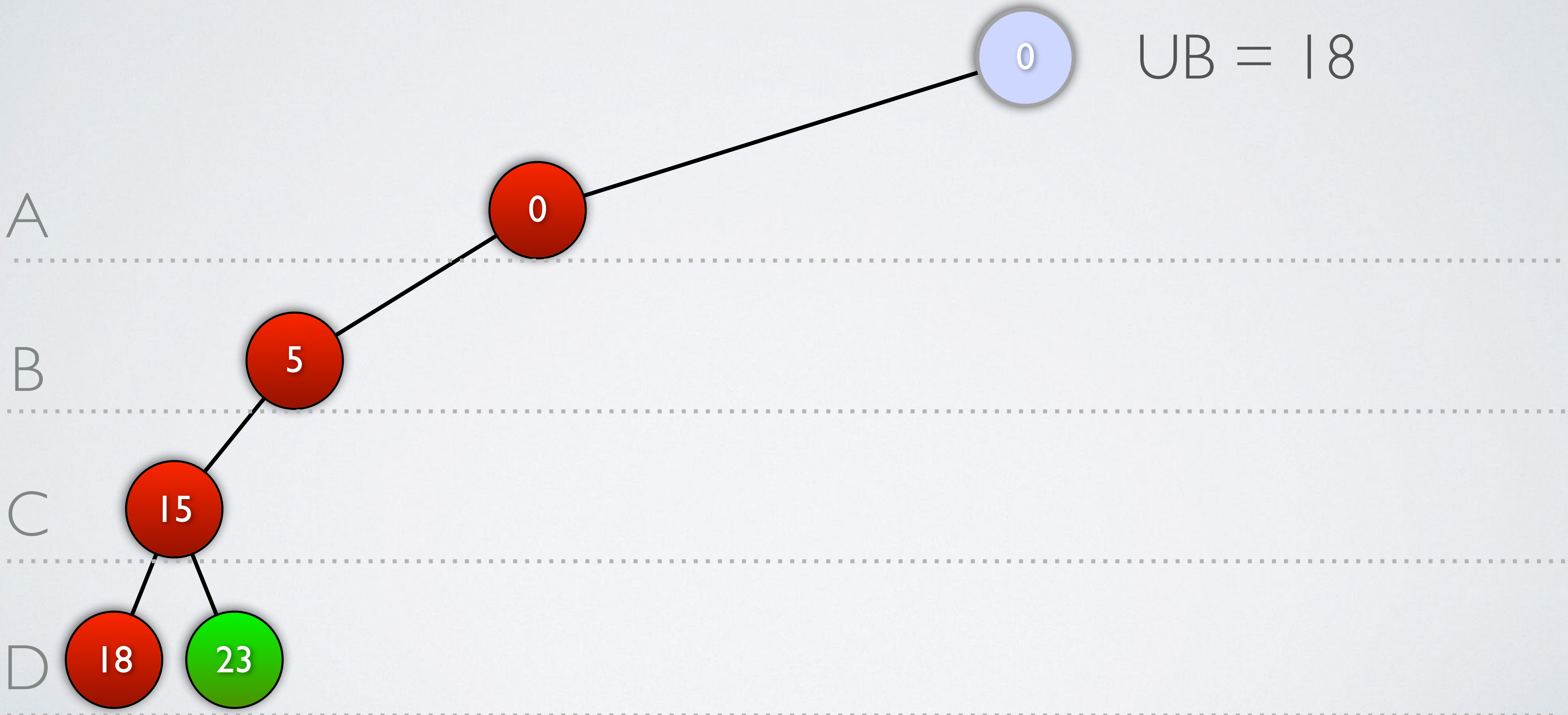
SBB



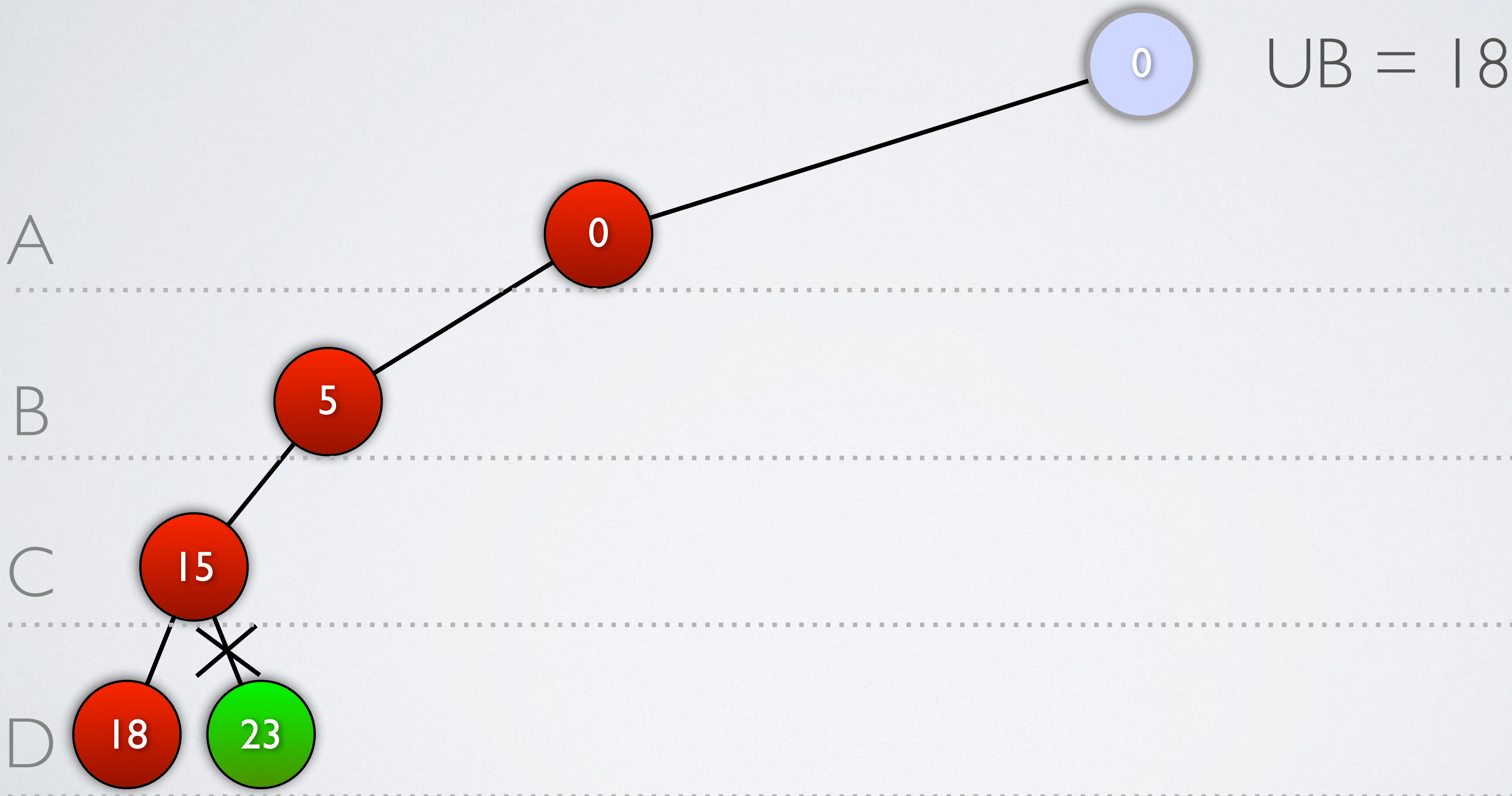
SBB



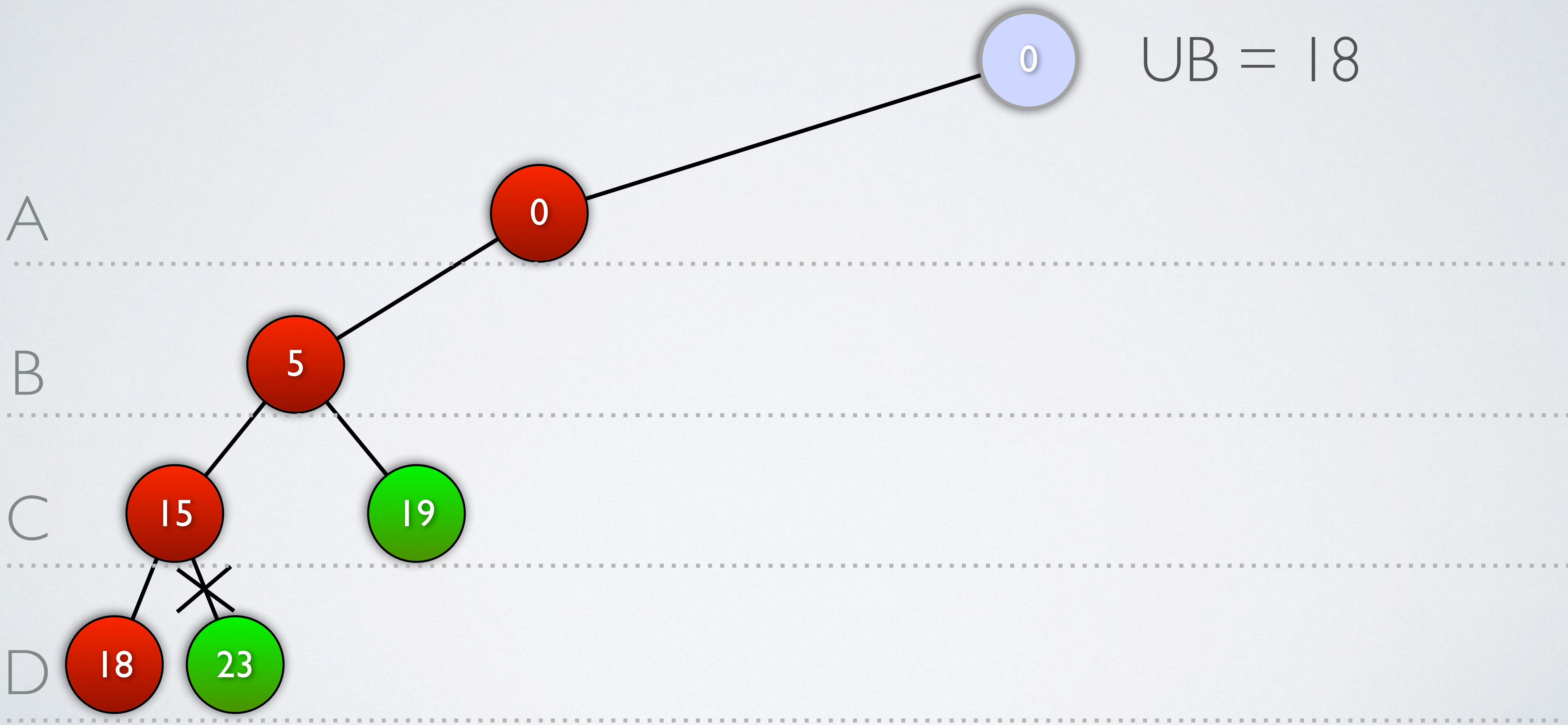
SBB



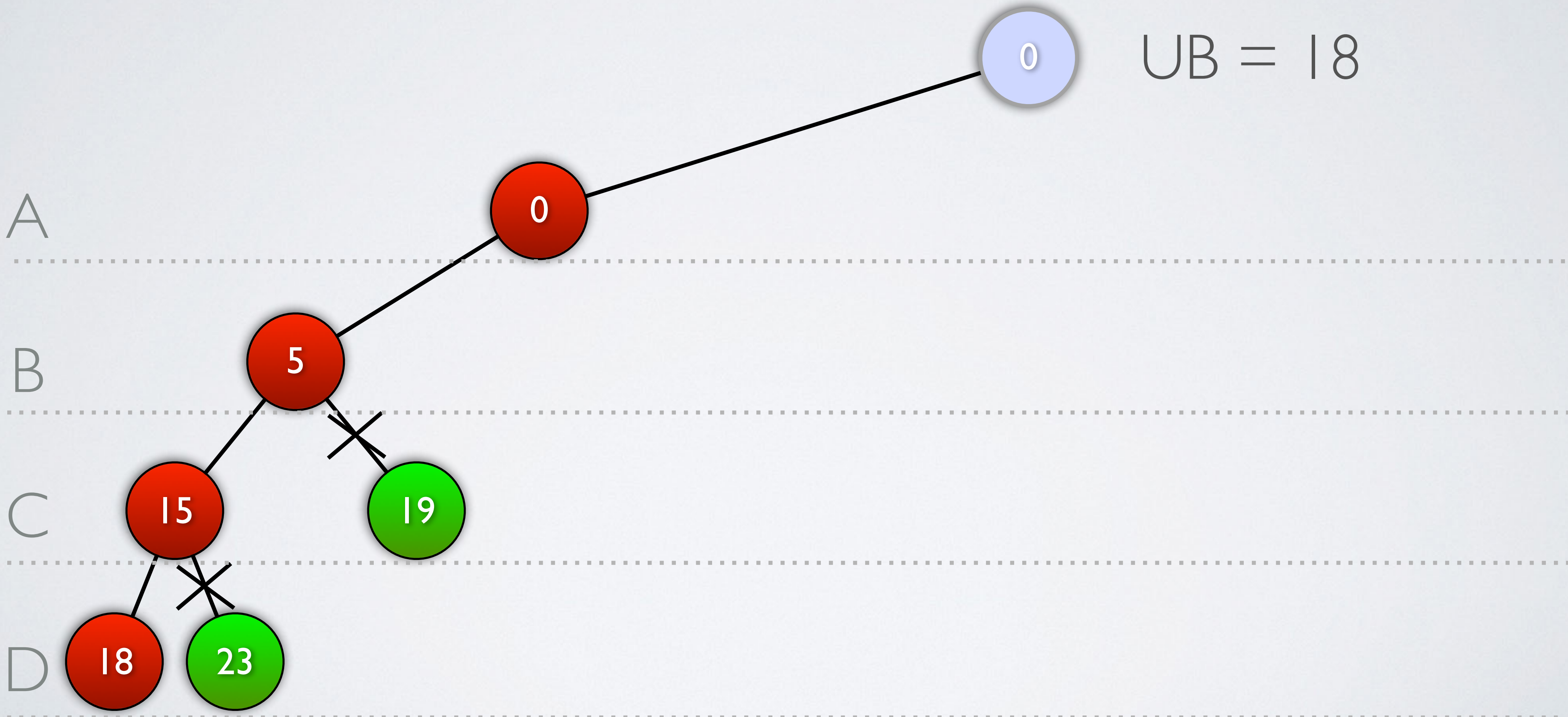
SBB



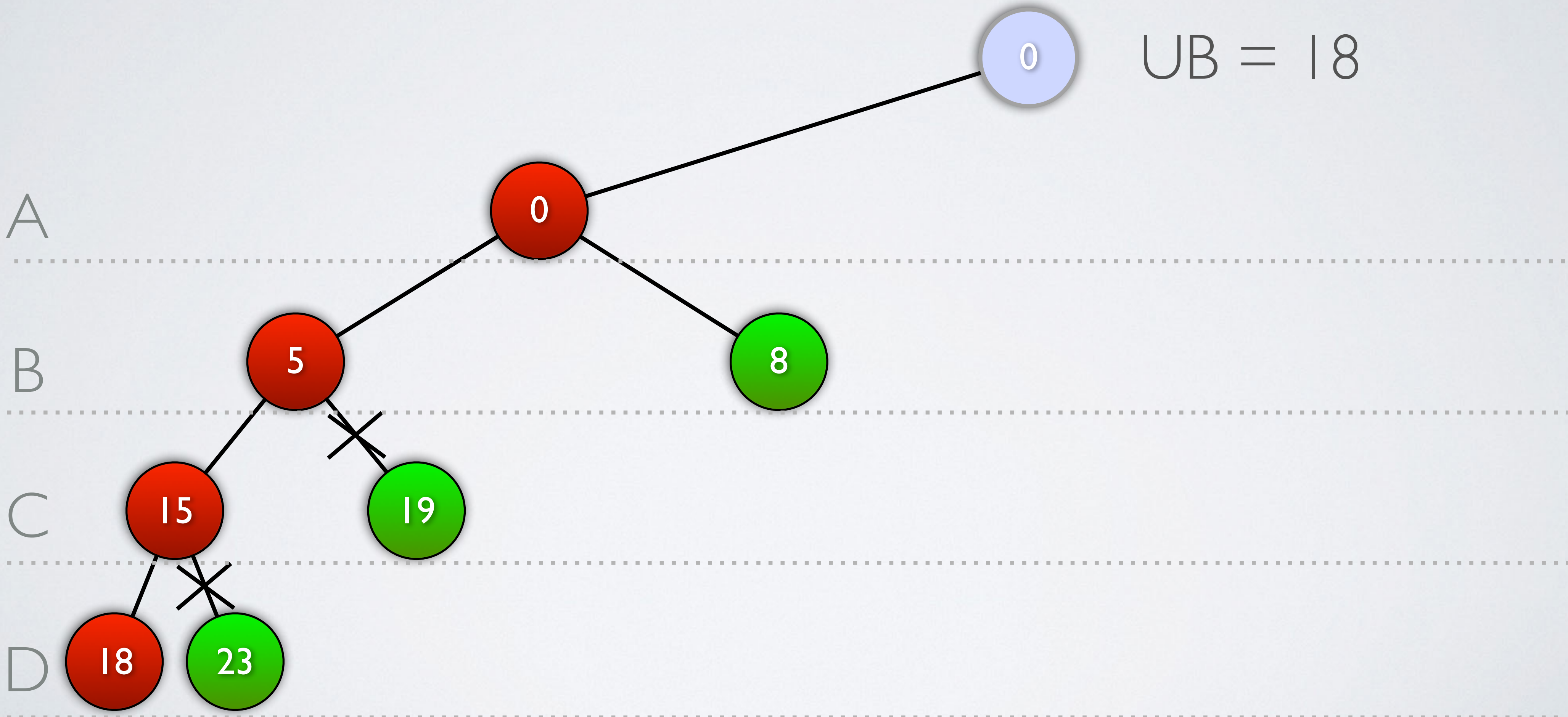
SBB



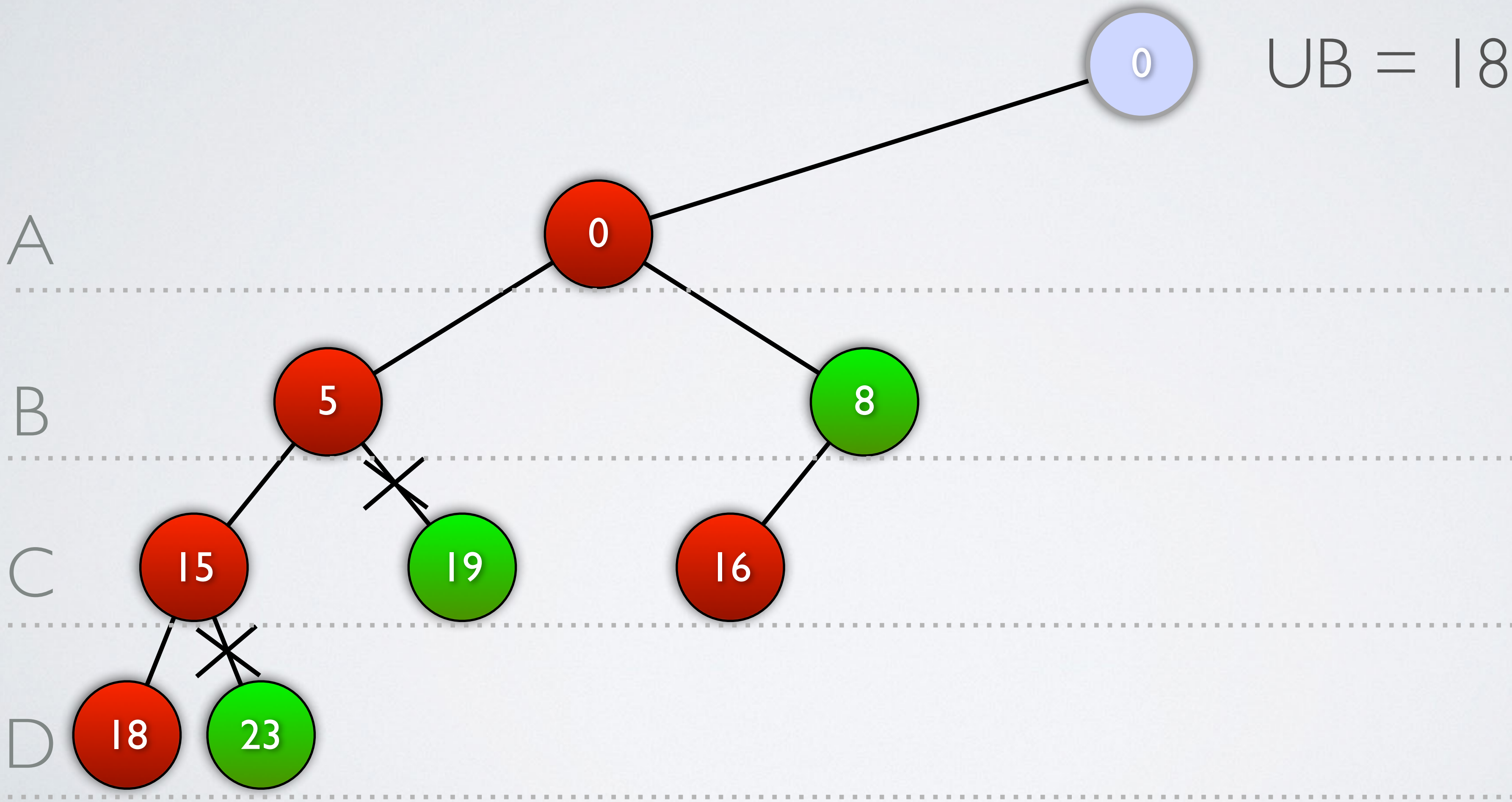
SBB



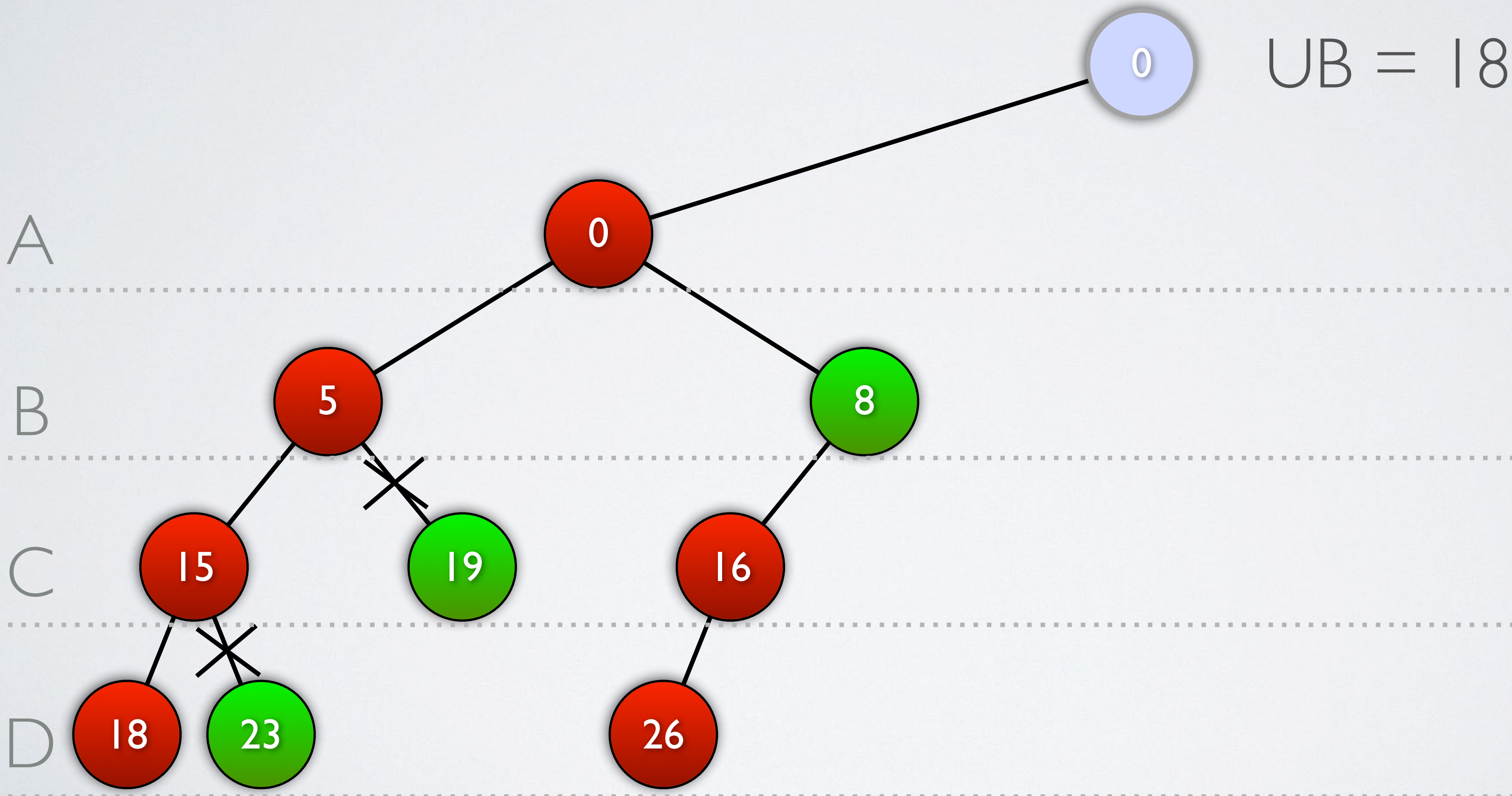
SBB



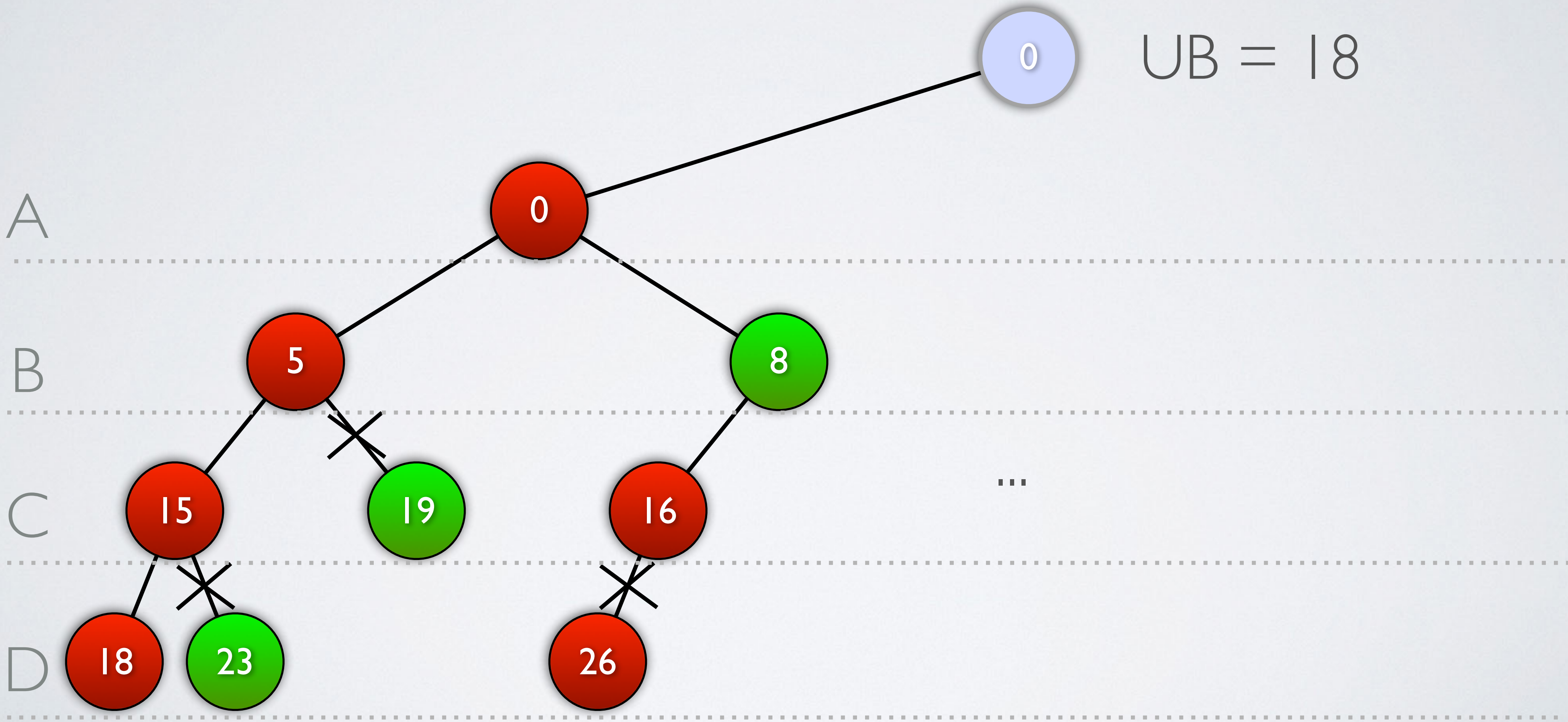
SBB



SBB



SBB

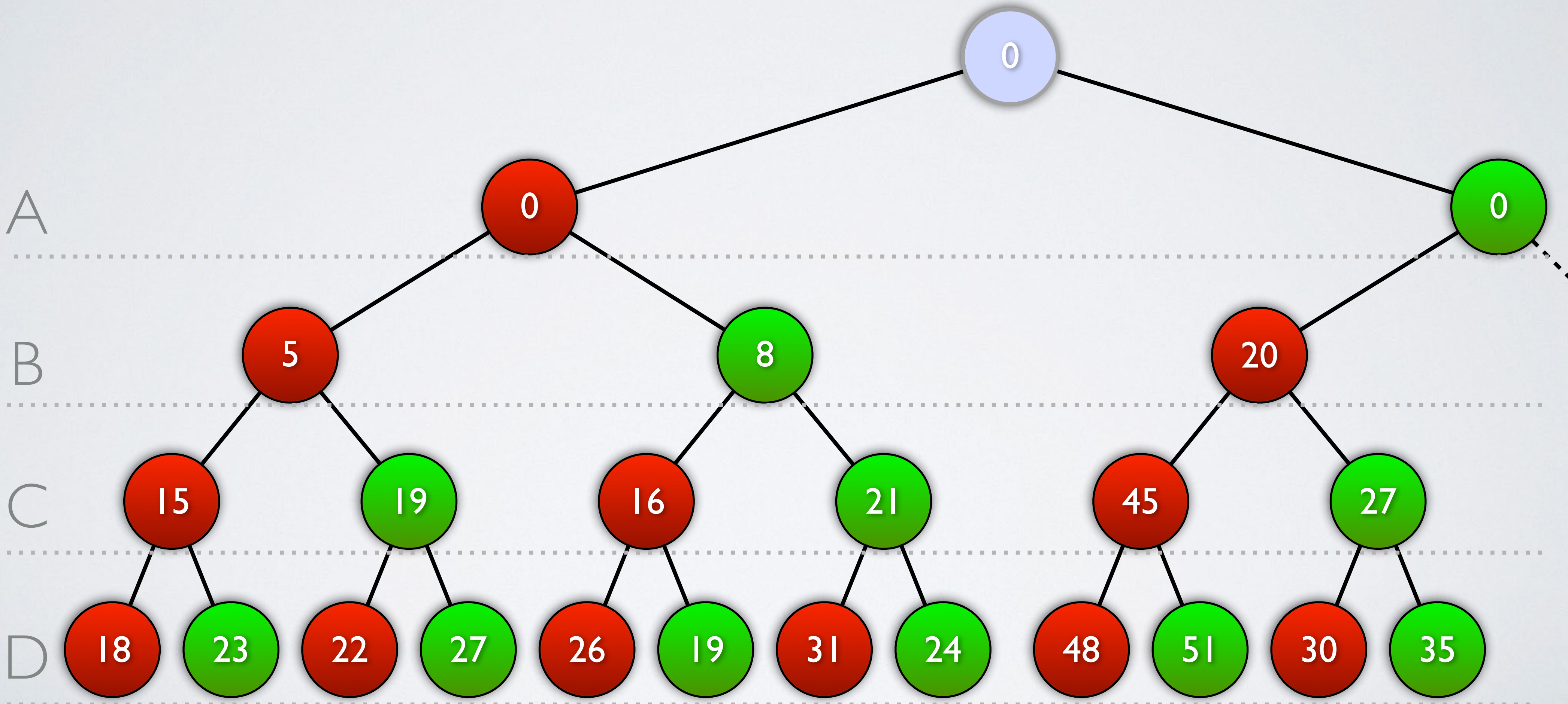


SBB

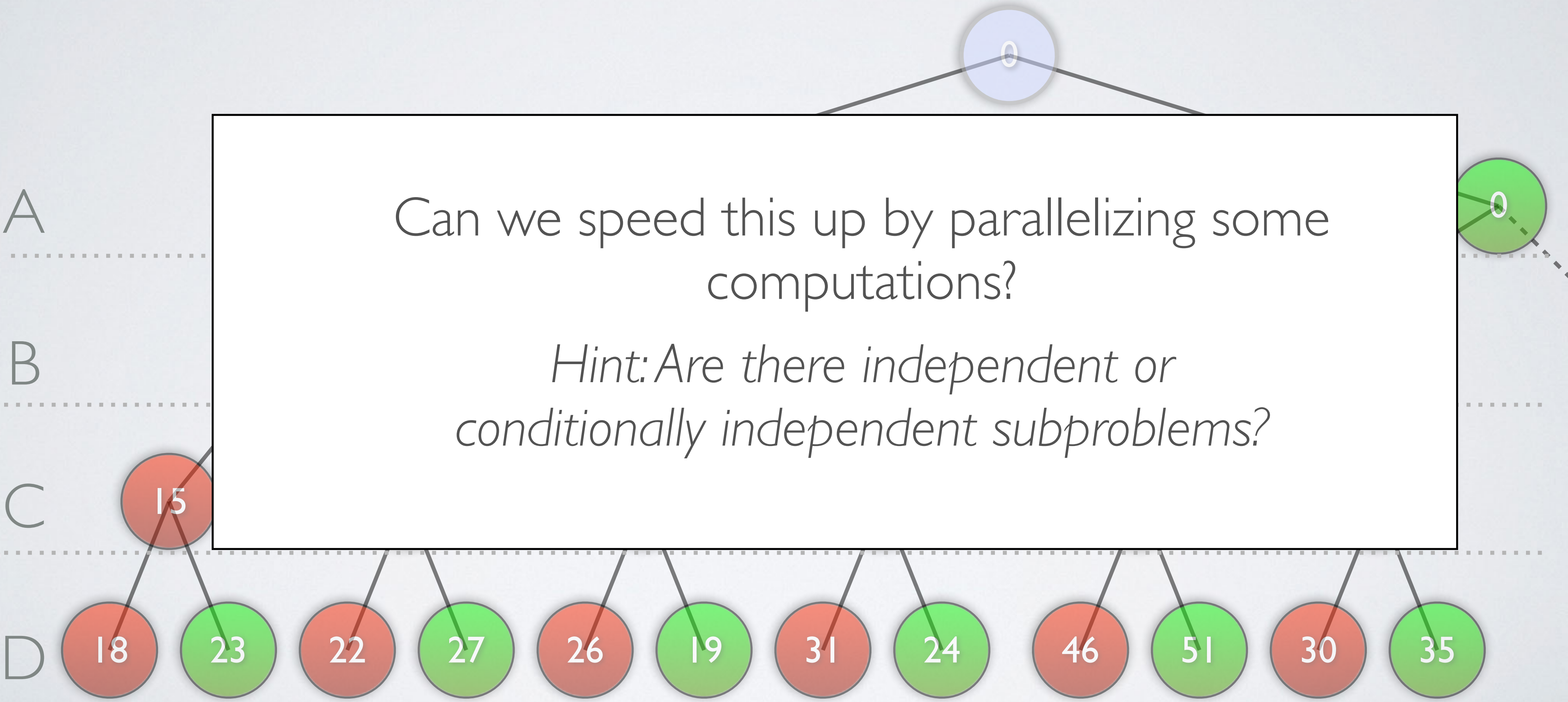
	SBB
Correct the solution it finds is optimal	Yes
Complete it terminates	Yes
Message Complexity max size of a message	$O(d)$
Network Load max number of messages	$O(b^d)$
Runtime	$O(b^d)$

branching factor = b
num variables = d

SBB



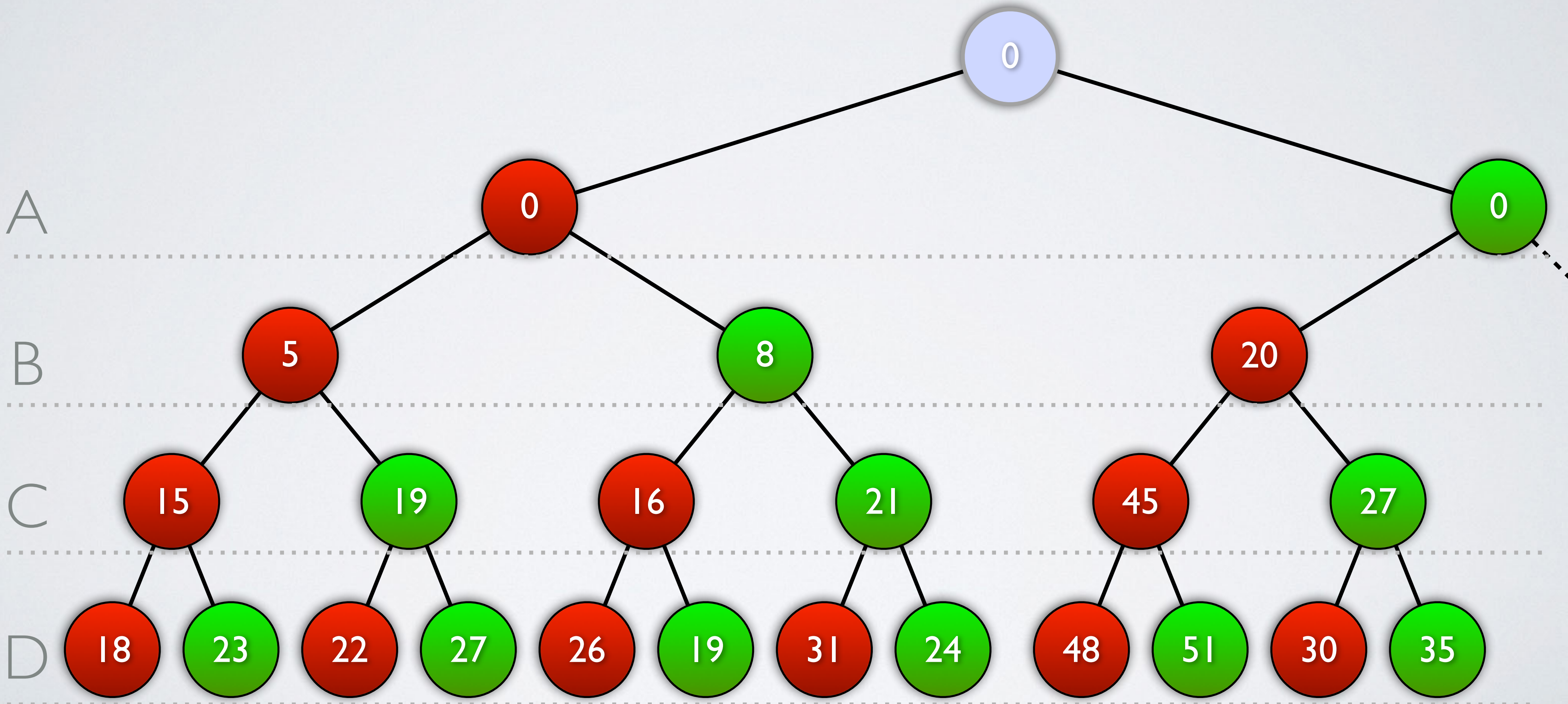
SBB



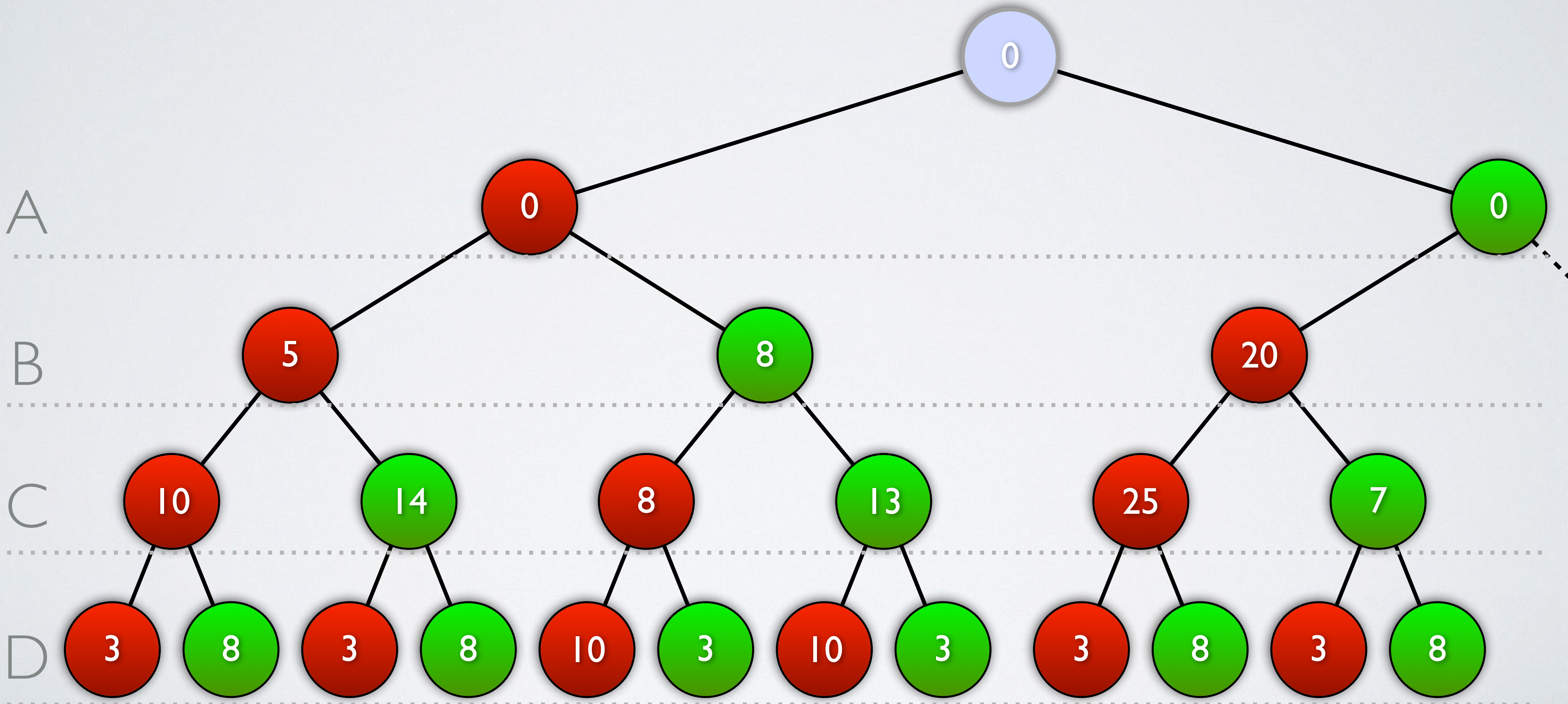
Can we speed this up by parallelizing some computations?

Hint: Are there independent or conditionally independent subproblems?

SBB

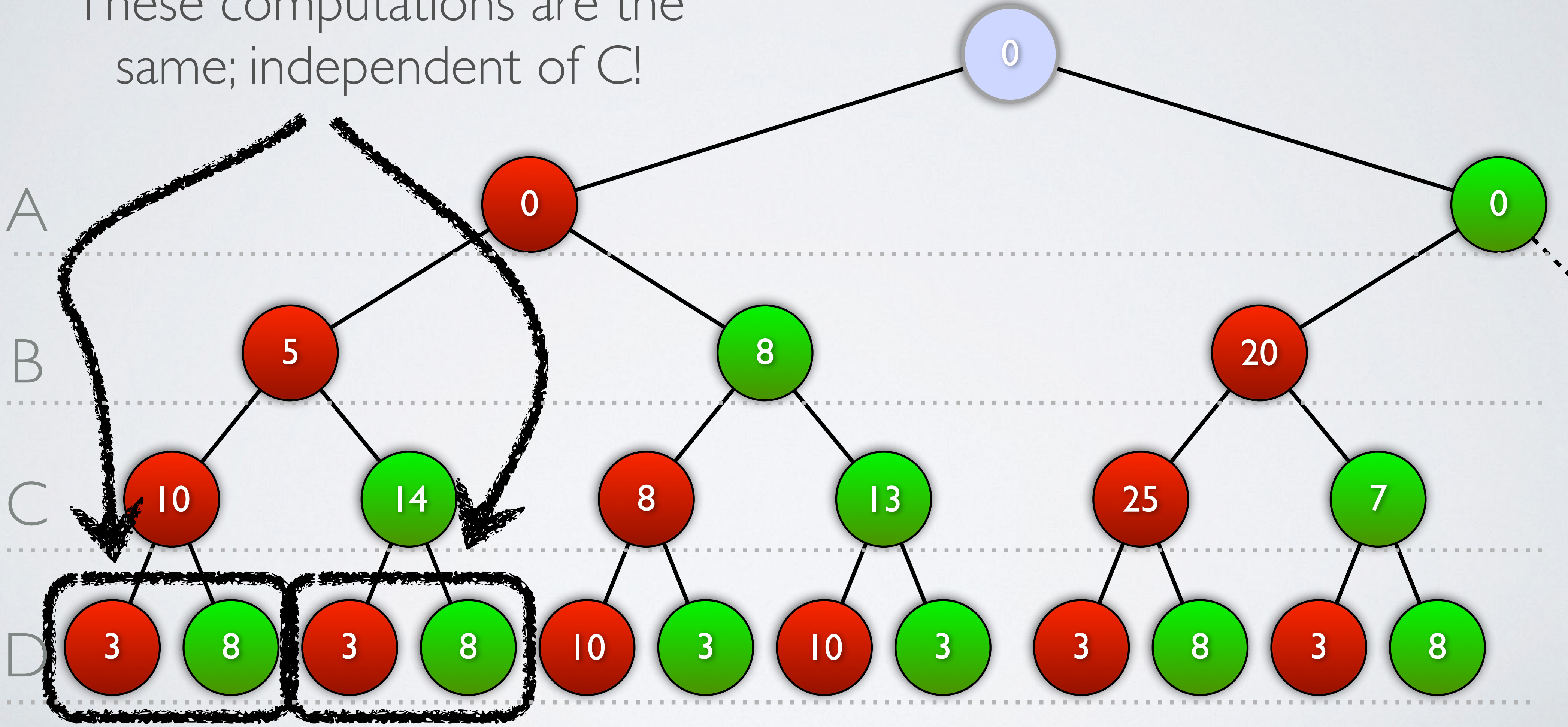


SBB

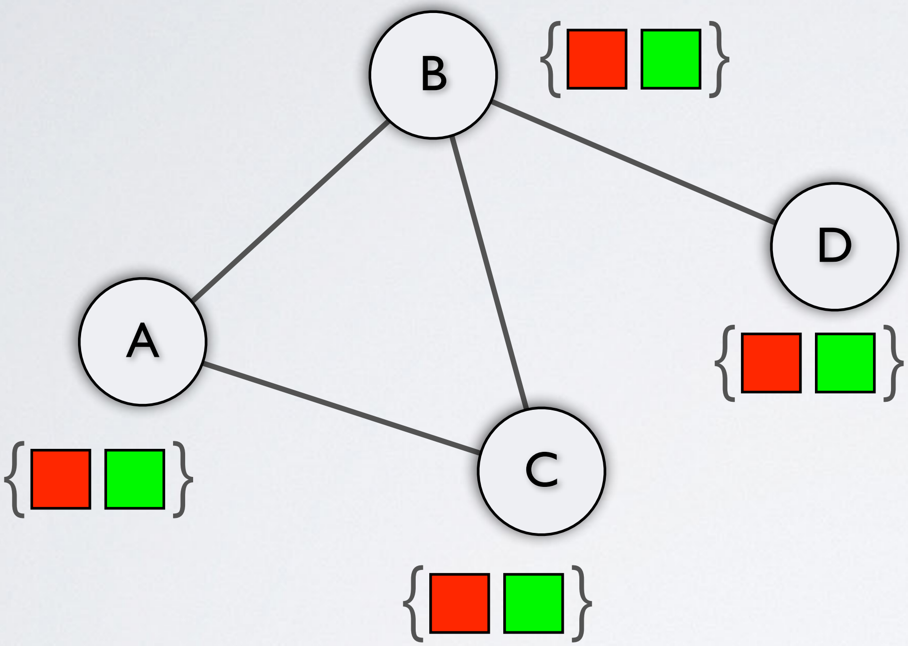


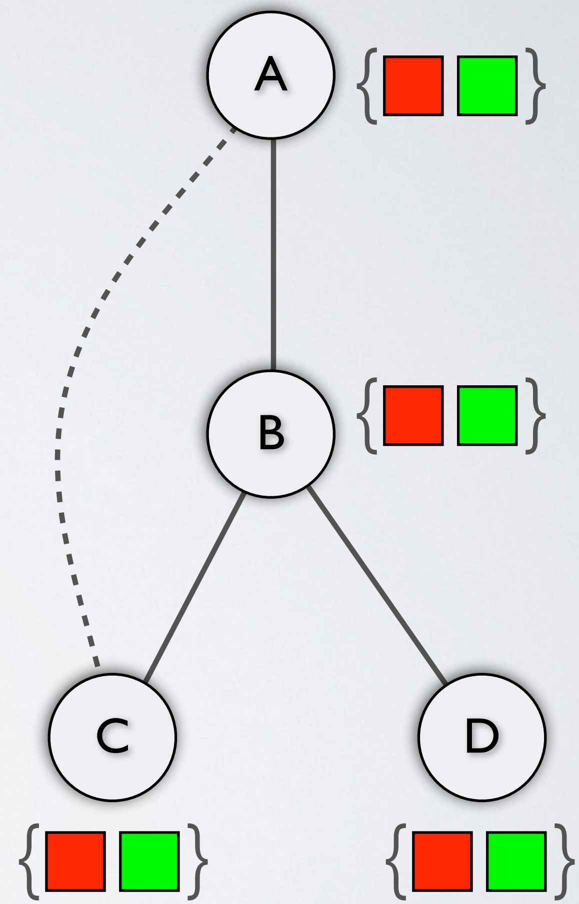
SBB

These computations are the same; independent of C!



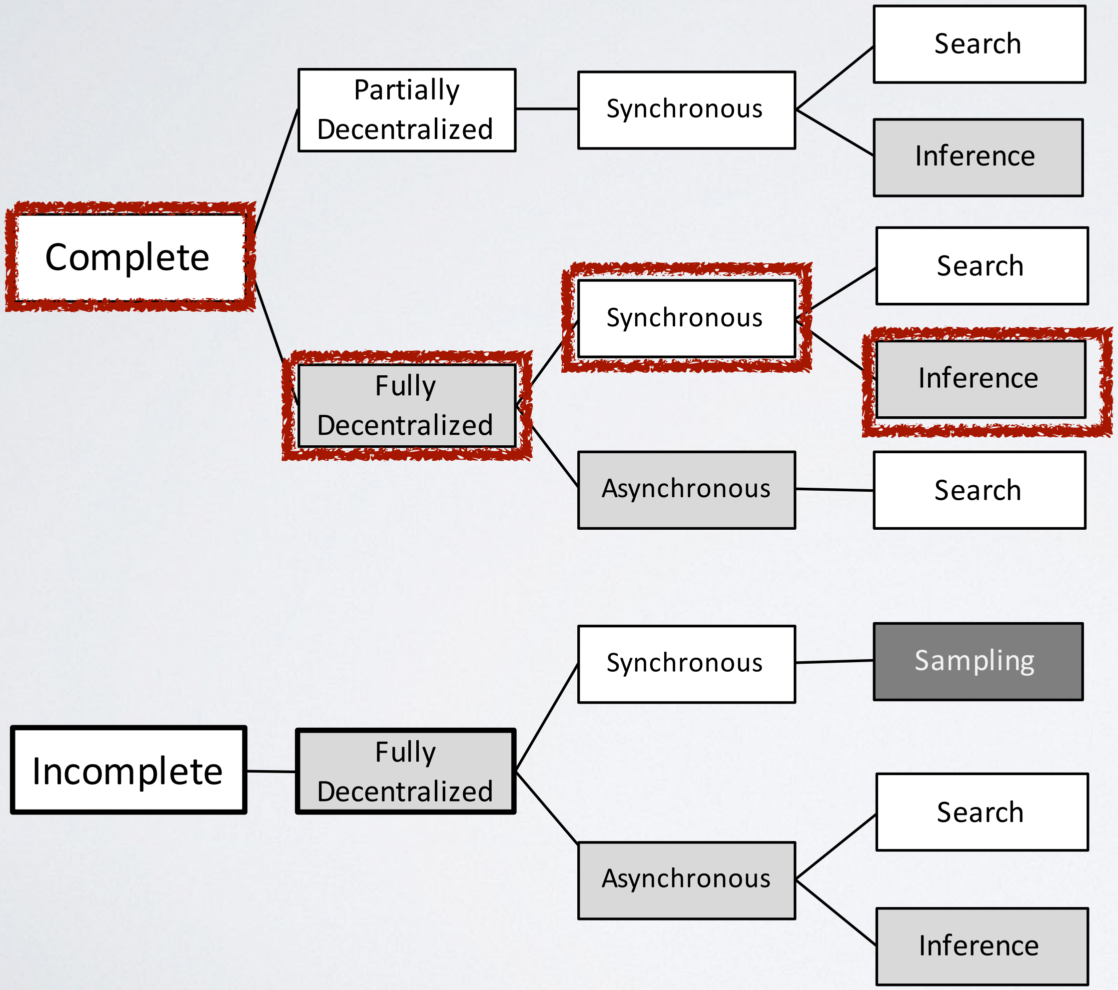
PSEUDO-TREE





Definition: *A spanning tree of the constraint graph such that no two nodes in sibling subtrees share a constraint in the constraint graph*

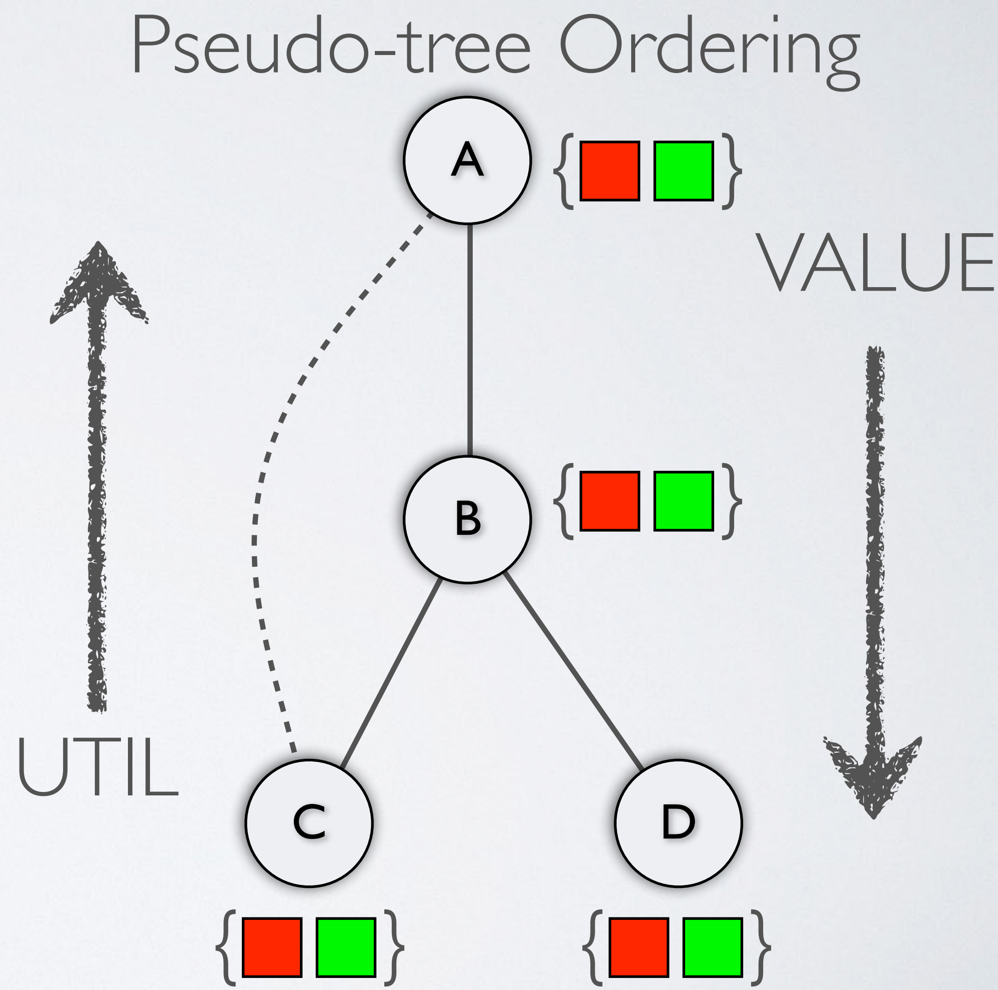
DCOP ALGORITHMS



Distributed Pseudotree Optimization Procedure (DPOP)

DPOP

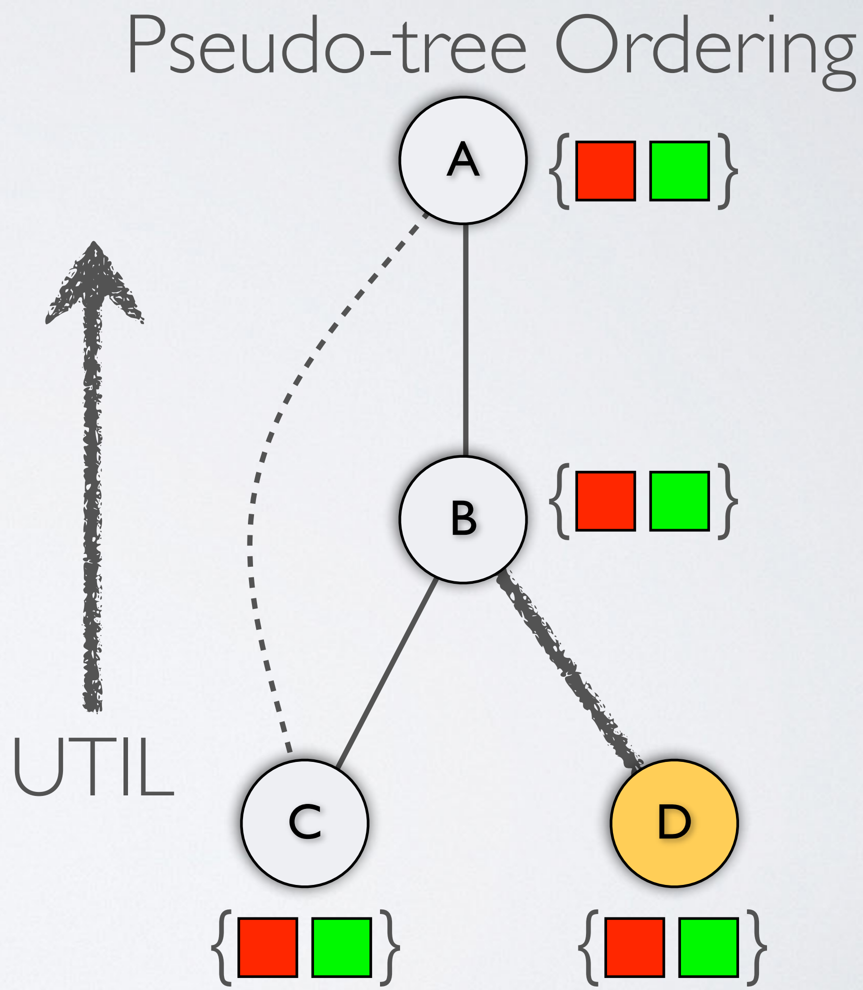
- Extension of the Bucket Elimination (BE)
- Agents operate on a pseudo-tree ordering
- UTIL phase: Leaves to root
- VALUE phase: Root to leaves



Adrian Petcu, Boi Faltings: A Scalable Method for Multiagent Constraint Optimization. IJCAI 2005: 266-271

DPOP

B	D	(B,D)
r	r	3
r	g	8
g	r	10
g	g	3



DPOP

B	D	(B,D)
r	r	3
r	g	8
g	r	10
g	g	3

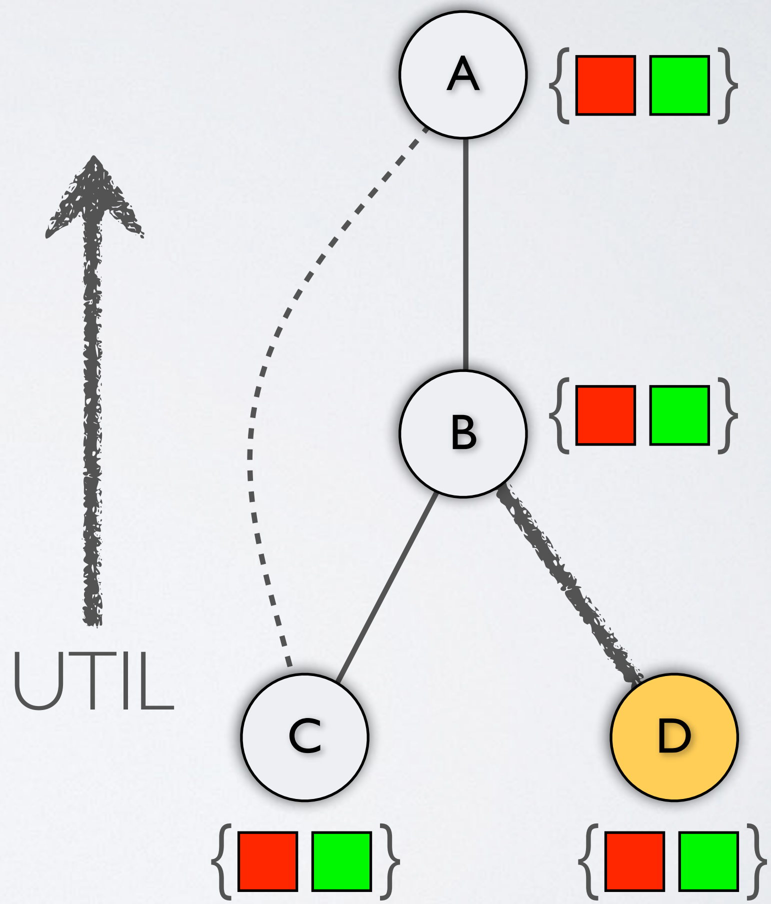
min{3, 8} = 3

min{10, 3} = 3

MSG to B

B	cost
r	3
g	3

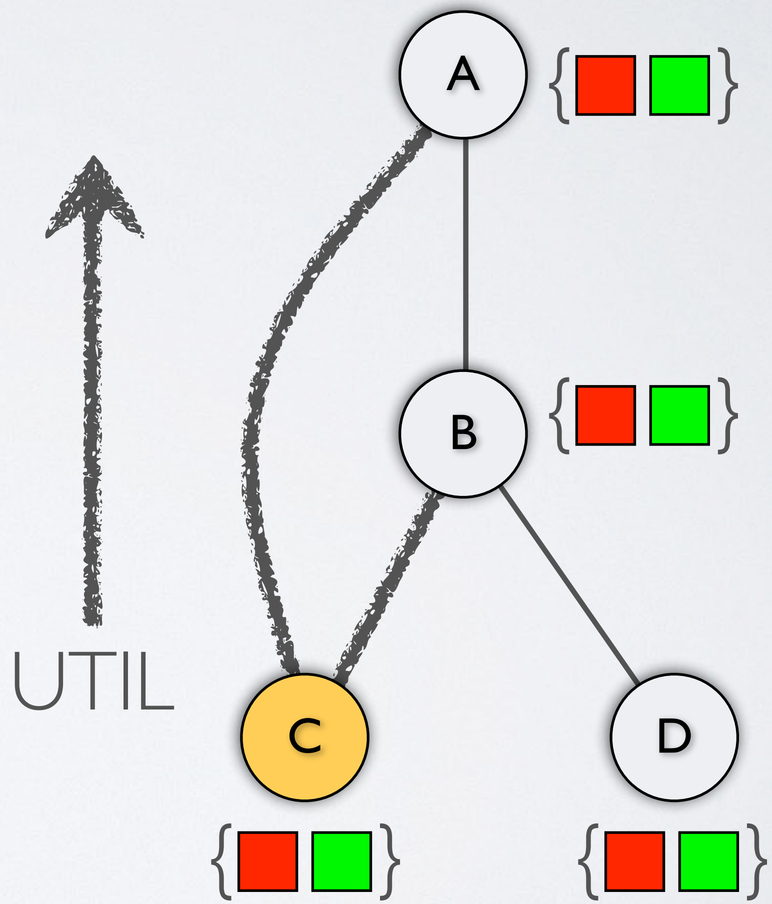
Pseudo-tree Ordering



DPOP

A	B	C	(B,C)	(A,C)	
r	r	r	5	5	10
r	r	g	4	8	12
r	g	r	3	5	8
r	g	g	3	8	11
g	r	r	5	10	15
g	r	g	4	3	7
g	g	r	3	10	13
g	g	g	3	3	6

Pseudo-tree Ordering



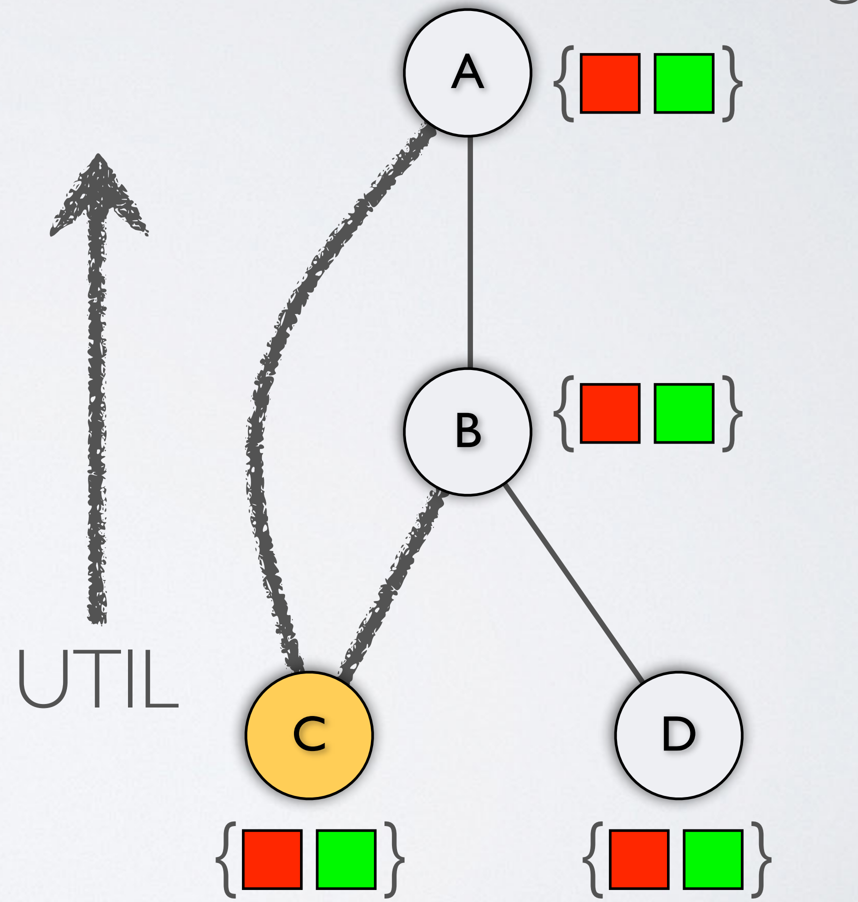
DPOP

A	B	C	(B,C)	(A,C)	
r	r	r	5	5	10
r	r	g	4	8	12
r	g	r	3	5	8
r	g	g	3	8	11
g	r	r	5	10	15
g	r	g	4	3	7
g	g	r	3	10	13
g	g	g	3	3	6

MSG to B

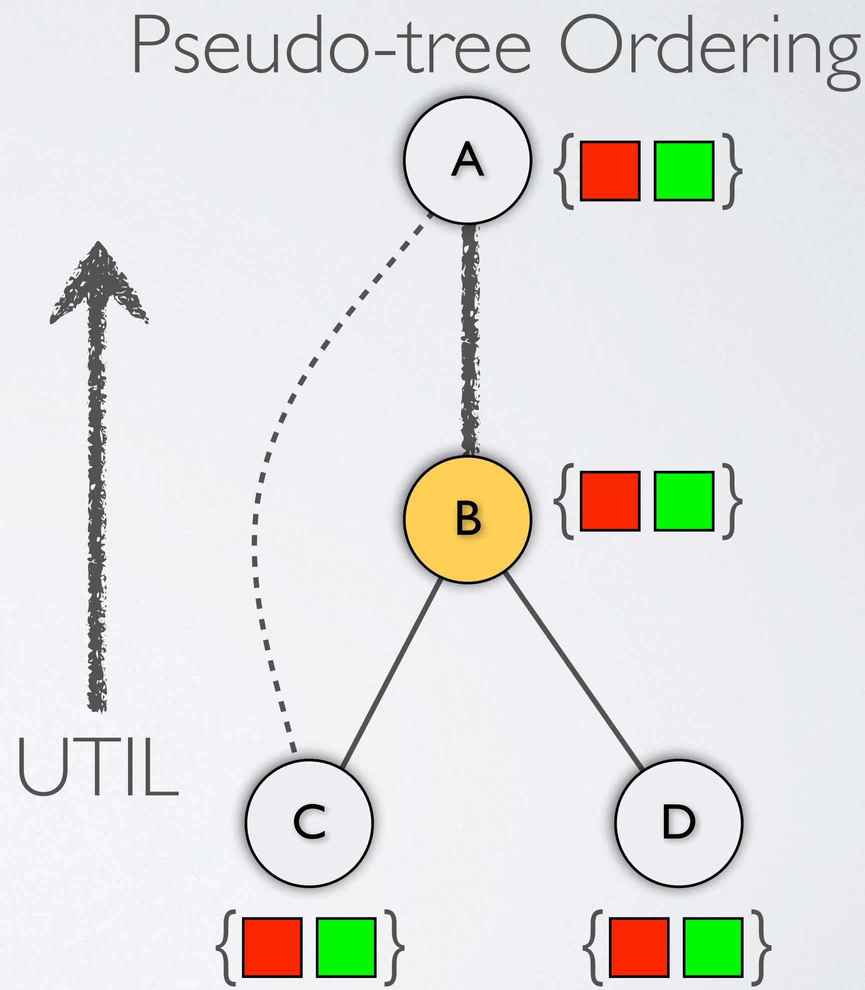
A	B	
r	r	10
r	g	8
g	r	7
g	g	6

Pseudo-tree Ordering



DPOP

A	B	(A,B)	Util C	Util D	
r	r	5	10	3	18
r	g	8	8	3	19
g	r	20	7	3	30
g	g	3	6	3	12



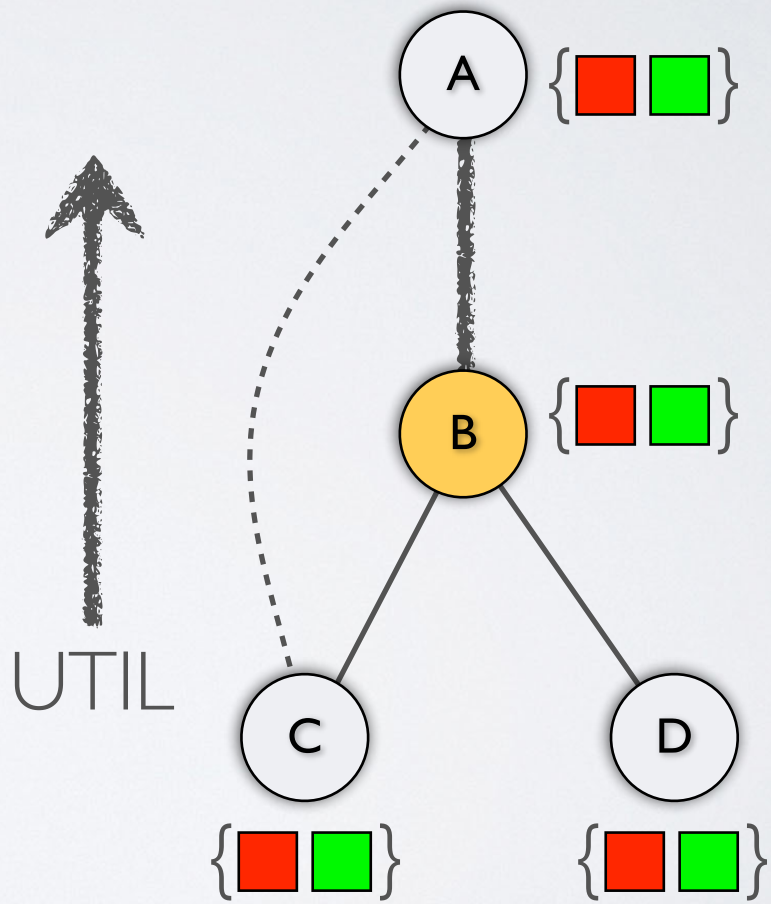
DPOP

A	B	(A,B)	Util C	Util D	
r	r	5	10	3	18
r	g	8	8	3	19
g	r	20	7	3	30
g	g	3	6	3	12

MSG to A

A	cost
r	18
g	12

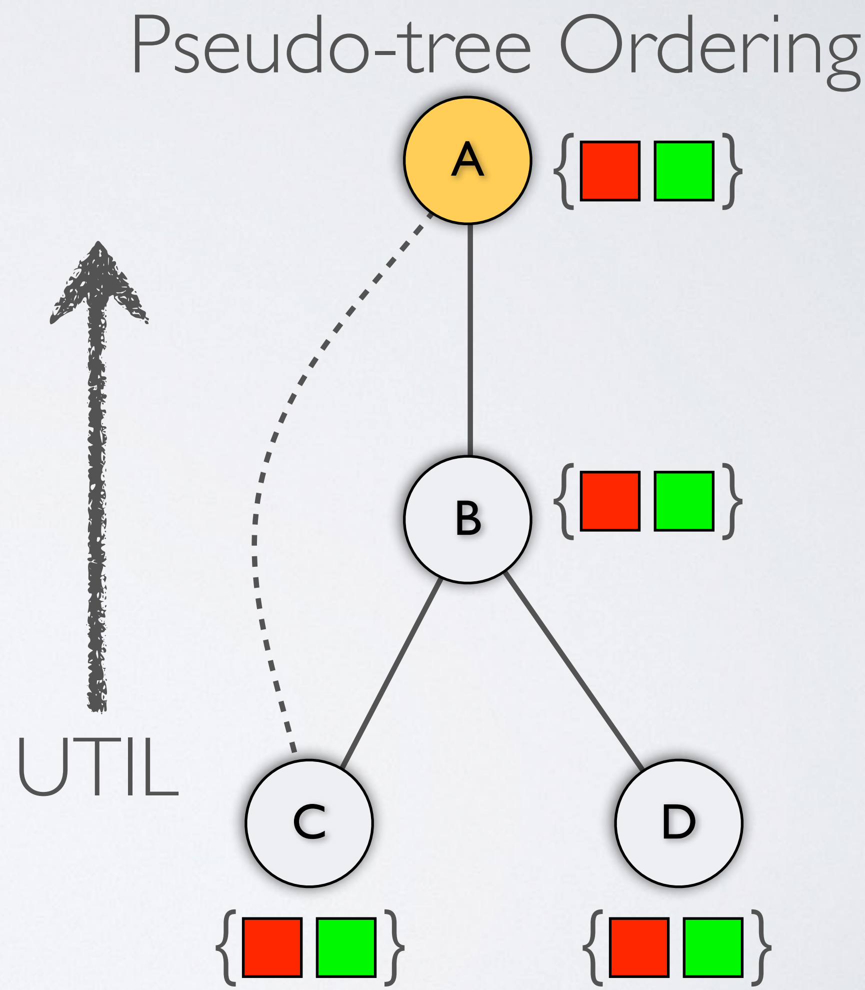
Pseudo-tree Ordering



DPOP

A	cost
r	18
g	12

optimal cost = 12

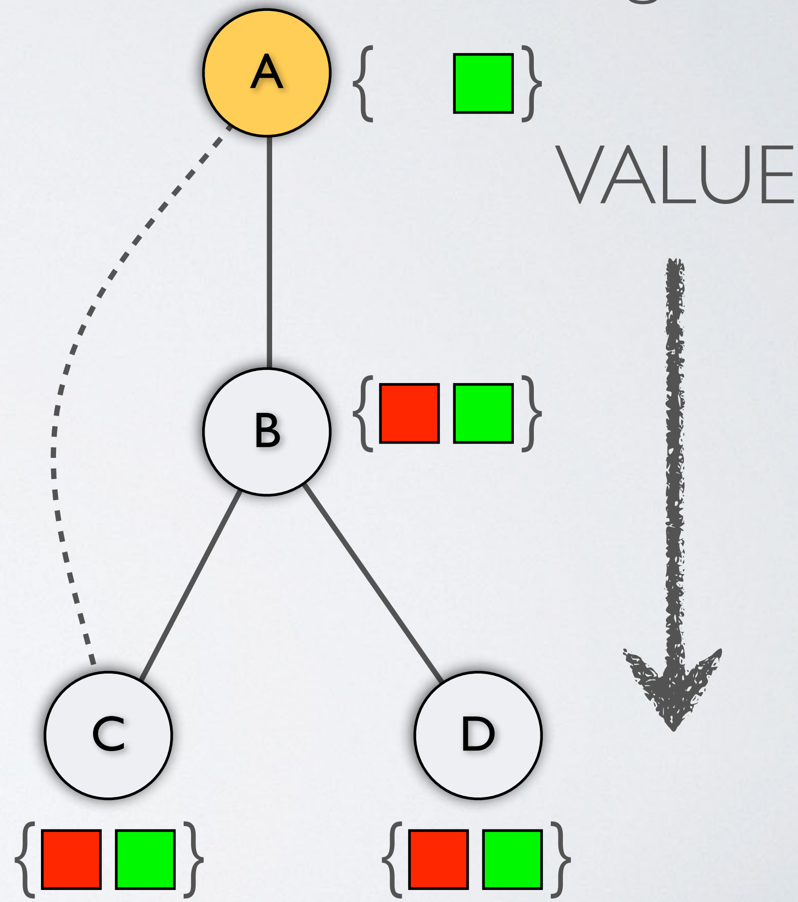


DPOP

A	cost
r	18
g	12

- Select value for A = ‘g’
- Send MSG A = ‘g’ to agents B and C

Pseudo-tree Ordering

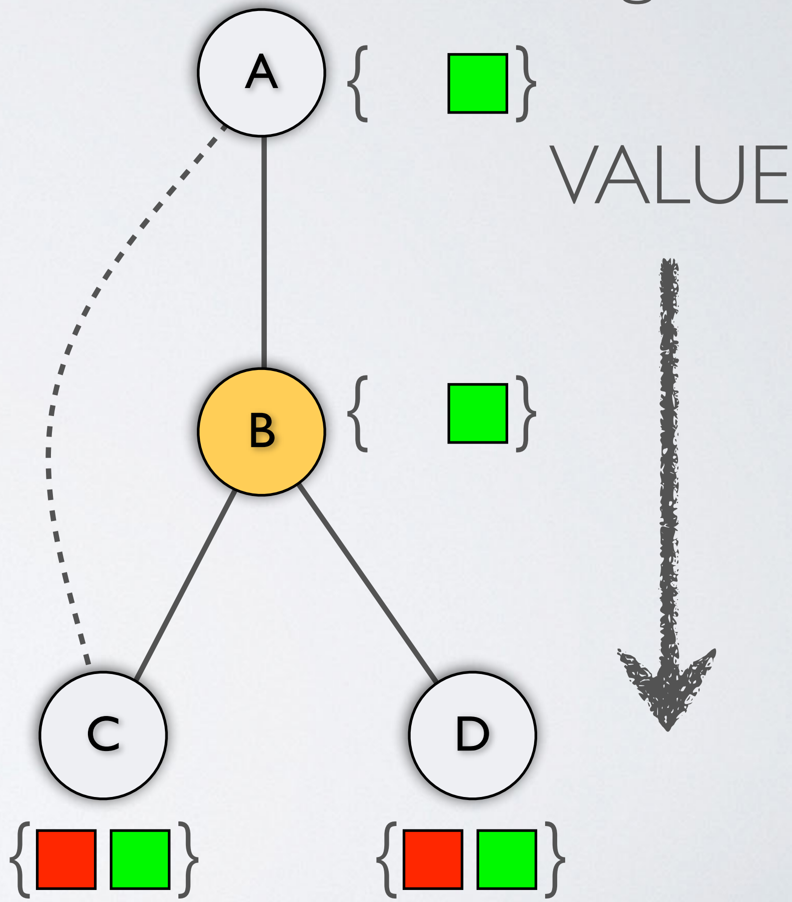


DPOP

A	B	(A,B)	Util C	Util D	
r	r	5	10	3	18
r	g	8	8	3	19
g	r	20	7	3	30
g	**g**	3	6	3	12

- Select value for B = 'g'
- Send MSG B = 'g' to agents C and D

Pseudo-tree Ordering

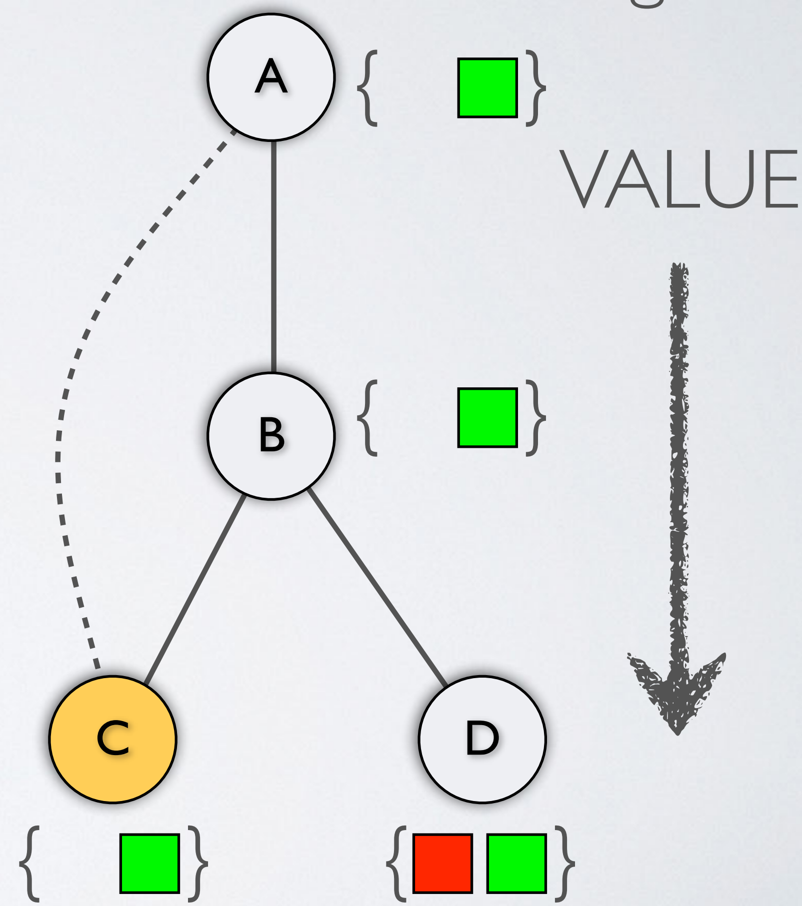


DPOP

A	B	C	(B,C)	(A,C)	
r	r	r	5	5	10
r	r	g	4	8	12
r	g	r	3	5	8
r	g	g	3	8	11
g	r	r	5	10	15
g	r	g	4	3	7
g	g	r	3	10	13
g	g	**g**	3	3	6

- Select value for C = 'g'

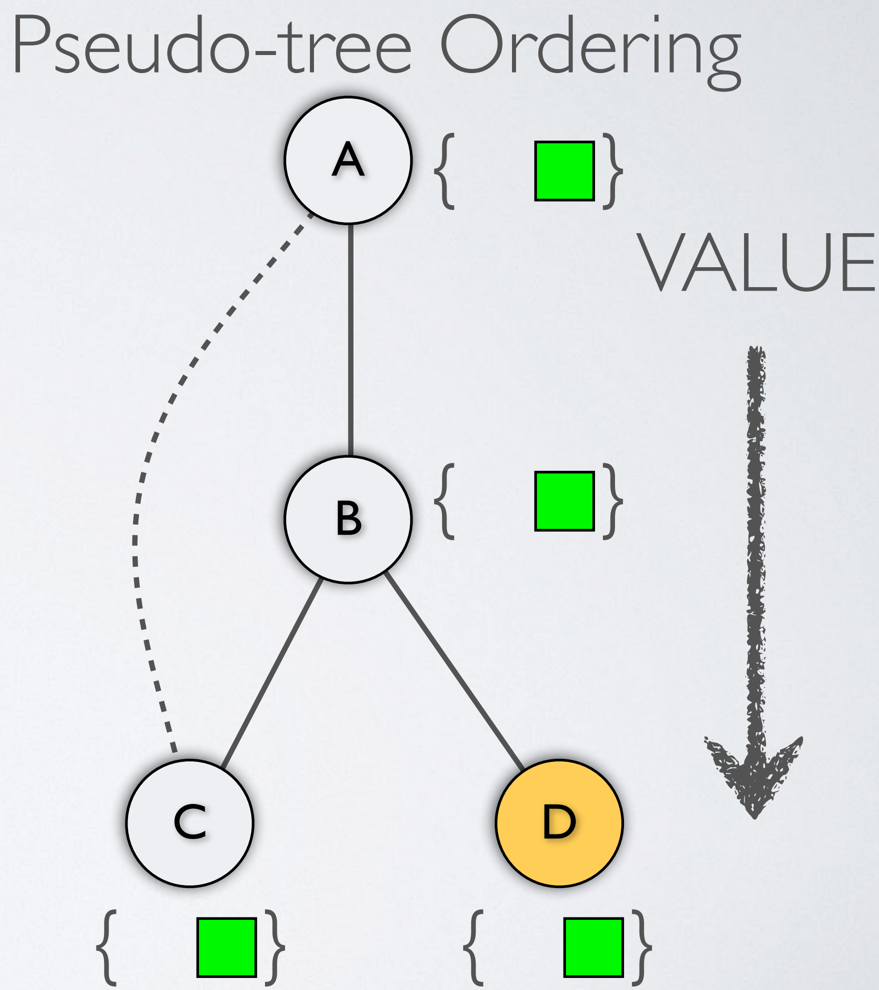
Pseudo-tree Ordering



DPOP

B	D	(B,D)
r	r	3
r	g	8
g	r	10
g	**g**	3

- Select value for D = 'g'



DPOP

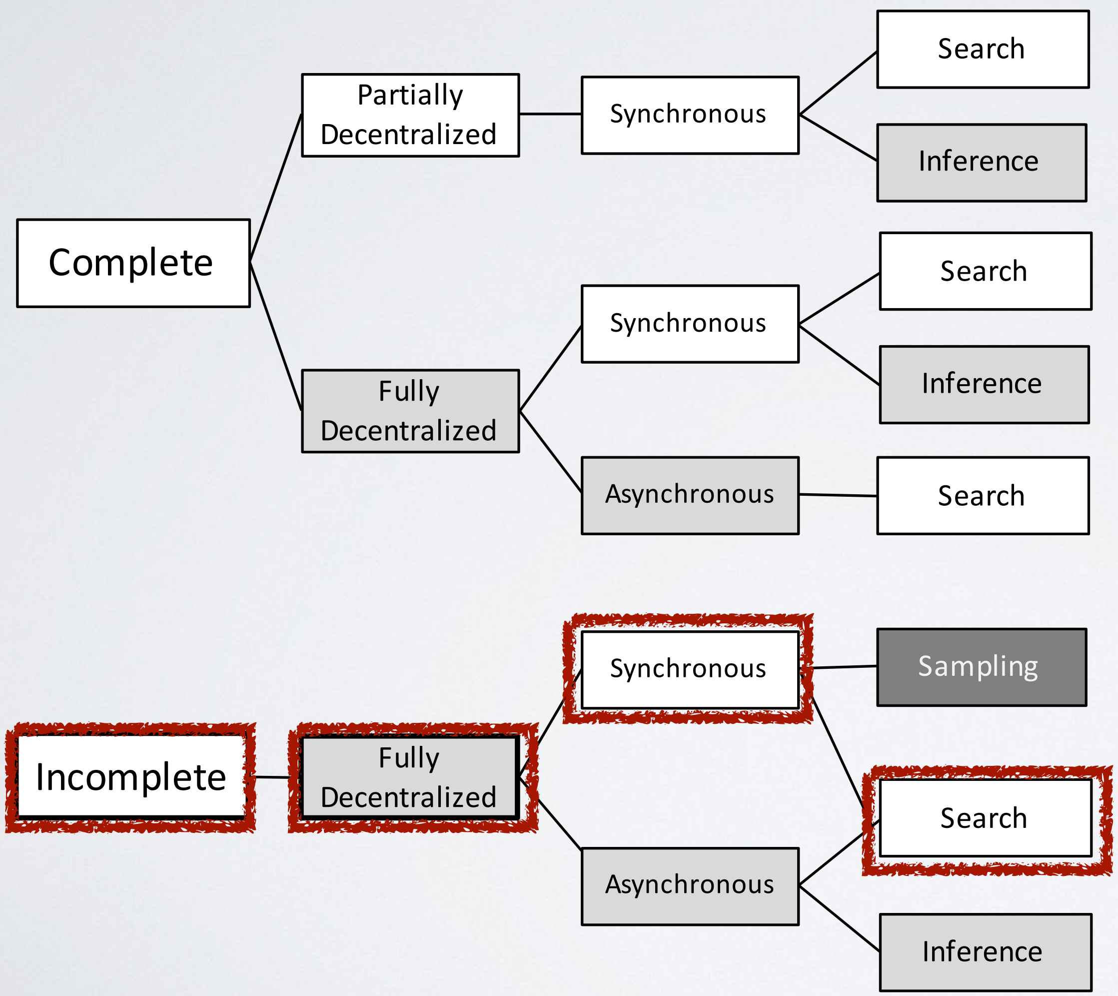
	SBB	DPOP
Correct the solution it finds is optimal	Yes	Yes
Complete it terminates	Yes	Yes
Message Complexity max size of a message	$O(d)$	$O(b^d)$
Network Load max number of messages	$O(b^d)$	$O(d)$
Runtime	$O(b^d)$	$O(b^d)$

branching factor = b
num variables = d

CRITICAL OVERVIEW

Search Algorithms		Inference Algorithms
	increasing memory →	
polynomial		exponential
	← decreasing network load	
exponential		polynomial

DCOP ALGORITHMS

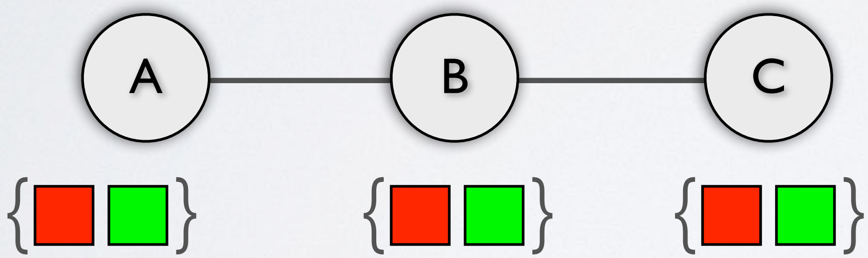


Distributed Local Search

LOCAL SEARCH ALGORITHMS

- DSA: Distributed Stochastic Algorithm
- MGM: Maximum Gain Messages Algorithm

- Note: We now maximize utilities



x_i	x_j	*Utility (A,B)*	*Utility (B,C)*
		5	5
		0	0
		0	0
		8	8

Weixiong Zhang, Guandong Wang, Zhao Xing, Lars Wittenburg: Distributed stochastic search and distributed breakout: properties, comparison and applications to constraint optimization problems in sensor networks. Artif. Intell. 161(1-2): 55-87 (2005)
Rajiv Maheswaran, Jonathan Pearce, Milind Tambe: Distributed Algorithms for DCOP: A Graphical-Game-Based Approach. ISCA PDCS 2004: 432-439

LOCAL SEARCH ALGORITHMS

- DSA: Distributed Stochastic Algorithm
- MGM: Maximum Gain Messages Algorithm

- Every agent individually decides whether to change its value or not
- Decision involves
  - knowing neighbors' values
  - calculation of utility gain by changing values
  - probabilities

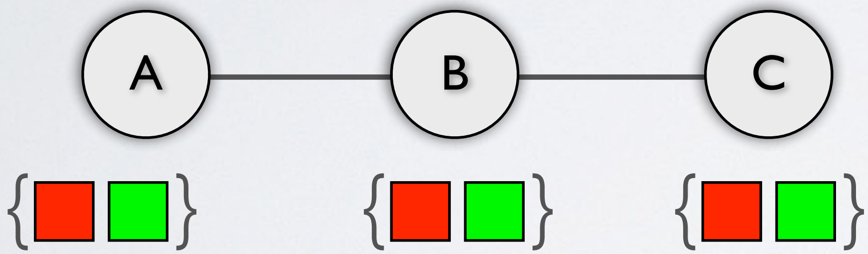
Weixiong Zhang, Guandong Wang, Zhao Xing, Lars Wittenburg: Distributed stochastic search and distributed breakout: properties, comparison and applications to constraint optimization problems in sensor networks. Artif. Intell. 161(1-2): 55-87 (2005)
Rajiv Maheswaran, Jonathan Pearce, Milind Tambe: Distributed Algorithms for DCOP: A Graphical-Game-Based Approach. ISCA PDCS 2004: 432-439

DSA ALGORITHM

- All agents execute the following
  - Randomly choose a value
  - while (termination is not met)
    - if (a new value is assigned)
      - send the new value to neighbors
    - collect neighbors' new values if any
    - select and assign the next value based on assignment rule

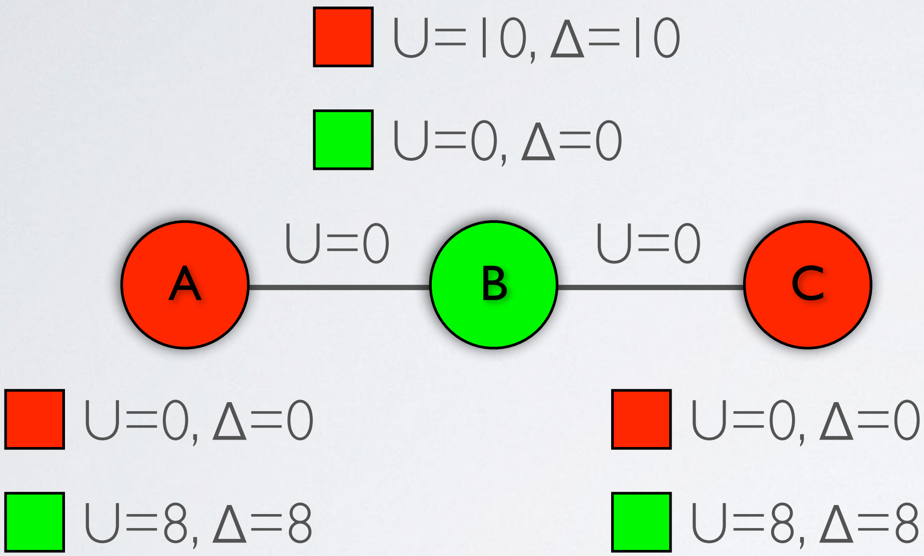
Weixiong Zhang, Guandong Wang, Zhao Xing, Lars Wittenburg: Distributed stochastic search and distributed breakout: properties, comparison and applications to constraint optimization problems in sensor networks. Artif. Intell. 161(1-2): 55-87 (2005)

DSA ALGORITHM



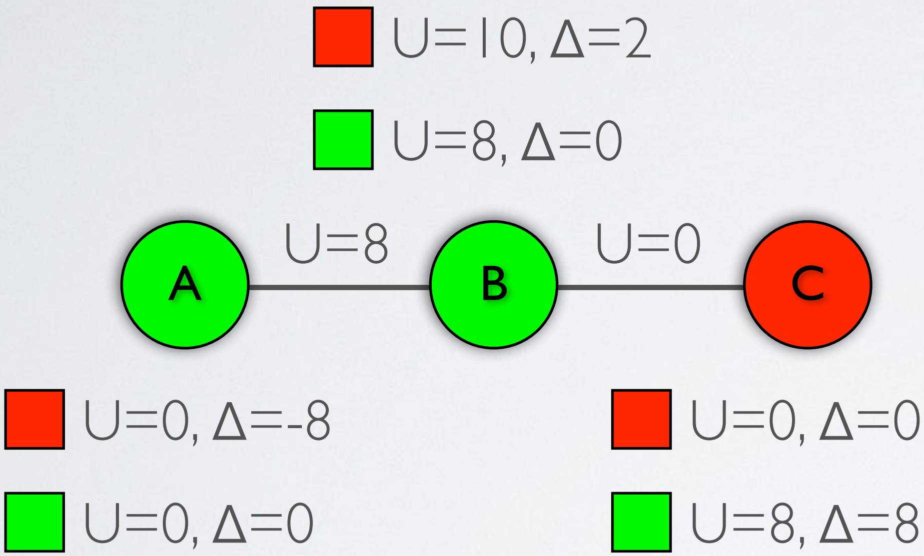
x_i	x_j	*Utility (A,B)*	*Utility (B,C)*
		5	5
		0	0
		0	0
		8	8

DSA ALGORITHM



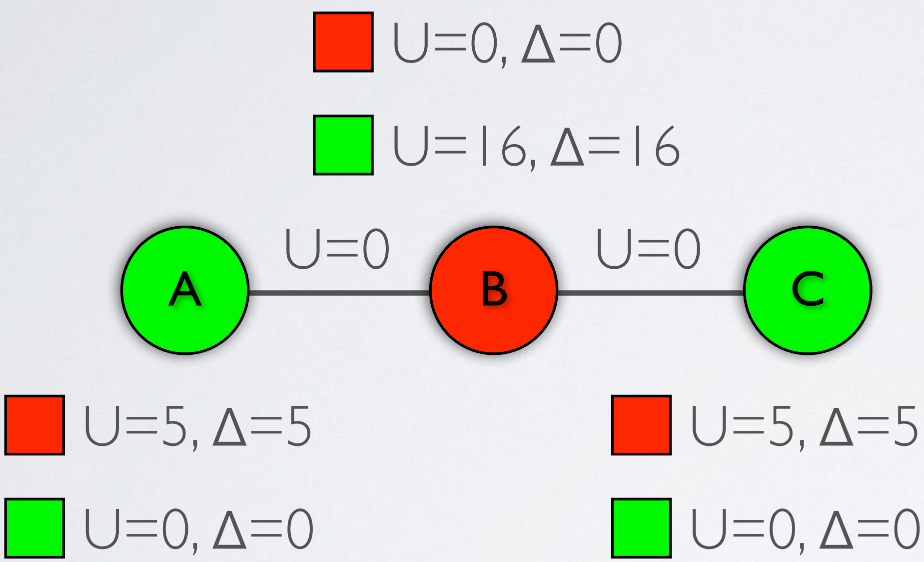
x_i	x_j	*Utility* (A,B)	*Utility* (B,C)
		5	5
		0	0
		0	0
		8	8

DSA ALGORITHM



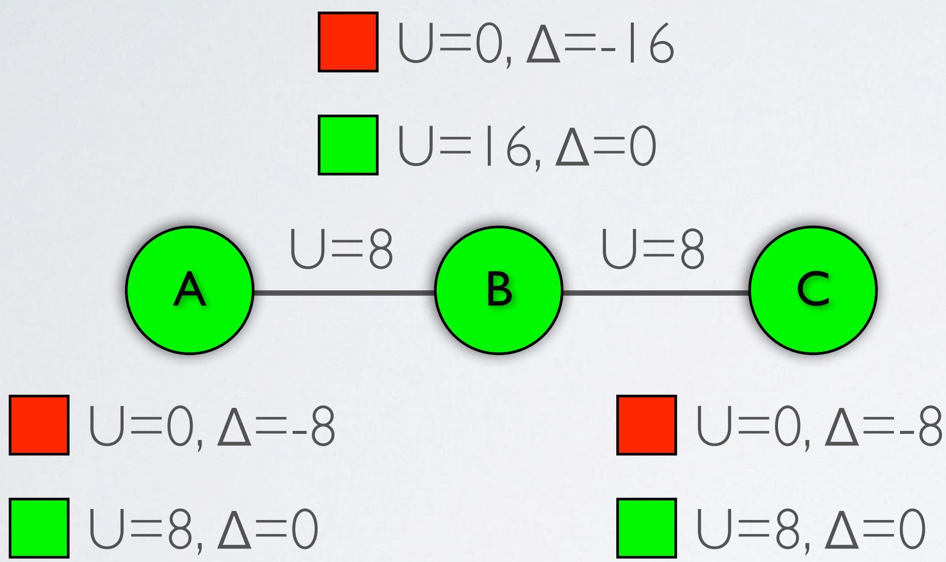
x_i	x_j	*Utility (A,B)*	*Utility (B,C)*
		5	5
		0	0
		0	0
		8	8

DSA ALGORITHM



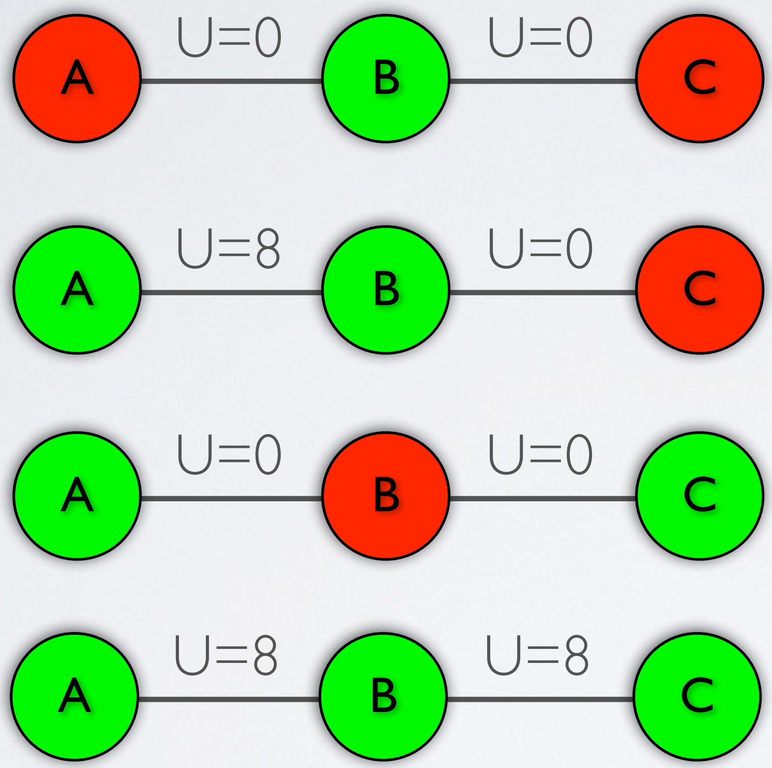
x_i	x_j	*Utility (A,B)*	*Utility (B,C)*
		5	5
		0	0
		0	0
		8	8

DSA ALGORITHM



x_i	x_j	*Utility (A,B)*	*Utility (B,C)*
		5	5
		0	0
		0	0
		8	8

DSA ALGORITHM



One possible execution trace

x_i	x_j	Utility (A,B)	Utility (B,C)
		5	5
		0	0
		0	0
		8	8

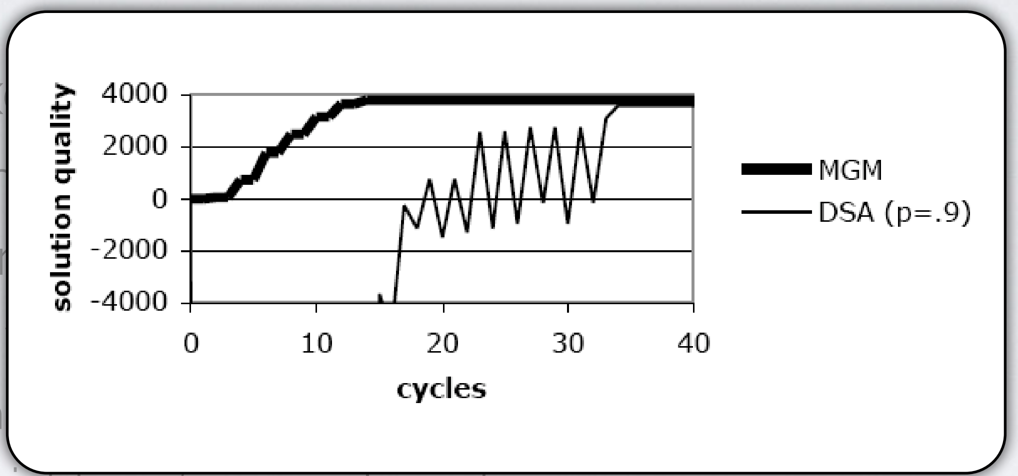
MGM ALGORITHM

- All agents execute the following
  - Randomly choose a value
  - while (termination is not met)
    - if (a new value is assigned)
      - send the new value to neighbors
    - collect neighbors' new values if any
    - *calculate gain and send it to neighbors*
    - *collect neighbors' gains*
    - *if (it has the highest gain among all neighbors)*
      - *change value to the value that maximizes gain*

Rajiv Maheswaran, Jonathan Pearce, Milind Tambe: Distributed Algorithms for DCOP: A Graphical-Game-Based Approach. ISCA PDCS 2004: 432-439

MGM ALGORITHM

- All agents ex
  - Randomly ch
  - while (termi
    - if (a new
      - send th
    - collect neighbors' new values if any



Great if you need an anytime algorithm

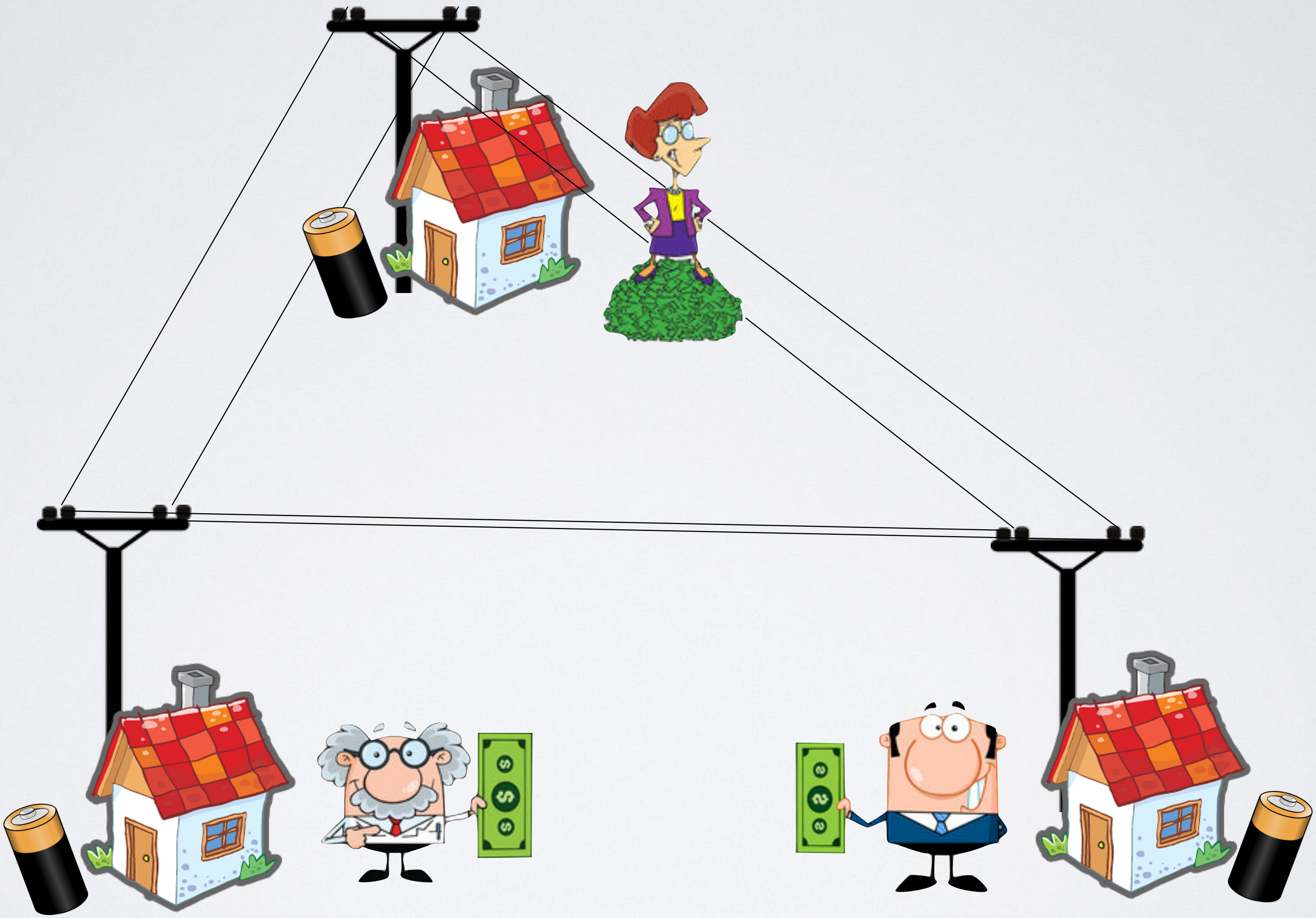
    - *collect neighbors' gains*
    - *if (it has the highest gain among all neighbors)*
      - *change value to the value that maximizes gain*

Rajiv Maheswaran, Jonathan Pearce, Milind Tambe: Distributed Algorithms for DCOP: A Graphical-Game-Based Approach. ISCA PDCS 2004: 432-439

DCOP EXTENSIONS

AAAI-18 Tutorial on
Multi-Agent Distributed Constrained Optimization

PROSUMER ENERGY TRADING

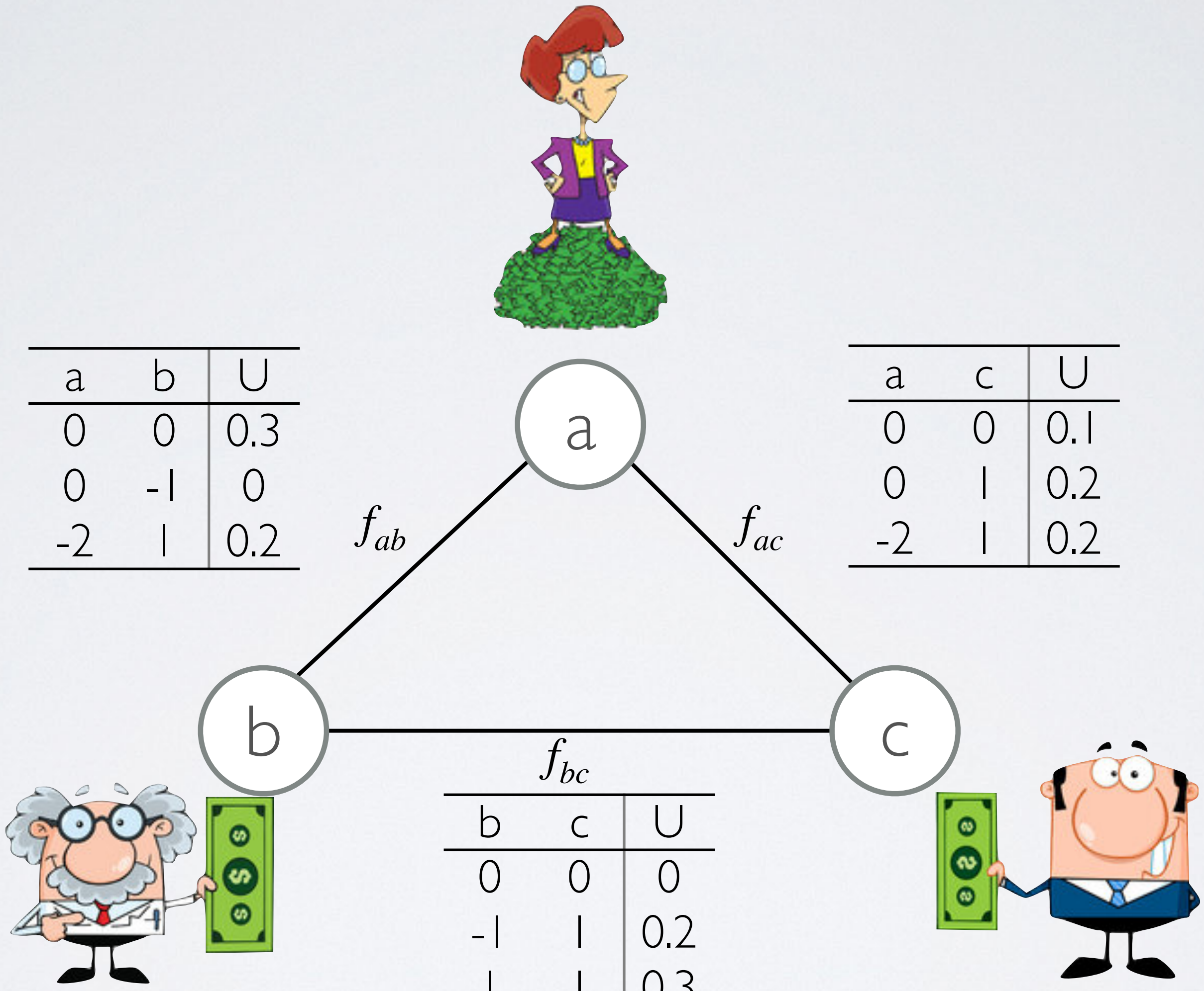


Designing a Marketplace for the Trading and Distribution of Energy in the Smart Grid. AAMAS 2015: 1285-1293

PROSUMER ENERGY TRADING

- Prosumers: capable of both generating and consuming resources
- Each prosumer can sell or buy a given amount of power to another prosumer
- Line capacity and flow constraints are required to be satisfied
- Each offer has a desired utility
- Goal: Find a buy/selling assignment that maximizes the actors' rewards and is feasible with the operating power constraints

PROSUMER ENERGY TRADING



a	b	U
0	0	0.3
0	-1	0
-2	1	0.2

a	c	U
0	0	0.1
0	1	0.2
-2	1	0.2

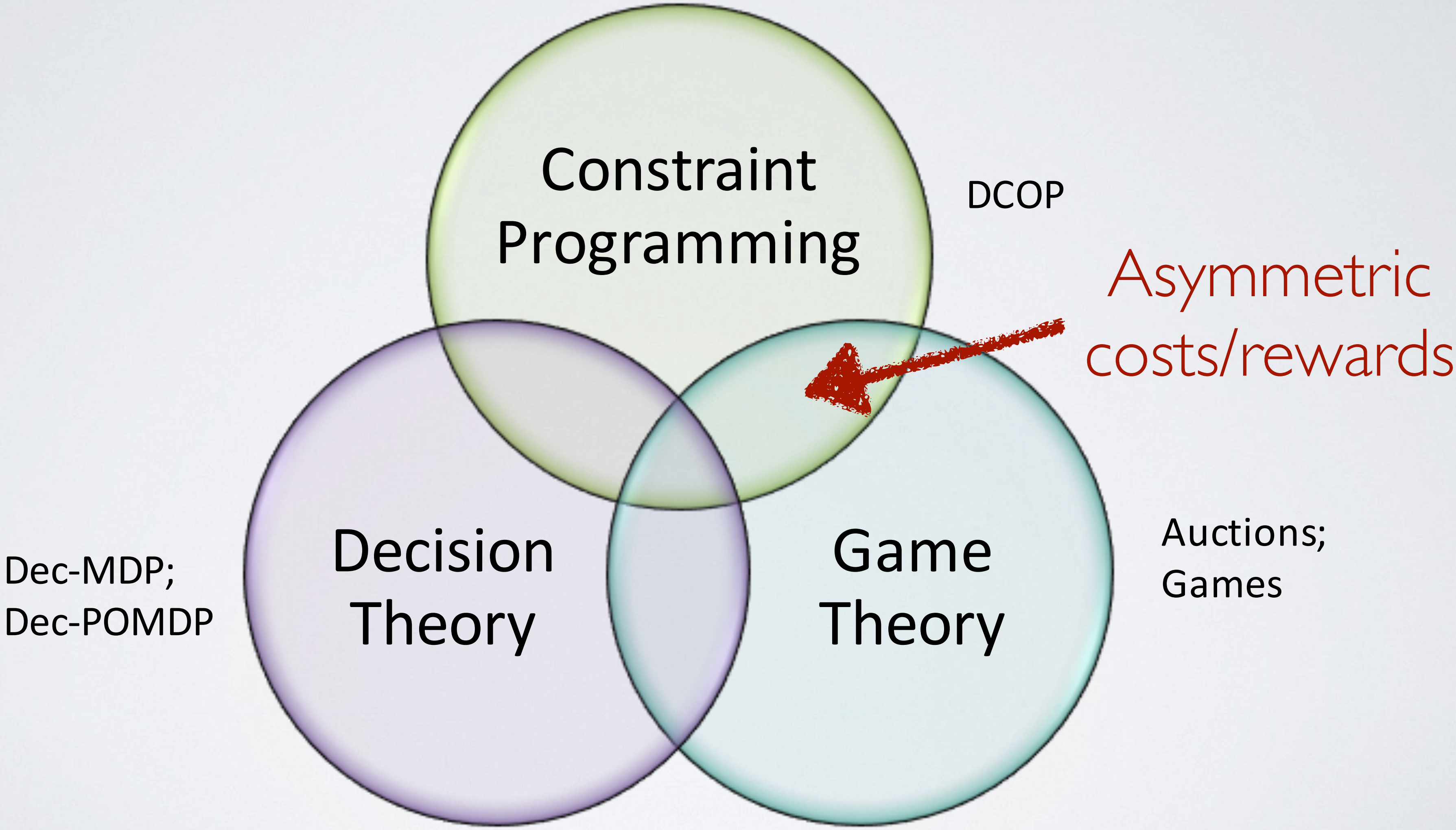
b	c	U
0	0	0
-1	1	0.2
1	1	0.3

Designing a Marketplace for the Trading and Distribution of Energy in the Smart Grid. AAMAS 2015: 1285-1293

PROSUMER ENERGY TRADING

- What if Alice cannot disclose the costs associated her action?
- What if we want to describe the scenario in which
  - Bob desires to gain 0.2 for selling1 KW of power to Carl
  - Car desires to gain 0.1 for buying 1 KW of power from Bob?

ASYMMETRIC DCOP



ASYMMETRIC DCOP

- Asymmetric DCOPs are DCOPs where:
- A joint assignment may produce different costs for the agents participating in a constraint

A	*B*	*Cost*
r	g	3
r	g	2
g	r	10
g	g	0

ASYMMETRIC DCOP

- Asymmetric DCOPs are DCOPs where:
- A joint assignment may produce different costs for the agents participating in a constraint

A	*B*	*Cost A*	*Cost B*
r	g	2	1
r	g	0	2
g	r	3	7
g	g	0	0

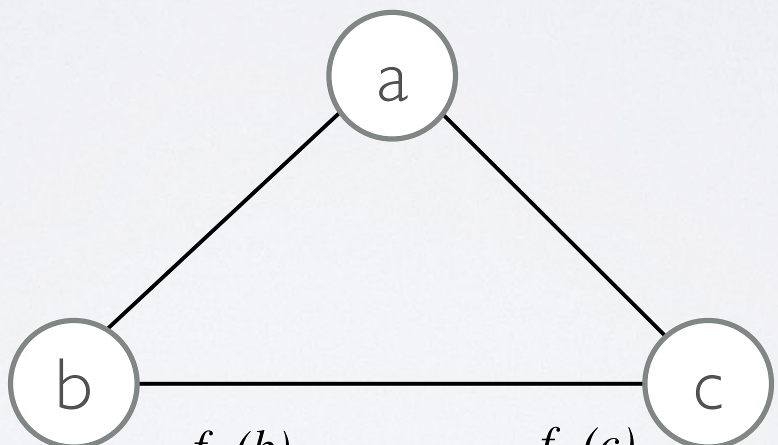
PROSUMER ENERGY TRADING

$f_{ab}(a)$

a	b	U
0	0	0.2
0	-1	0
-2	1	0.2

$f_{ac}(a)$

a	c	U
0	0	0.1
0	1	0.1
-2	1	0.2



$f_{ab}(b)$

a	b	U
0	0	0.1
0	-1	0.1
-2	1	0

$f_{ac}(c)$

a	c	U
0	0	1
0	1	0
-2	1	0

$f_{bc}(b)$

a	c	U
0	0	0
-1	1	0.2
-2	1	0.1

$f_{bc}(c)$

a	c	U
0	0	0
-1	1	0.1
-2	1	0.2

ASYMMETRIC DCOP

- Why asymmetric DCOPs?
  - …

ASYMMETRIC DCOP

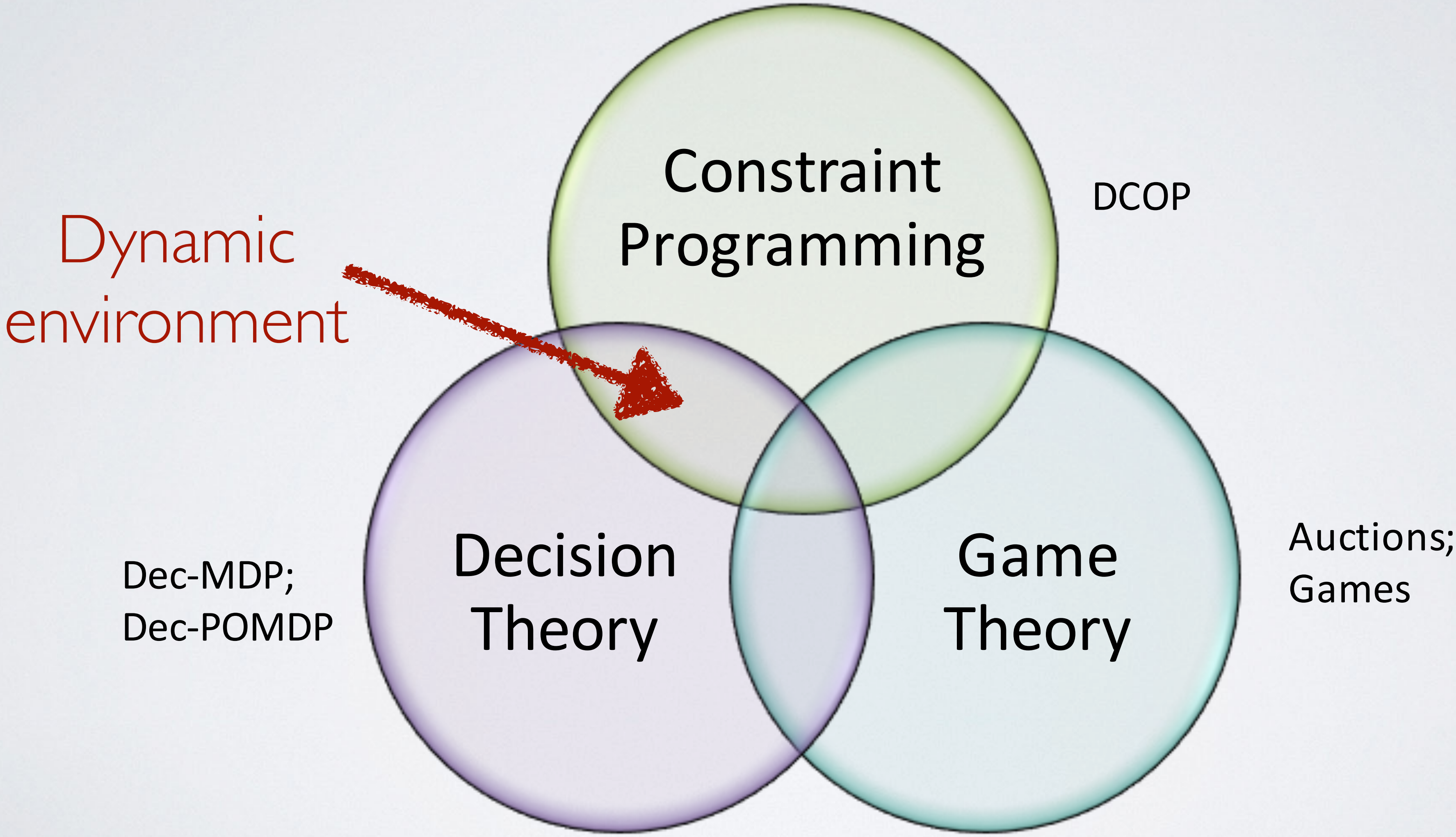
- Why asymmetric DCOPs?
  - Models richer forms of cooperation
  - Privacy: Agents do not need to reveal the costs associated to their action
  - Resource allocation problems:
    - Different costs for using the same resource
    - Different preferences

PROSUMER ENERGY TRADING

- What if a new prosumer would like to join the market?

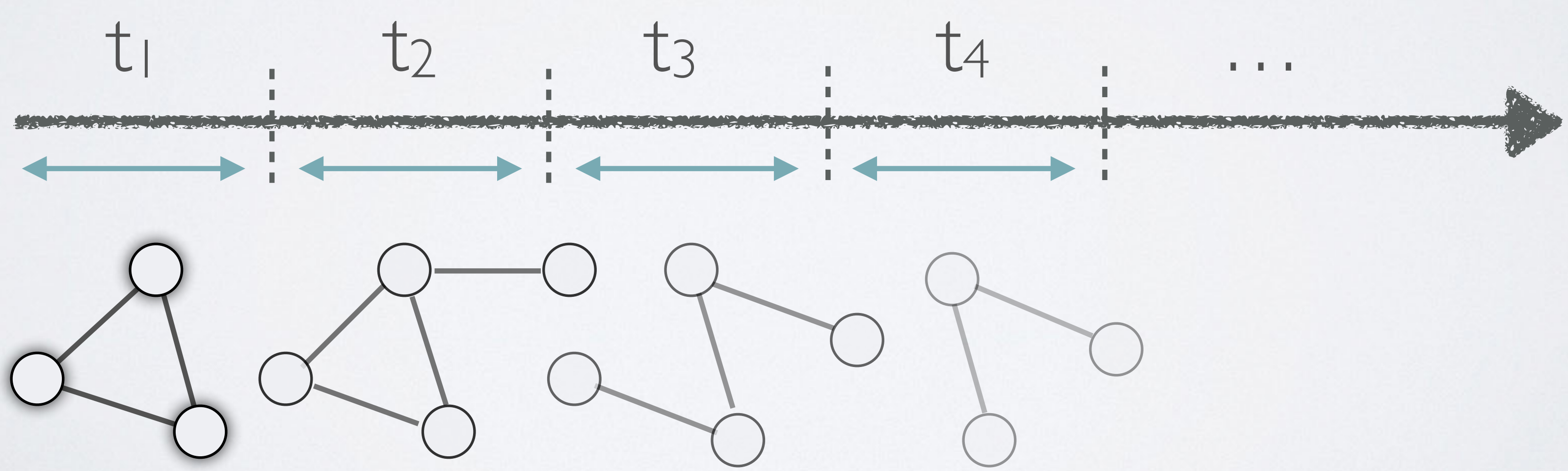
- What if a prosumer would like to modify her preferences?

DYNAMIC DCOP



DYNAMIC DCOP

- A Dynamic DCOP is sequence $P_1, P_2, \ldots, P_k$ of k DCOPs
- The agent knowledge about the environment is confined within each time step
- Each DCOP is solved sequentially



DYNAMIC DCOP

- Why dynamic DCOPs?
  - …

DYNAMIC DCOP

- Why dynamic DCOPs?
  - MAS commonly exhibit dynamic environments
  - The capture scenarios with:
    - Moving agents, change of constraints, change of preferences
    - Additional information become available during problem solving
  - Application domains: Sensor networks, cloud computing, smart home automation, …

APPLICATIONS

AAAI-18 Tutorial on
Multi-Agent Distributed Constrained Optimization

DCOP APPLICATIONS

- Scheduling Problems
  - Taking DCOP to the Real World: Efficient Complete Solutions for Distributed Multi-Event Scheduling. AAMAS 2004
- Radio Frequency Allocation Problems
  - Improving DPOP with Branch Consistency for Solving Distributed Constraint Optimization Problems. CP 2014
- Sensor Networks
  - Preprocessing techniques for accelerating the DCOP algorithm ADOPT. AAMAS 2005
- Home Automation
  - A Multiagent System Approach to Scheduling Devices in Smart Homes. AAMAS 2017, IJCAI 2016
- Traffic Light Synchronization
  - Evaluating the performance of DCOP algorithms in a real world, dynamic problem. AAMAS 2008
- Disaster Evacuation
  - Disaster Evacuation Support. AAAI 2007; JAIR 2017
- Combinatorial Auction Winner Determination
  - H-DPOP: Using Hard Constraints for Search Space Pruning in DCOP. AAAI 2008

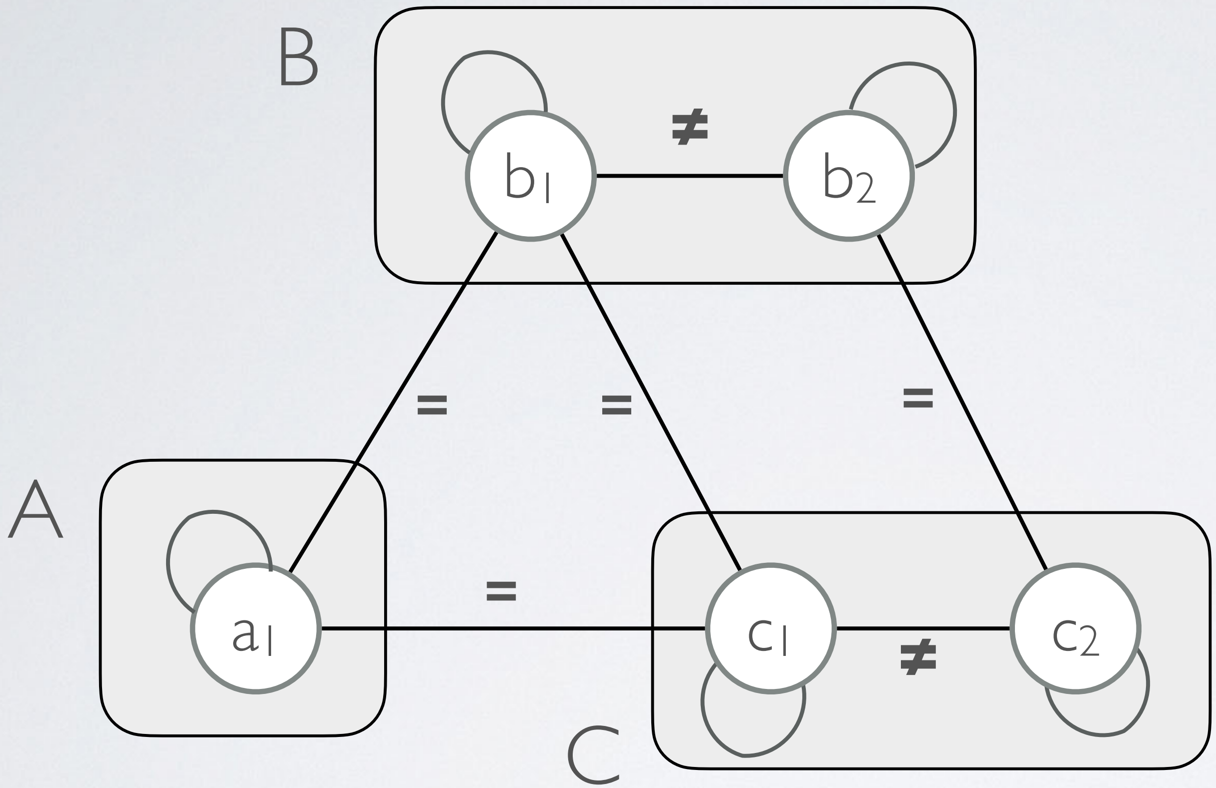
MEETING SCHEDULING

- Meeting 1: Alice, Bob, Carl
- Meeting 2: Bob, Carl
- …

- Alice is only free in the mornings from 9am-noon
- Bob prefers to not meet during lunch (noon-1pm)
- Carl does not wake up until 11am and loves late evening meetings
- …

Rajiv T. Maheswaran, Milind Tambe, Emma Bowring, Jonathan P. Pearce, Pradeep Varakantham: Taking DCOP to the Real World: Efficient Complete Solutions for Distributed Multi-Event Scheduling. AAMAS 2004: 310-317

MEETING SCHEDULING

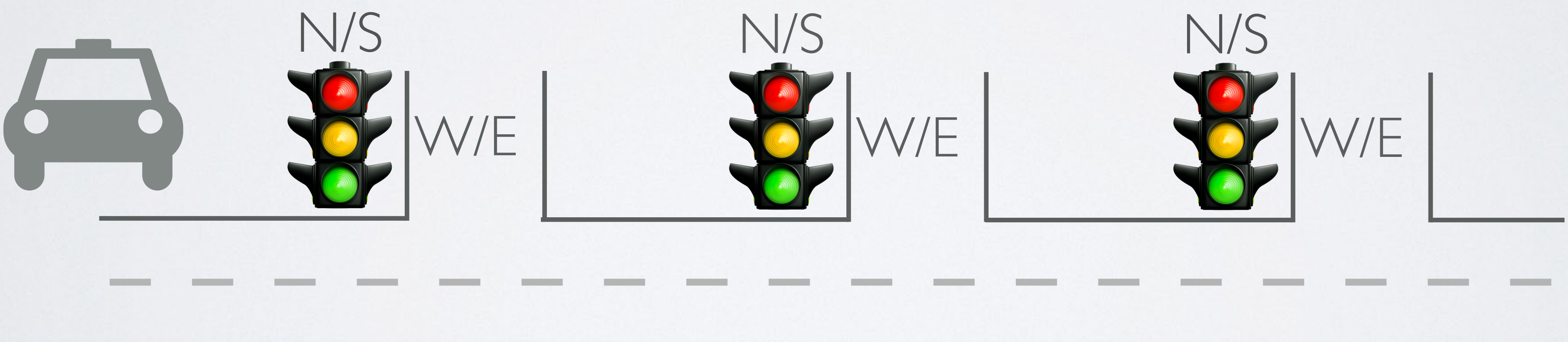


- Values: time slots to hold the meetings
- All agents participating in a meeting must meet at the same time
- All meetings of an agent must occur at different times

Rajiv T. Maheswaran, Milind Tambe, Emma Bowring, Jonathan P. Pearce, Pradeep Varakantham: Taking DCOP to the Real World: Efficient Complete Solutions for Distributed Multi-Event Scheduling. AAMAS 2004: 310-317

TRAFFIC FLOW CONTROL

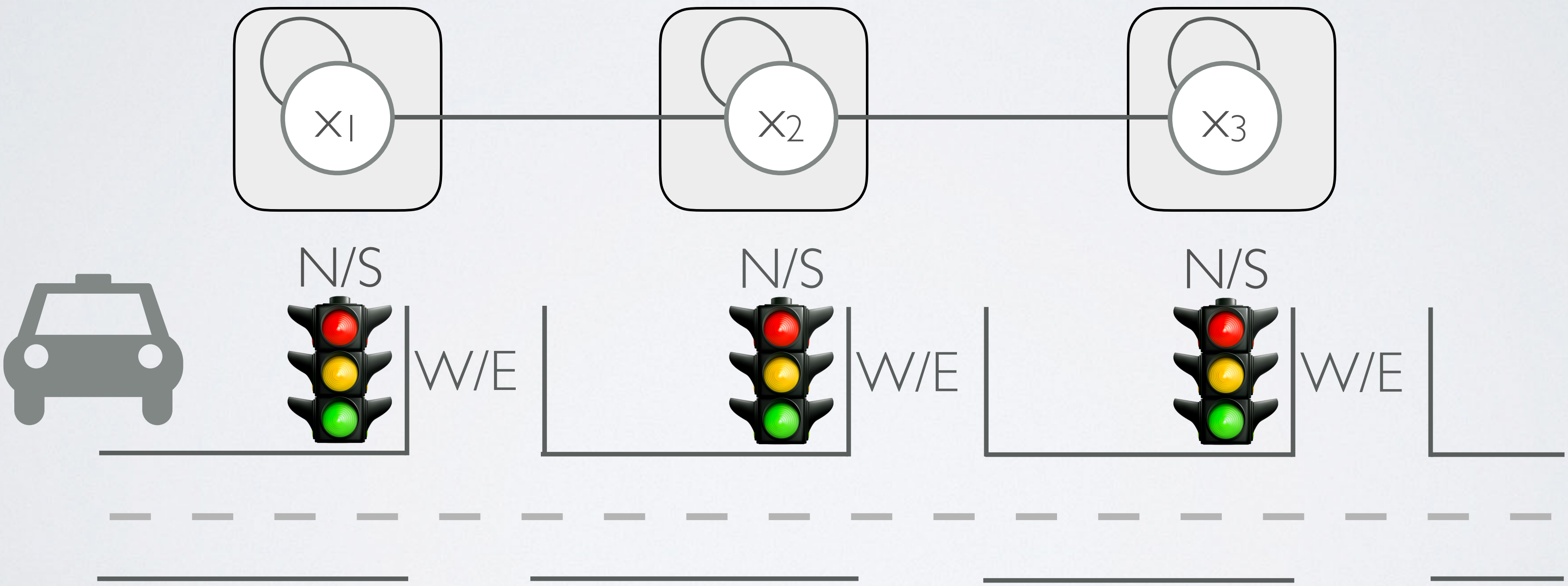
- Given a set of traffic lights in adjacent intersections
- How coordinate them to create green waves?



de Oliveira, D., Bazzan, A. L., & Lesser, V.. Using cooperative mediation to coordinate traffic lights: a case study. AAMAS 2005: 371-378
Junges, R., & Bazzan, A. L. Evaluating the performance of DCOP algorithms in a real world, dynamic problem. AAMAS 2008: 463-470

TRAFFIC FLOW CONTROL

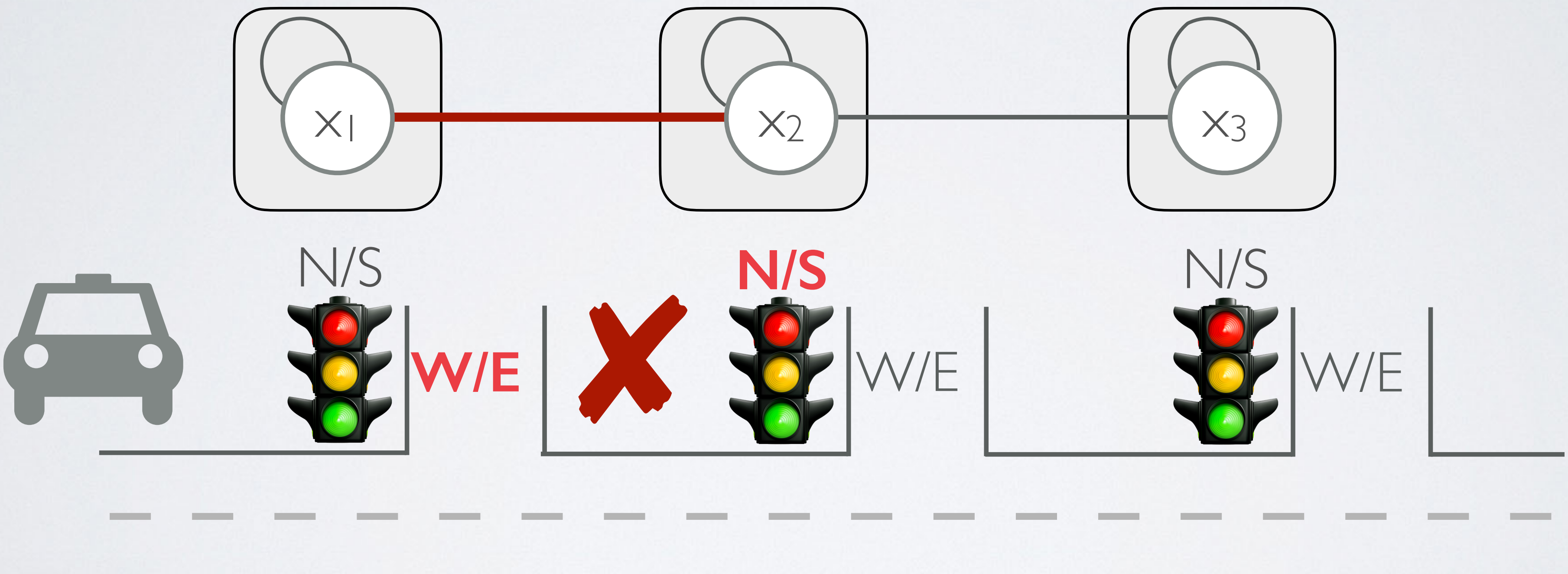
- Agents: Each traffic light
- Values: Flow traffic direction



de Oliveira, D., Bazzan, A. L., & Lesser, V.. Using cooperative mediation to coordinate traffic lights: a case study. AAMAS 2005: 371-378
Junges, R., & Bazzan, A. L. Evaluating the performance of DCOP algorithms in a real world, dynamic problem. AAMAS 2008: 463-470

TRAFFIC FLOW CONTROL

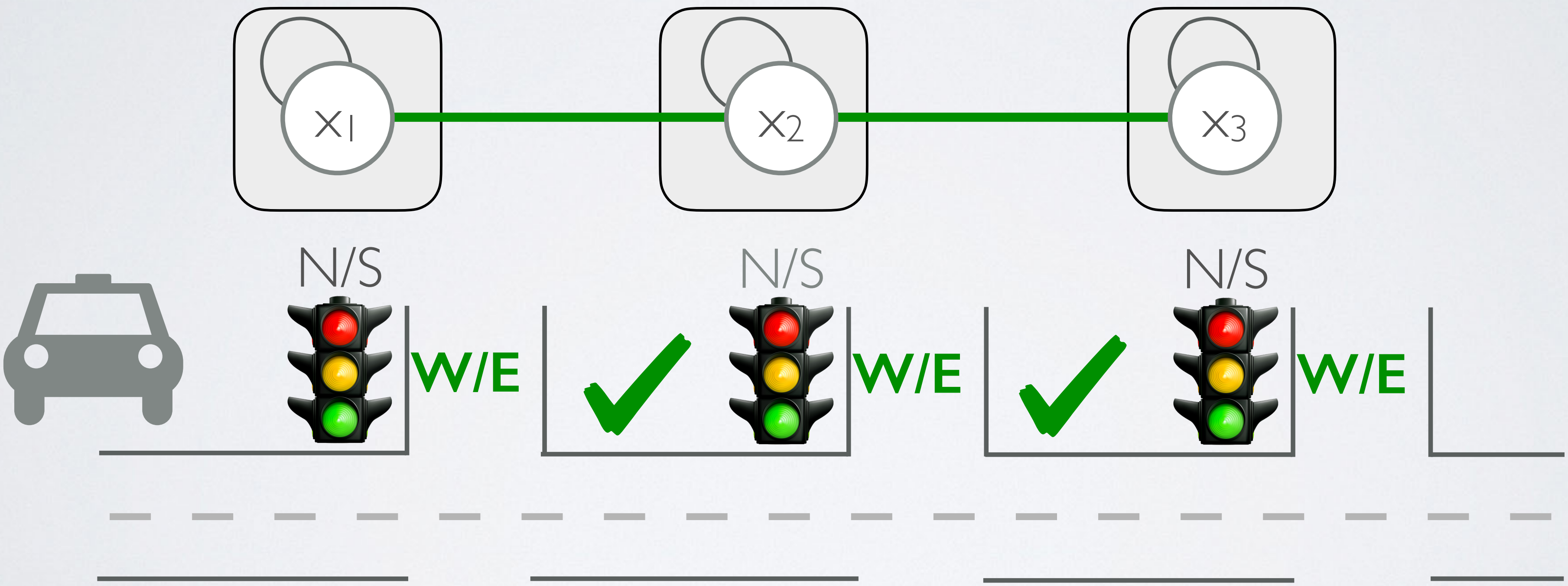
- Agents: Each traffic light
- Values: Flow traffic direction
- Conflict if 2 neighboring signals choose different directions



de Oliveira, D., Bazzan, A. L., & Lesser, V.. Using cooperative mediation to coordinate traffic lights: a case study. AAMAS 2005: 371-378
Junges, R., & Bazzan, A. L. Evaluating the performance of DCOP algorithms in a real world, dynamic problem. AAMAS 2008: 463-470

TRAFFIC FLOW CONTROL

- Cost functions model the number of incoming vehicles
- Maximize the traffic flow

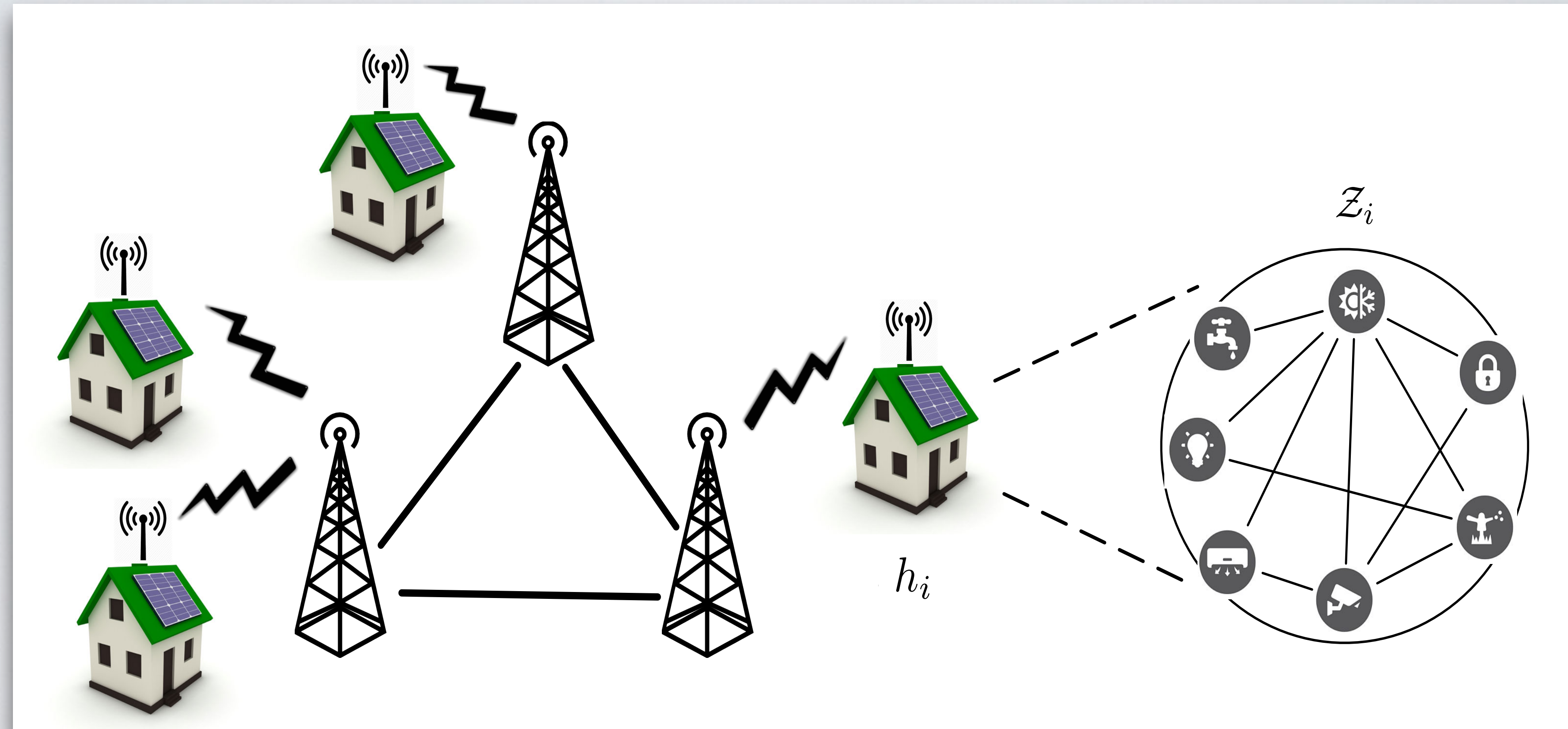


de Oliveira, D., Bazzan, A. L., & Lesser, V.. Using cooperative mediation to coordinate traffic lights: a case study. AAMAS 2005: 371-378
Junges, R., & Bazzan, A. L. Evaluating the performance of DCOP algorithms in a real world, dynamic problem. AAMAS 2008: 463-470

SMART DEVICES

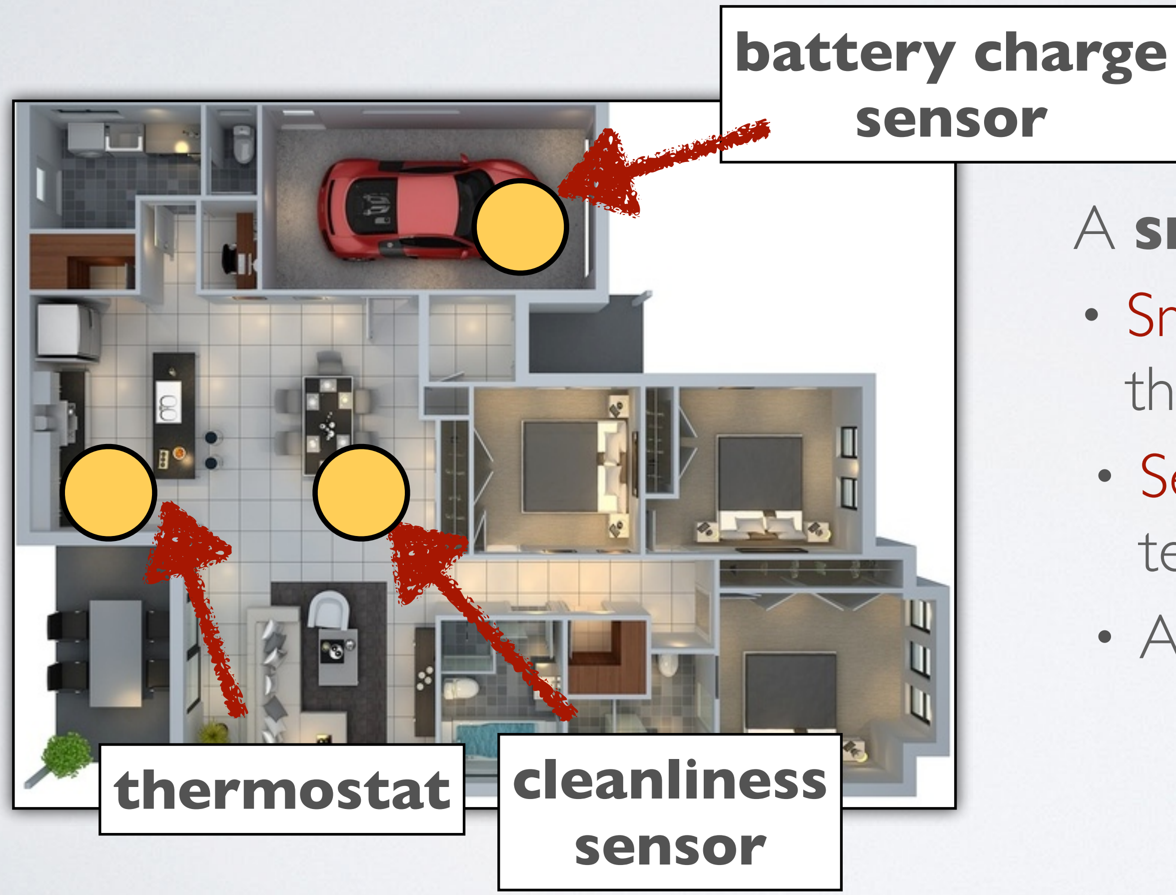
HOME ASSISTANTS

SMART HOMES DEVICE SCHEDULING



Ferdinando Fioretto, William Yeoh, Enrico Pontelli. "A Multiagent System Approach to Scheduling Devices in Smart Homes". AAMAS, 2017.

SMART HOMES DEVICE SCHEDULING



A **smart home** has:

- Smart devices (roomba, HVAC) that it can control
- Sensors (cleanliness, temperature)
- A set of locations

Ferdinando Fioretto, William Yeoh, Enrico Pontelli. "A Multiagent System Approach to Scheduling Devices in Smart Homes". AAMAS, 2017.

SMART HOMES DEVICE SCHEDULING

Smart device:

- A set of actions it can perform (clean, charge)
- Power consumption associated to each action.

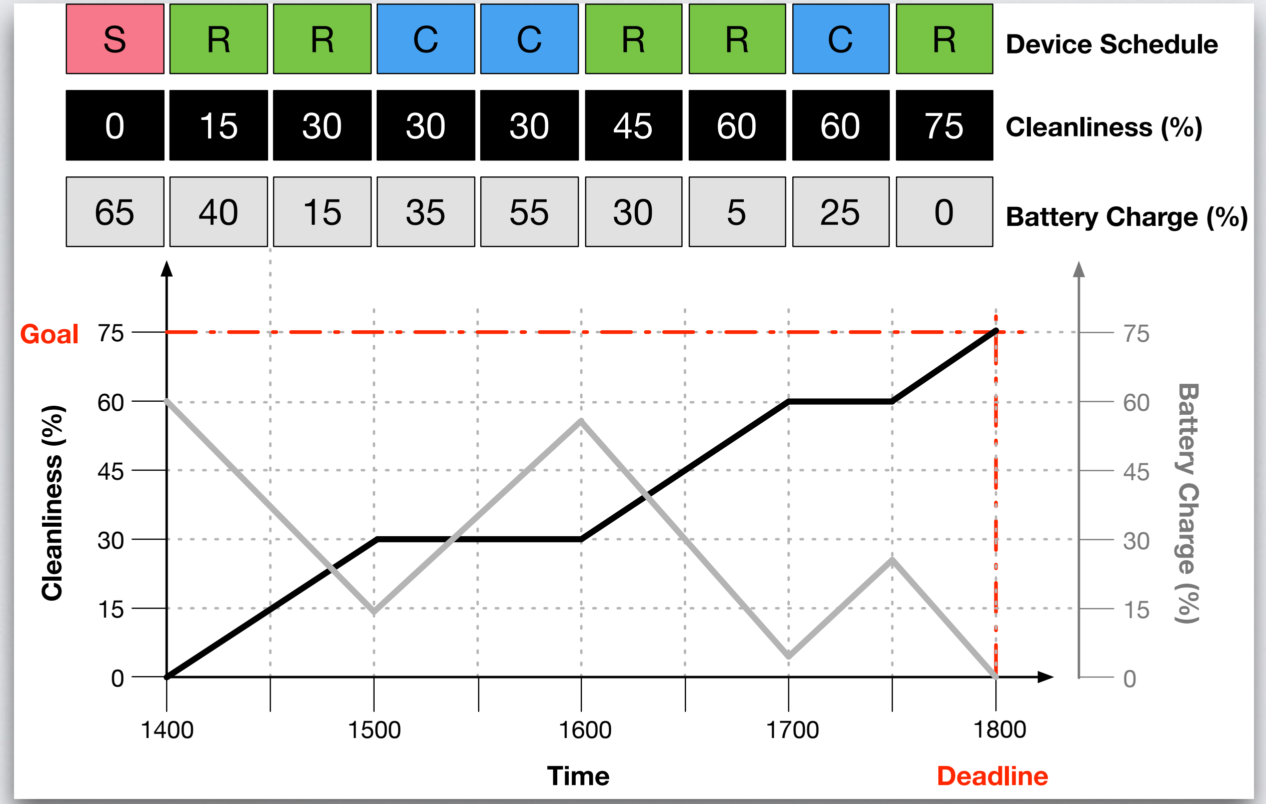
Scheduling Rules:

`living_room cleanliness ≥ 75 before 1800`

Ferdinando Fioretto, William Yeoh, Enrico Pontelli. "A Multiagent System Approach to Scheduling Devices in Smart Homes". AAMAS, 2017.

SMART HOMES DEVICE SCHEDULING

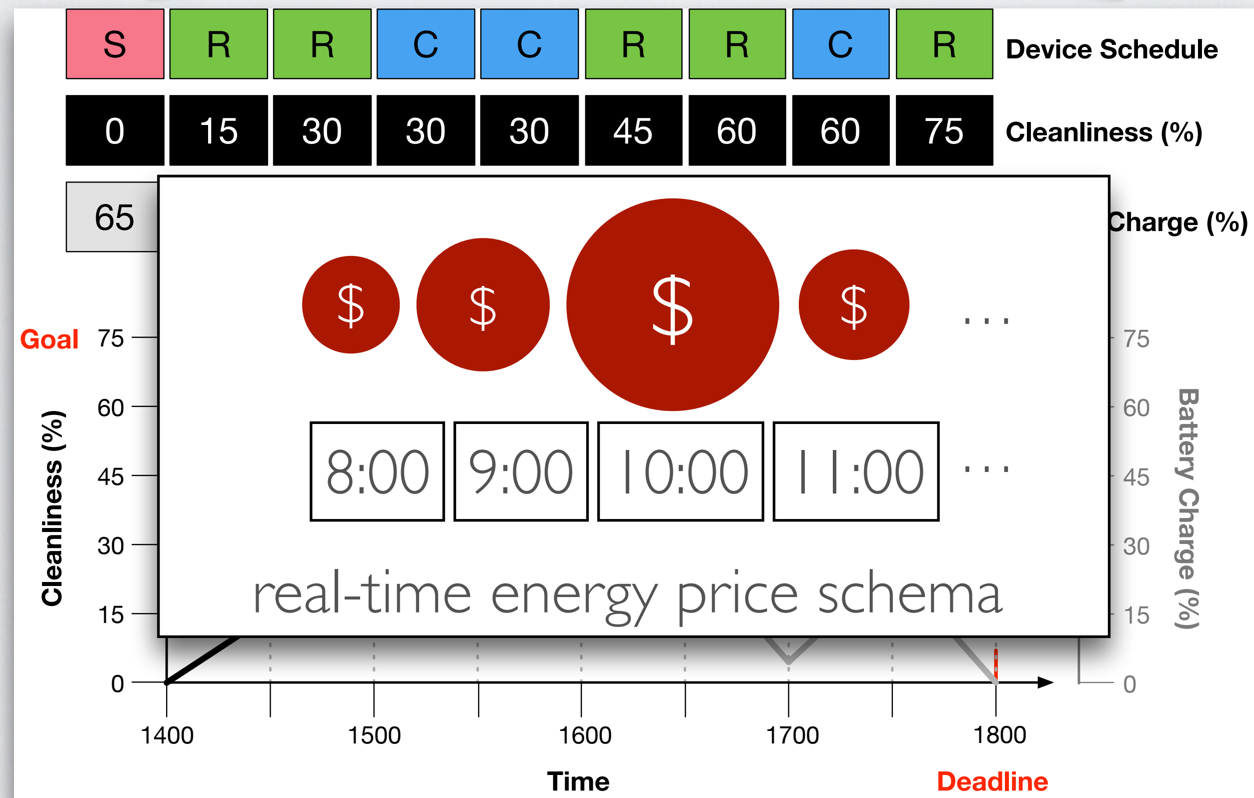
`living_room cleanliness ≥ 75 before 1800`



Ferdinando Fioretto, William Yeoh, Enrico Pontelli. "A Multiagent System Approach to Scheduling Devices in Smart Homes". AAMAS, 2017.

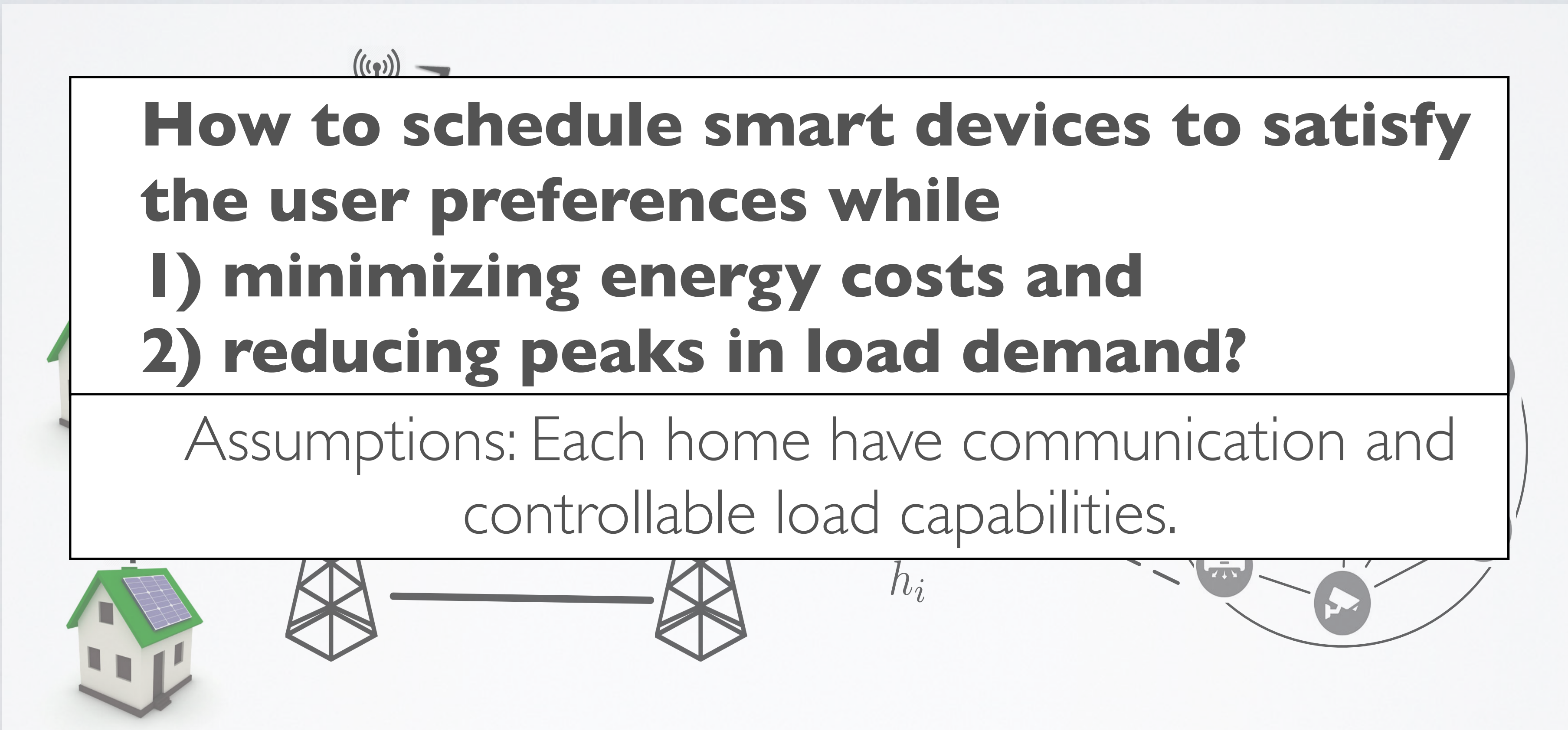
SMART HOMES DEVICE SCHEDULING

`living_room cleanliness ≥ 75 before 1800`



Ferdinando Fioretto, William Yeoh, Enrico Pontelli. "A Multiagent System Approach to Scheduling Devices in Smart Homes". AAMAS, 2017.

SMART HOMES DEVICE SCHEDULING

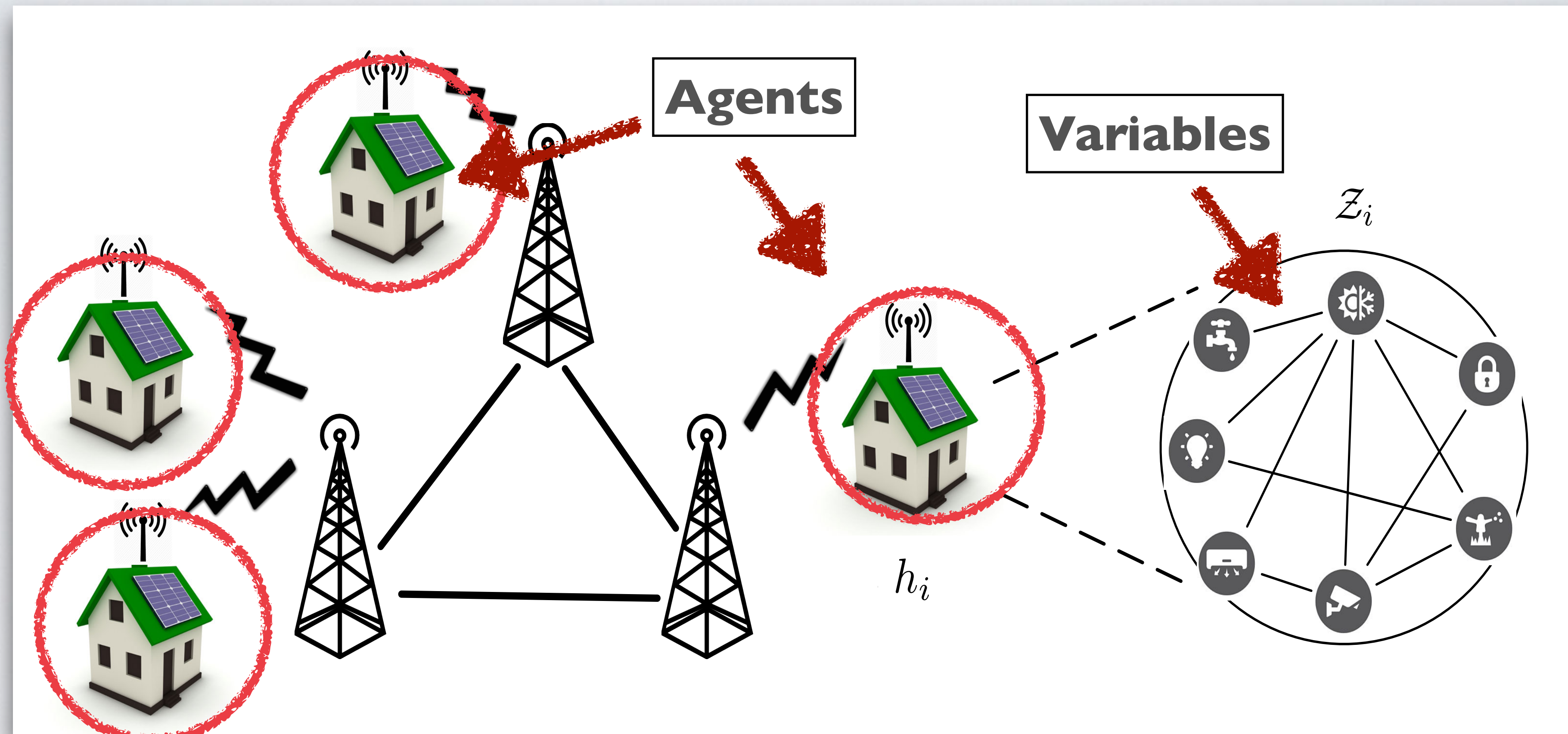


How to schedule smart devices to satisfy the user preferences while
1) minimizing energy costs and
2) reducing peaks in load demand?

Assumptions: Each home have communication and controllable load capabilities.

Ferdinando Fioretto, William Yeoh, Enrico Pontelli. "A Multiagent System Approach to Scheduling Devices in Smart Homes". AAMAS, 2017.

SMART HOMES DEVICE SCHEDULING



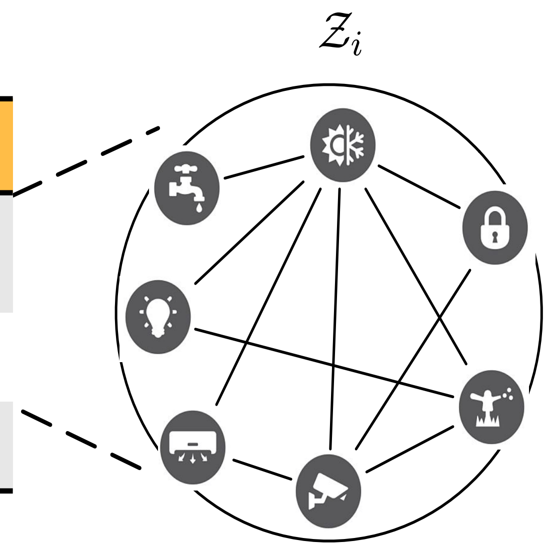
Ferdinando Fioretto, William Yeoh, Enrico Pontelli. "A Multiagent System Approach to Scheduling Devices in Smart Homes". AAMAS, 2017.

SMART HOMES DEVICE SCHEDULING

Domains

Environmental variables

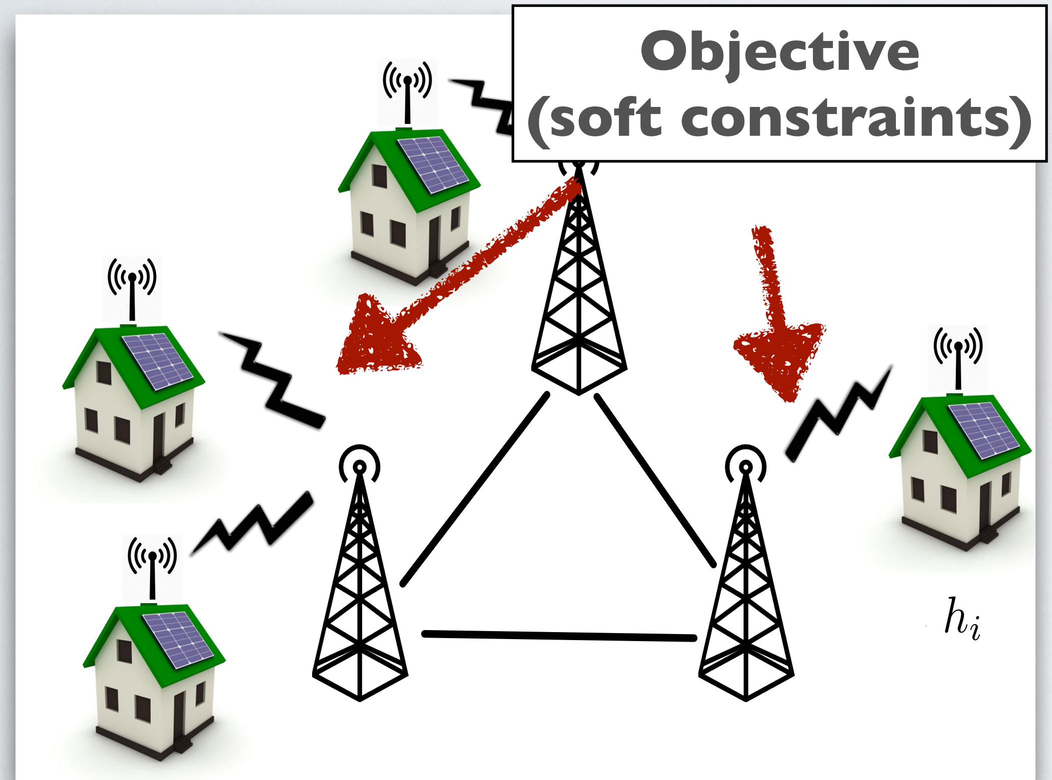
Action	State property	Power (kW/h)
run	cleanliness, battery charge	0.0
charge	battery charge	0.26
stop		0.0

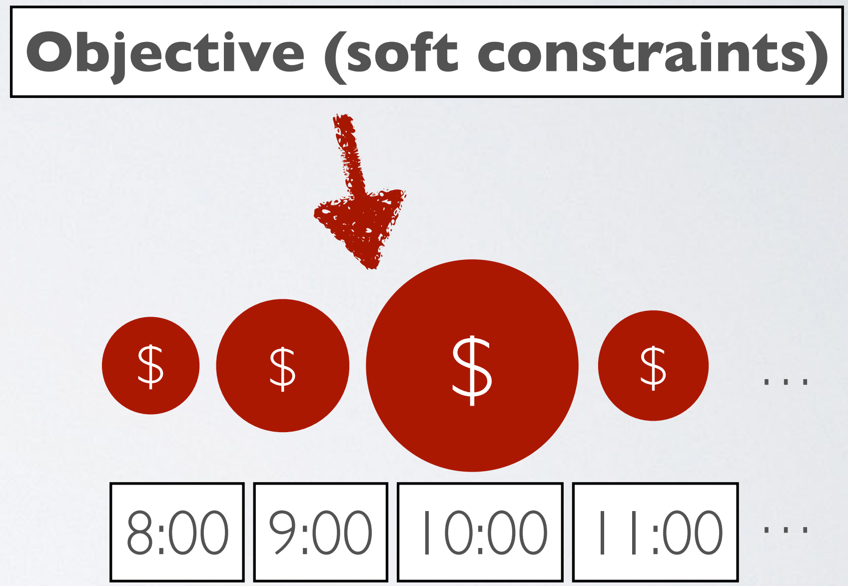


`living_room cleanliness ≥ 75 before 1800`

Hard constraints

SMART HOMES DEVICE SCHEDULING





real-time energy price schema

Ferdinando Fioretto, William Yeoh, Enrico Pontelli. "A Multiagent System Approach to Scheduling Devices in Smart Homes". AAMAS, 2017.

CHALLENGES AND OPEN QUESTIONS

AAAI-18 Tutorial on
Multi-Agent Distributed Constrained Optimization

MAS DECISION MAKING

- Decentralized decision making difficult due to:
  - Large number of interacting entities

MAS DECISION MAKING

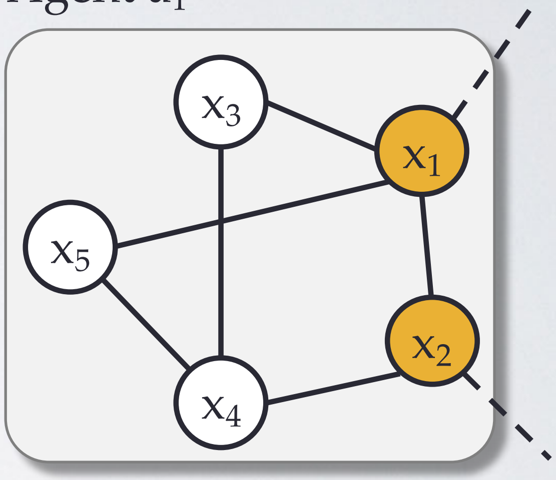
- Decentralized decision making difficult due to:
  - Large number of interacting entities
  - Can we use **decomposition techniques** to reduce the amount of interactions?
  - Can we create **hierarchical models** to increase parallelism and efficiency?

- Bistaffa, Farinelli, Bombieri. "Optimising memory management for belief propagation in junction trees using GPGPUs ", ICPADS 2014
- Ferdinando Fioretto, William Yeoh, Enrico Pontelli. "Multi-Variable Agent Decomposition for DCOPs". In Proceedings of the AAAI Conference on Artificial Intelligence (AAAI), 2016.
- Ferdinando Fioretto, Hong Xu, Sven Koenig, TK Satish Kumar. "Constraint Composite Graph-Based Lifted Message Passing for Distributed Constraint Optimization Problems". In International Symposium on Artificial Intelligence and Mathematics (ISAIM), 2018.
- Ferdinando Fioretto, Enrico Pontelli, William Yeoh, Rina Dechter. "Accelerating Exact and Approximate Inference for (Distributed) Discrete Optimization with GPUs". In Constraints, 2018.

MAS DECISION MAKING

- Decentralized decision making difficult due to:
  - Large number of interacting entities
  - Can we use **decomposition techniques** to reduce the amount of interactions?
  - Can we create **hierarchical models** to increase parallelism and efficiency?

Agent a_1



- Bistaffa, Farinelli, Bombieri. "Optimising memory management for belief propagation in junction trees using GPGPUs ", ICPADS 2014
- Ferdinando Fioretto, William Yeoh, Enrico Pontelli. "Multi-Variable Agent Decomposition for DCOPs". In Proceedings of the AAAI Conference on Artificial Intelligence (AAAI), 2016.
- Ferdinando Fioretto, Hong Xu, Sven Koenig, TK Satish Kumar. "Constraint Composite Graph-Based Lifted Message Passing for Distributed Constraint Optimization Problems". In International Symposium on Artificial Intelligence and Mathematics (ISAIM), 2018.
- Ferdinando Fioretto, Enrico Pontelli, William Yeoh, Rina Dechter. "Accelerating Exact and Approximate Inference for (Distributed) Discrete Optimization with GPUs". In Constraints, 2018.

DYNAMIC ENVIRONMENT

- Interaction in a dynamic environment is required to be robust to several changes

- R. Mailler, H. Zheng, and A. Ridgway. 2017. Dynamic, distributed constraint solving and thermodynamic theory. Auton Agent Multi-Agent Syst (2017).
- Zhang, C., & Lesser, V. (2013). Coordinating multi-agent reinforcement learning with limited communication. In Proceedings of the International Conference on Autonomous Agents and Multiagent Systems (AAMAS), pp. 1101–1108.

DYNAMIC ENVIRONMENT

- Interaction in a dynamic environment is required to be robust to several changes
  - How do agents respond to dynamic changes?
  - Can we study adaptive algorithms so that the MAS interaction is resilient and adaptive to changes in the communication layer, the underlying constraint graph, etc.?

- R. Mailler, H. Zheng, and A. Ridgway. 2017. Dynamic, distributed constraint solving and thermodynamic theory. Auton Agent Multi-Agent Syst (2017).
- Zhang, C., & Lesser, V. (2013). Coordinating multi-agent reinforcement learning with limited communication. In Proceedings of the International Conference on Autonomous Agents and Multiagent Systems (AAMAS), pp. 1101–1108.

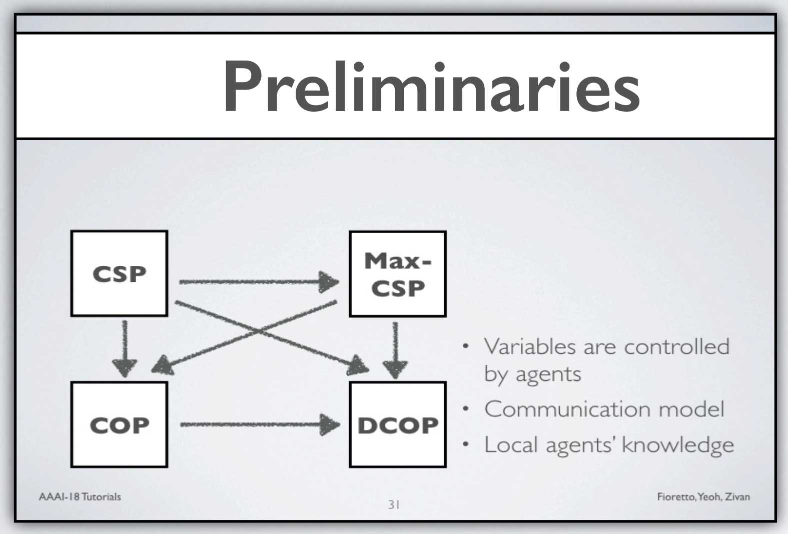
AGENT PREFERENCES

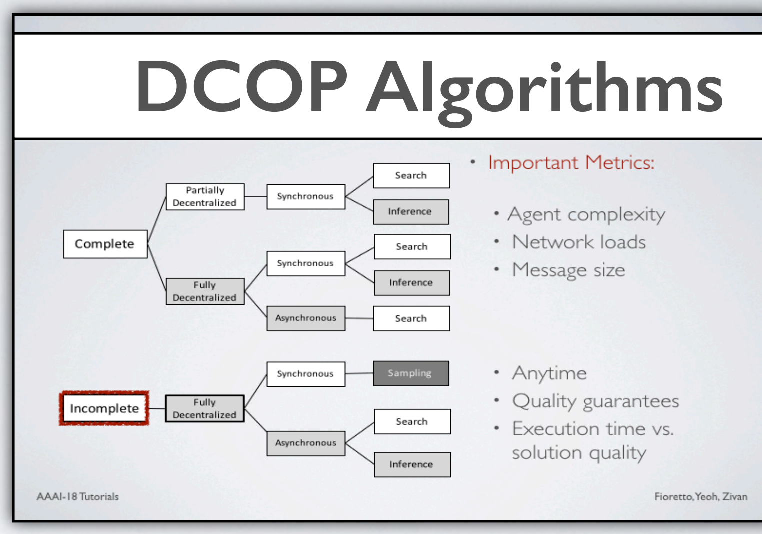
- How to model, learn, and update agent preferences?

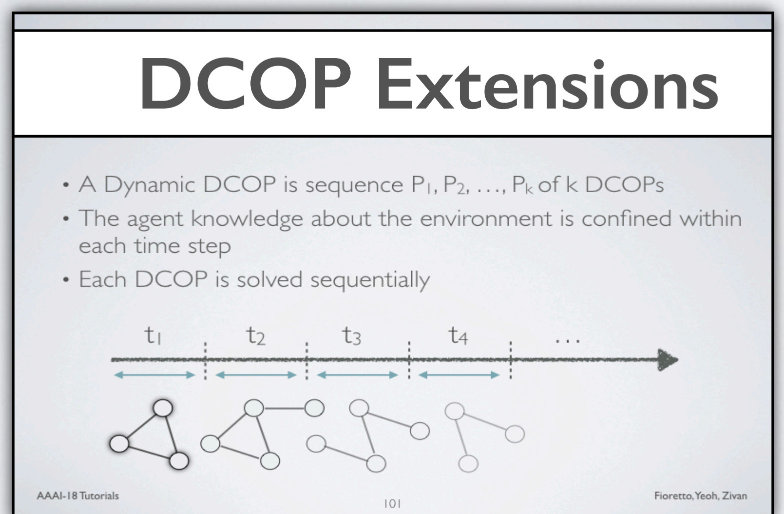
AGENT PREFERENCES

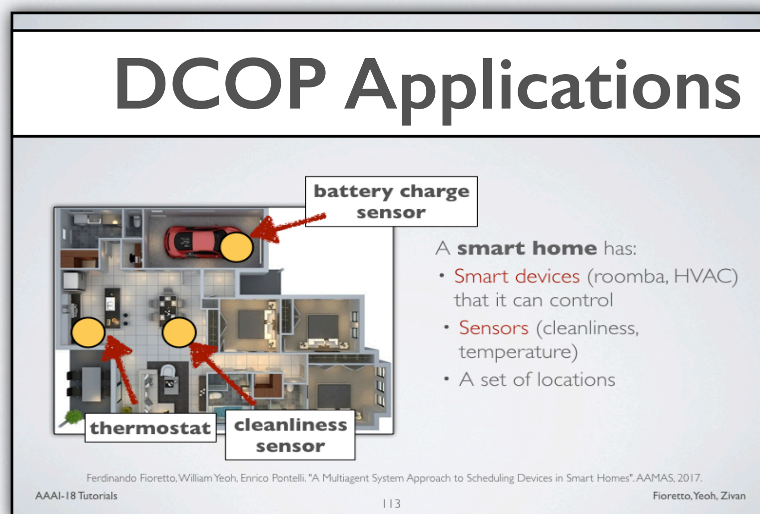
- How to model, learn, and update agent preferences?
  - Agent's preferences are assumed to be available. This is not always feasible. How to efficiently elicit agents' preferences?
  - When full elicitation is not possible, how to adaptively learn the preference of an agent?

- Atena M. Tabakhi, Tiep Le, Ferdinando Fioretto, and William Yeoh. "Preference Elicitation for DCOPs." In Proceedings of the International Conference on Principles and Practice of Constraint Programming (CP), pages 278-296, 2017









THANK YOU!

Ferdinando Fioretto, William Yeoh, Roie Zivan