RÉPUBLIQUE FRANÇAISE
Liberté
Égalité
Fraternité

ONERA
THE FRENCH AEROSPACE LAB

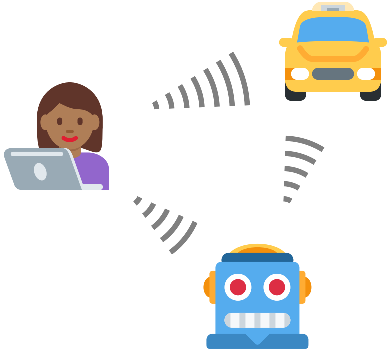
Multi-agent Coordination using Distributed Constraint Optimization and Auction-based Techniques

Gauthier Picard
gauthier.picard@onera.fr

Tutorial at PFIA – 01/07/2025

AILab — ONERA/DTIS, Université de Toulouse

- 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

Decision Making

Decision Making

x_i ?

Mono-Agent Decision Making

x_i ?

« I'm satisfied with x_i »

Multi-Agent Decision Making

x_i ?

« I'm satisfied with x_i »

x_j ?

« agent i agrees with agent j »

Multi-Agent Decision Making

x_i ?

« I'm satisfied with x_i »

x_j ?

« agent i agrees with agent j »

How agents can make their decisions in an autonomous and coordinated manner?

Multi-Agent Decision Making

x_i ?

« I'm satisfied with x_i »

x_j ?

« agent i agrees with agent j »

How agents can make their decisions in an autonomous and coordinated manner?

$\Rightarrow$ Cooperative decentralized decision making

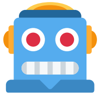
Focus on Cooperative Settings

Decentralized Decision Making

- Agents have to **coordinate** to perform best actions
- Cooperative settings
  - Agents form a team → **best actions** for the team

Sample Applications

- Surveillance (target tracking, coverage)
- Robotics (cooperative exploration)
- Autonomous vehicles (cooperative traffic management)
- Scheduling (meeting scheduling, EOS scheduling)
- Rescue Operation (task assignment)



About Distribution and Decentralization

If cooperative, why not centralizing decision making?

About Distribution and Decentralization

If cooperative, why not centralizing decision making?
$\Rightarrow$ **autonomy** (🤖) + **privacy** (🔒)

About Distribution and Decentralization

If cooperative, why not centralizing decision making?
$\Rightarrow$ **autonomy** (🤖) + **privacy** (🔒)

Why distribution might not be sufficient?

About Distribution and Decentralization

If cooperative, why not centralizing decision making?
$\Rightarrow$ **autonomy** (🤖) + **privacy** (🔒)

Why distribution might not be sufficient?
$\Rightarrow$ **autonomy** (🤖) + **privacy** (🔒) + **robustness** (🛡)

About Distribution and Decentralization

If cooperative, why not centralizing decision making?
$\Rightarrow$ **autonomy** (🤖) + **privacy** (🔒)

Why distribution might not be sufficient?
$\Rightarrow$ **autonomy** (🤖) + **privacy** (🔒) + **robustness** (🛡)

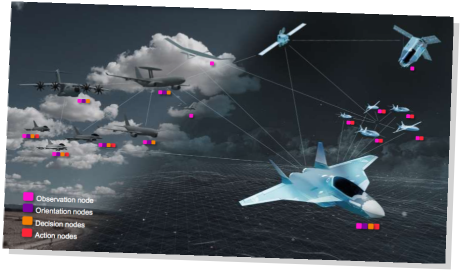
Decentralization

Introduction

Sample multi-agent systems

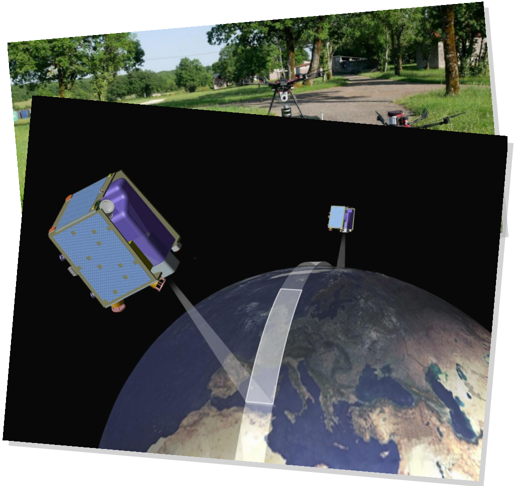
Introduction

Sample multi-agent systems



Introduction

Sample multi-agent systems



Introduction

Sample multi-agent systems

Introduction

Sample multi-agent systems

Expected Takeaway

- Modeling frameworks
- Algorithms
- Illustrative problems and applications

Today's Menu

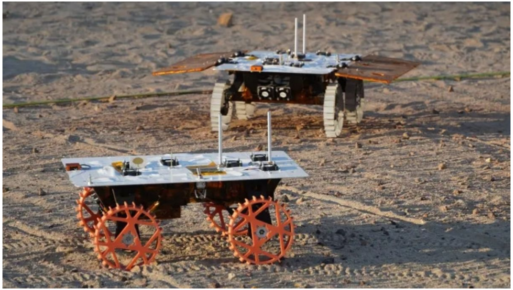
Today's Menu

Multi-Robot Task Allocation

Definition (MRTA)

- A set of **agents** (robots, satellites, etc.), $R = \{r_1, \ldots, r_{|R|}\}$ with capabilities
- A set of **tasks**, $T = \{t_1, \ldots, t_{|T|}\}$, with time-related and operation constraints and requirements
- Find an assignment of tasks to agents, wrt. some consistency **constraints**
  - e.g. capabilities, dependencies between tasks, resource capacity, plan consistency

  whilst optimizing some specific **objective**
  - e.g. completion time, energy



Mission CADRE – ©NASA

Who does **what** (when and in what order) ?

Multi-Robot Task Allocation

Simple Problem Formulation

$$
\begin{aligned}
\max_{\mathbf{x}} \quad & \sum_{i=1}^{n}\sum_{j=1}^{m} u_{ij}x_{ij} \\
\text{subject to} \quad & \sum_{j=1}^{m} x_{ij} \leq 1, \quad \forall i \in \{1, ..., n\} \\
& \sum_{i=1}^{n} x_{ij} \leq 1, \quad \forall j \in \{1, ..., m\} \\
& x_{ij} \in \{0, 1\}, \quad \forall i, j \\
\text{with} \quad & u_{ij} \text{ utility for robot } i \text{ executing task } j, \quad \forall i, j
\end{aligned}
$$

Multi-Robot Task Allocation

Simple Problem Formulation

$$
\begin{aligned}
\max_{\mathbf{x}} \quad & \sum_{i=1}^{n}\sum_{j=1}^{m} u_{ij}x_{ij} \\
\text{subject to} \quad & \sum_{j=1}^{m} x_{ij} \leq 1, \quad \forall i \in \{1,...,n\} \\
& \sum_{i=1}^{n} x_{ij} \leq 1, \quad \forall j \in \{1,...,m\} \\
& x_{ij} \in \{0,1\}, \quad \forall i,j \\
\text{with} \quad & u_{ij} \text{ utility for robot } i \text{ executing task } j, \quad \forall i,j
\end{aligned}
$$

NP-hard, requires advanced optimization methods

Multi-Robot Task Allocation

Classification and Solution Methods

Classification

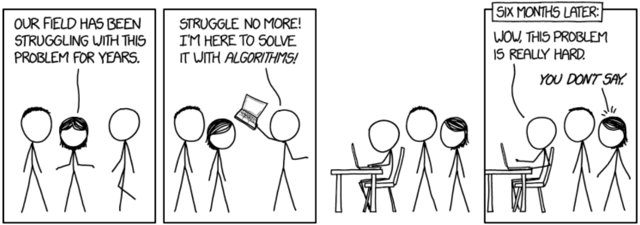
[Shiroma and Campos, 2009]

- Instantaneous (IA) vs. Time-Extended (TA) Allocation
- Single-Type (ST) vs. Multi-Type (MT) Robot Scenarios
- Single-Task (SR) vs. Multi-Task (MR) Request Scenarios

Solution Methods

[Chakraa et al., 2023; Shelkamy et al., 2020]

- Integer Linear Programming (ILP)
- Metaheuristics (e.g., Simulated Annealing, Genetic Algorithms)
- Distributed and Decentralized Approaches [Quinton et al., 2023]
- Machine Learning-based Methods



Here to Help ©XKCD – CC BY-NC 2.5

Multi-Robot Task Allocation

Classification and Solution Methods

Classification

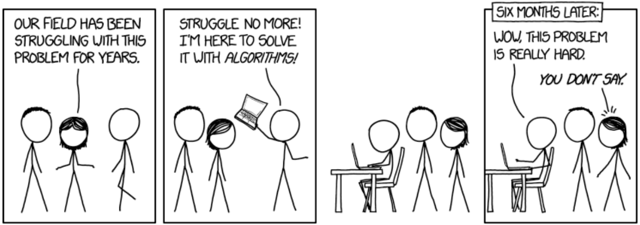
[Shiroma and Campos, 2009]

- Instantaneous (IA) vs. Time-Extended (TA) Allocation
- Single-Type (ST) vs. Multi-Type (MT) Robot Scenarios
- Single-Task (SR) vs. Multi-Task (MR) Request Scenarios

Solution Methods

[Chakraa et al., 2023; Shelkamy et al., 2020]

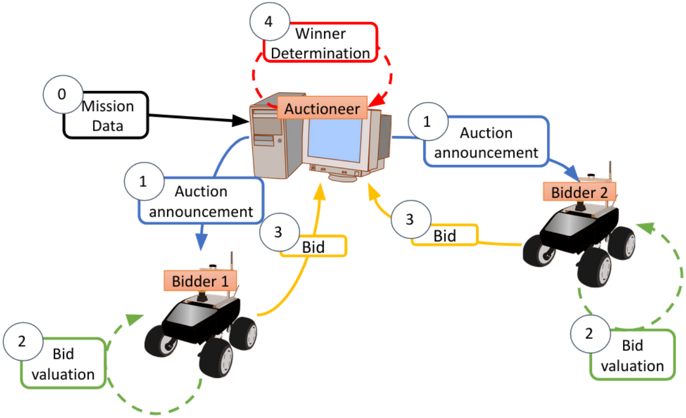
- Integer Linear Programming (ILP)
- Metaheuristics (e.g., Simulated Annealing, Genetic Algorithms)
- Distributed and Decentralized Approaches
  [Quinton et al., 2023]
- Machine Learning-based Methods



Here to Help ©XKCD – CC BY-NC 2.5

Multi-Robot Task Allocation

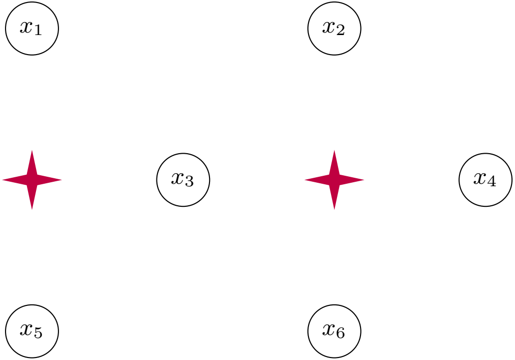
Distributed and Decentralized Algorithms



Today’s Menu

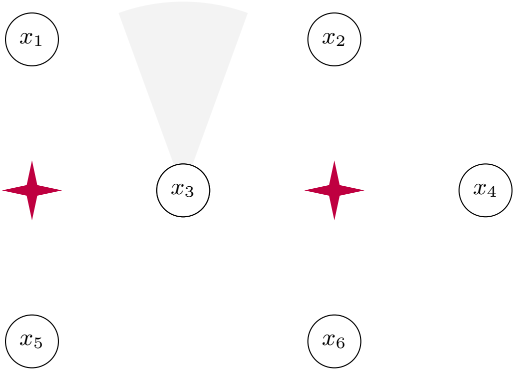
Motivating example

Sensor networks



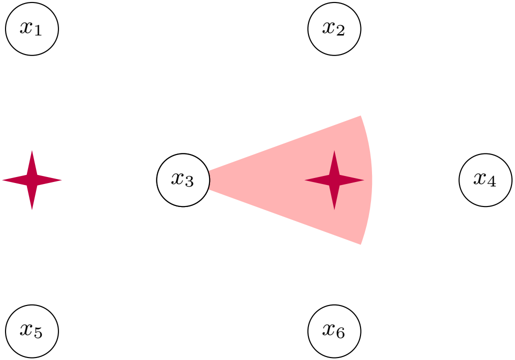
Motivating example

Sensor networks



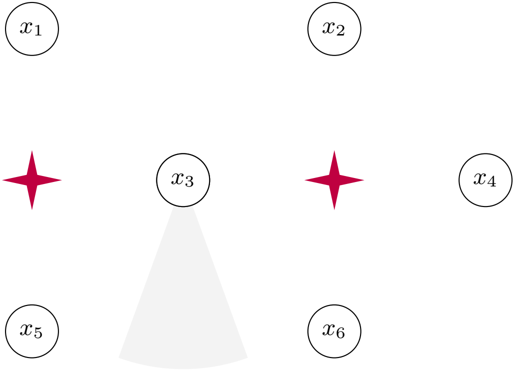
Motivating example

Sensor networks



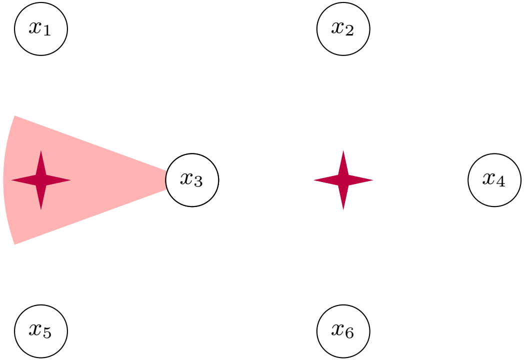
Motivating example

Sensor networks



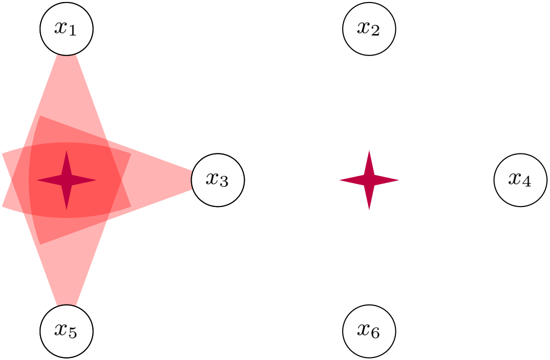
Motivating example

Sensor networks



Motivating example

Sensor networks



x_1	x_3	x_5	Sat?
N	N	N	✗
N	N	E	✗
…			✗
S	W	N	✓
…			✗
W	W	W	✗

Model the problem as a CSP!

CSP

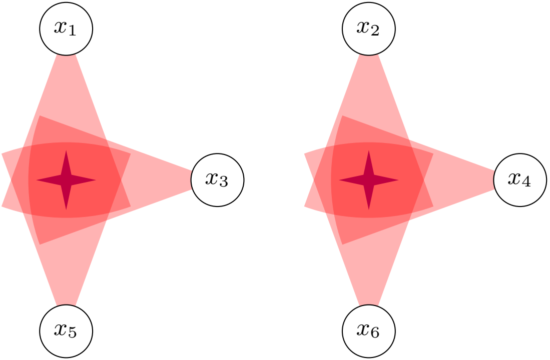
Constraint Satisfaction

- 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 $x_{i_1}, x_{i_1}, \ldots, x_{i_n}$ it involves

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

CSP

Constraint Satisfaction

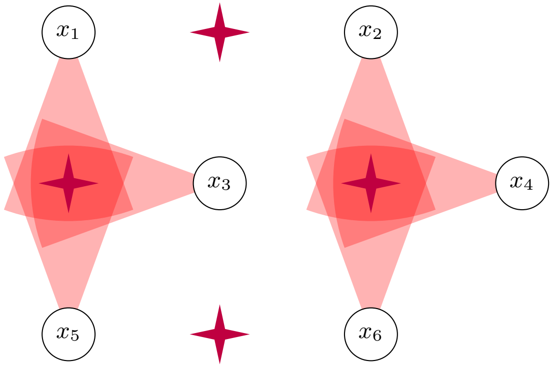


x_1	x_3	x_5	Sat?
N	N	N	✗
N	N	E	✗
	...		✗
S	W	N	✓
	...		✗
W	W	W	✗

Model the problem
as a CSP!

Max-CSP

Max Constraint Satisfaction



x_1	x_3	x_5	Sat?
N	N	N	✗
N	N	E	✗
	...		✗
S	W	N	✓
	...		✗
W	W	W	✗

Model the problem
as a Max-CSP!

Max-CSP

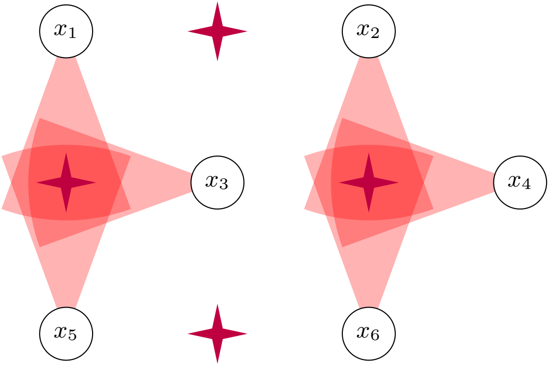
Max Constraint Satisfaction

- 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 $x_{i_1}, x_{i_1}, \ldots, x_{i_n}$ it involves

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

Max-CSP

Max Constraint Satisfaction

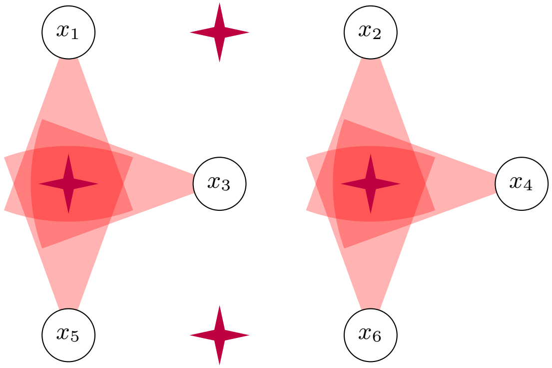


x_1	x_3	x_5	Sat?
N	N	N	✗
N	N	E	✗
	…		✗
S	W	N	✓
	…		✗
W	W	W	✗

Model the problem as a Max-CSP!

WCSP (or COP)

Constraint Optimization



x_1	x_3	x_5	Cost
N	N	N	∞
N	N	E	∞
	...		∞
S	W	N	10
	...		∞
W	W	W	∞

Model the problem as a COP!

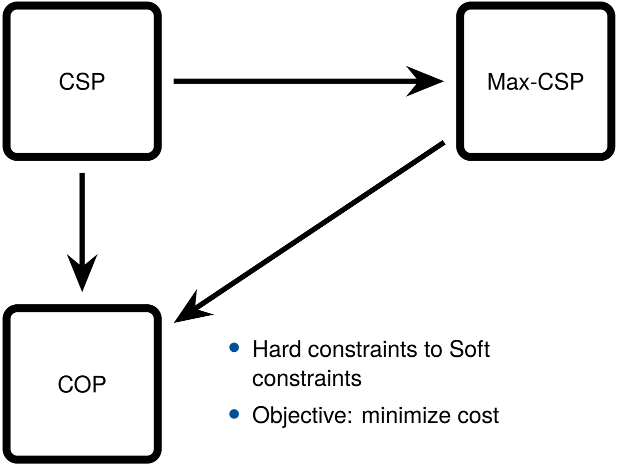
WCSP (or COP)

Constraint Optimization

- Variables $X = \{x_1, \dots, x_n\}$
- Domains $D = \{D_1, \dots, D_n\}$
- Constraints $C = \{c_1, \dots, c_m\}$
  where a constraint $c_i : D_{i_1} \times D_{i_2} \times \dots \times D_{i_n} \to \mathbb{R}_+ \cup \{\infty\}$ expresses the degree of constraint violation

- **Goal**: Find an assignment to all variables that minimizes the sum of all the constraints

Constraint Reasoning

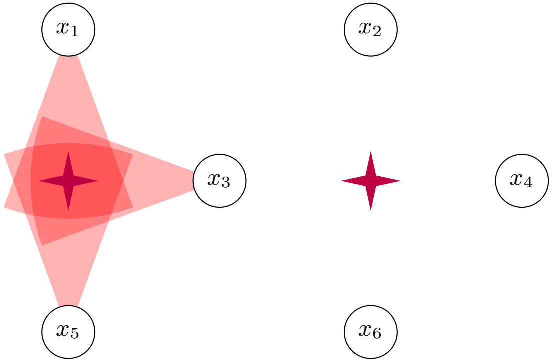


- Objective: maximize #constraints satisfied

- Hard constraints to Soft constraints
- Objective: minimize cost

WCSP (or COP)

Constraint Optimization

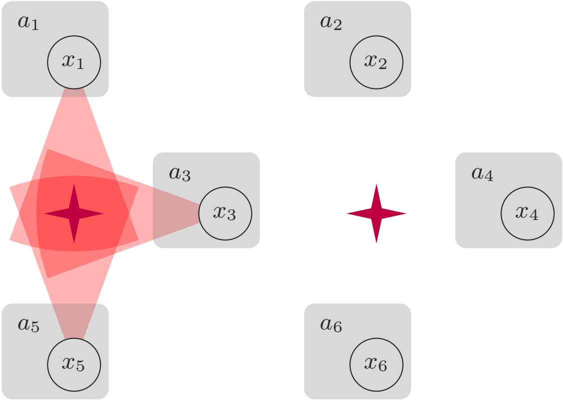


Imagine that each sensor is an autonomous agent

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

DCOP

Distributed Constraint Optimization [Modi et al., 2005]

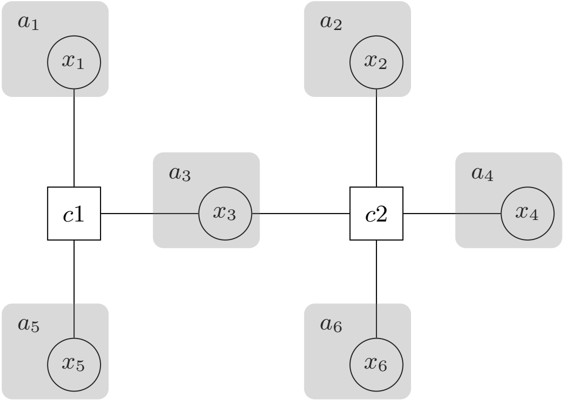


Imagine that each sensor is an autonomous agent

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

DCOP

Distributed Constraint Optimization [Modi et al., 2005]

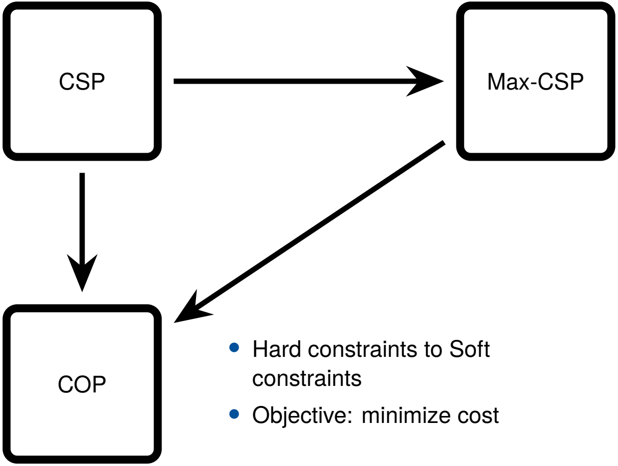


DCOP

Distributed Constraint Optimization [Modi et al., 2005]

- Agents $X = \{a_1, \ldots, a_l\}$
- 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 to all variables that minimizes the sum of all the constraints

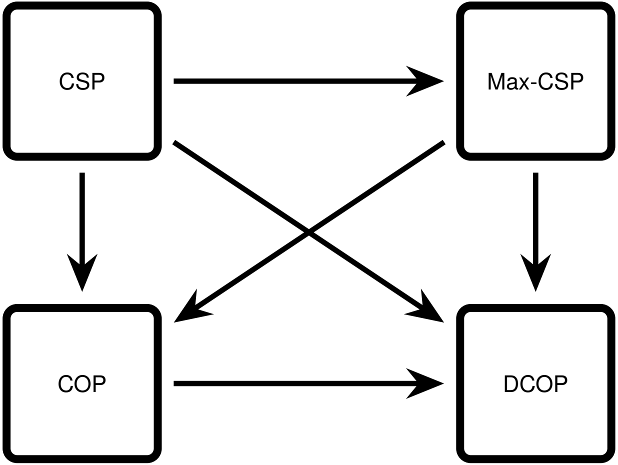


- Objective: maximize #constraints satisfied

- Hard constraints to Soft constraints
- Objective: minimize cost

DCOP

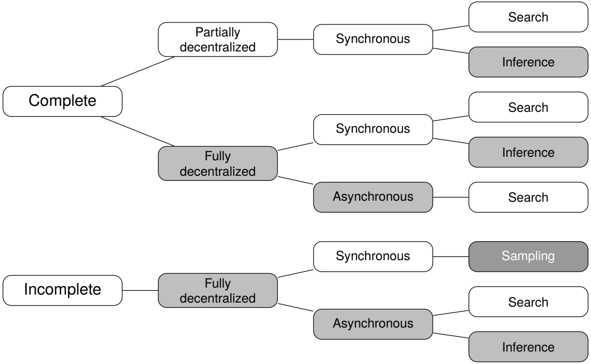
Distributed Constraint Optimization [Modi et al., 2005]



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

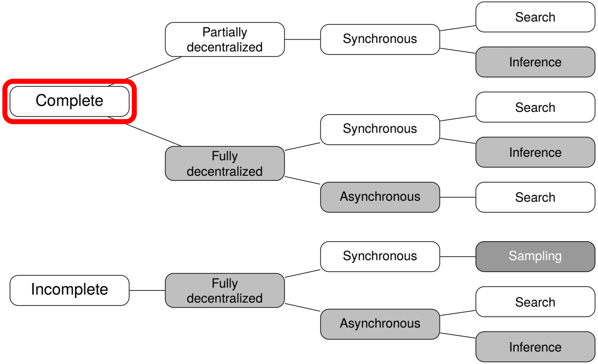
DCOP Algorithms

See [Fioretto et al., 2018]



DCOP Algorithms

See [Fioretto et al., 2018]

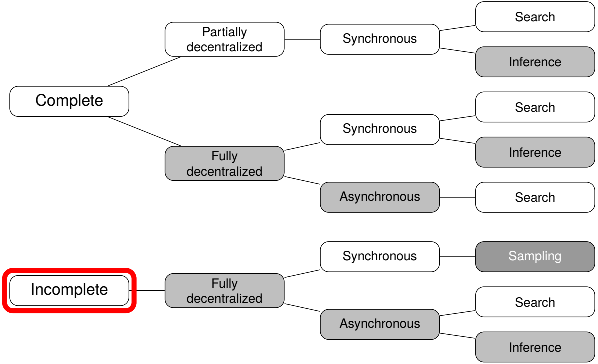


Important metrics

- Agent complexity
- Network loads
- Message size

DCOP Algorithms

See [Fioretto et al., 2018]



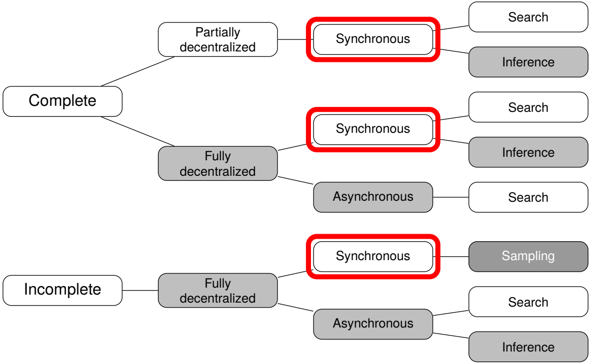
Important metrics

- Agent complexity
- Network loads
- Message size

- Anytime
- Quality guarantees
- Execution time vs. solution quality

DCOP Algorithms

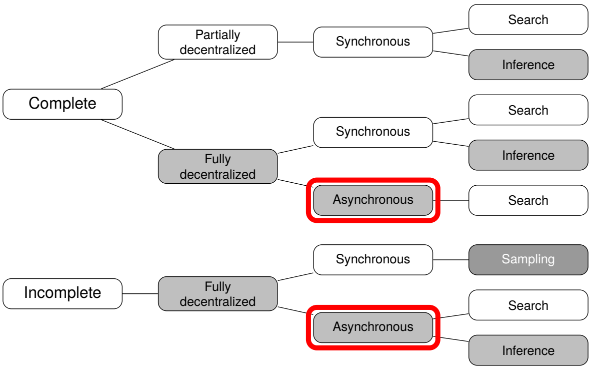
See [Fioretto et al., 2018]



- 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

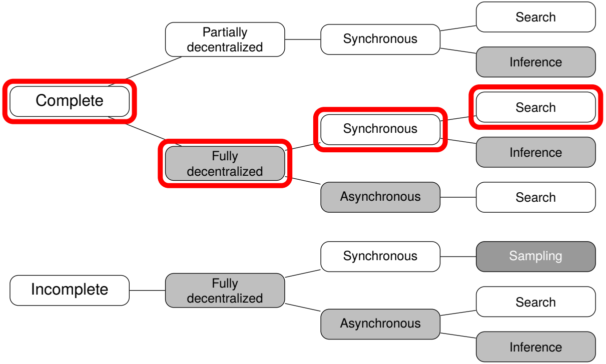
See [Fioretto et al., 2018]



- 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

See [Fioretto et al., 2018]

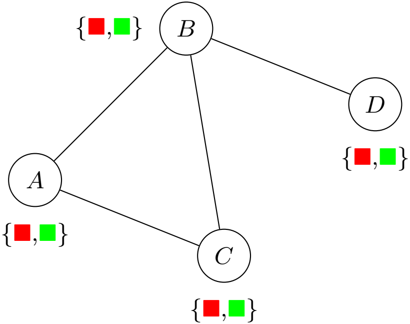


Synchronous Branch-and-Bound (SBB)

[Hirayama and Yokoo, 1997]

Synchronous Branch-and-Bound (SBB)

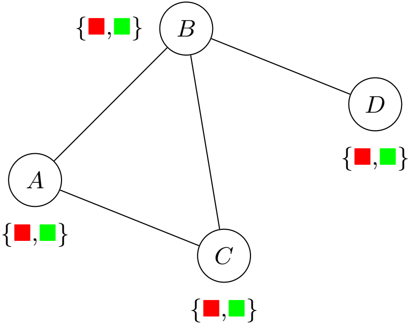
[HIRAYAMA and YOKOO, 1997]



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

Synchronous Branch-and-Bound (SBB)

[Hirayama and Yokoo, 1997]



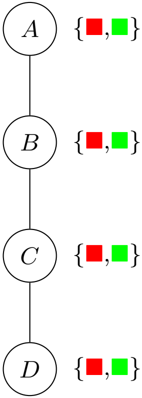
x_i	x_j	(A, B)	(A, C)	(B, C)	(B, C)
🟥	🟥	5	5	5	3
🟥	🟩	8	10	4	8
🟩	🟥	20	20	3	10
🟩	🟩	3	3	3	3

How do we solve this distributedly?

Synchronous Branch-and-Bound (SBB)

[HIRAYAMA and YOKOO, 1997]

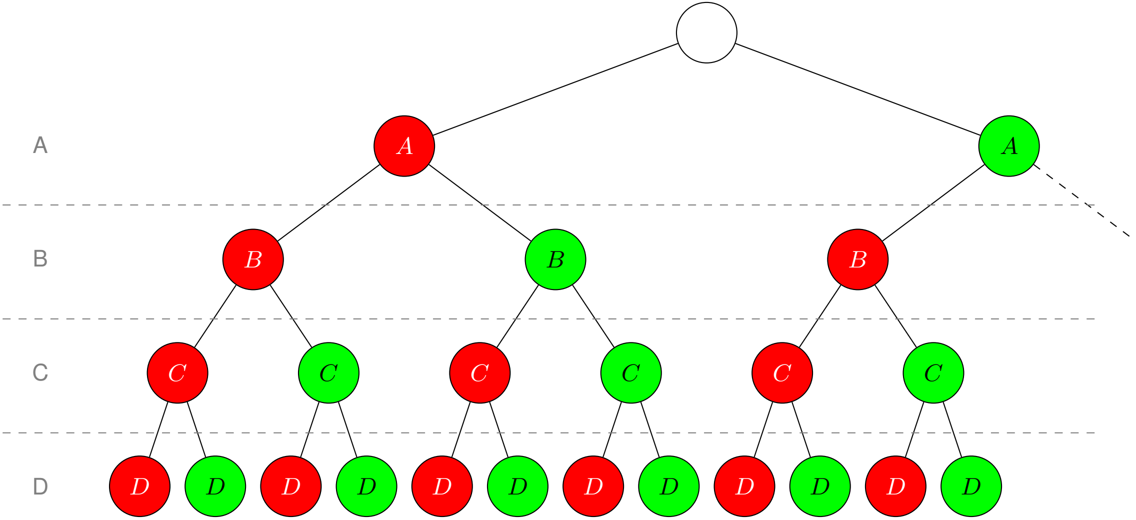
Complete ordering



- 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

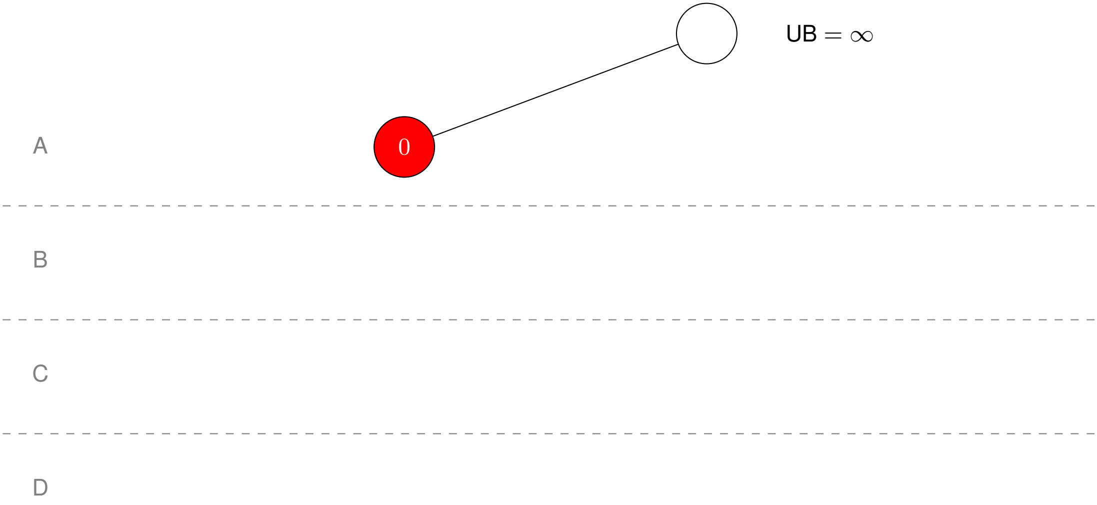
Synchronous Branch-and-Bound (SBB)

[Hirayama and Yokoo, 1997]



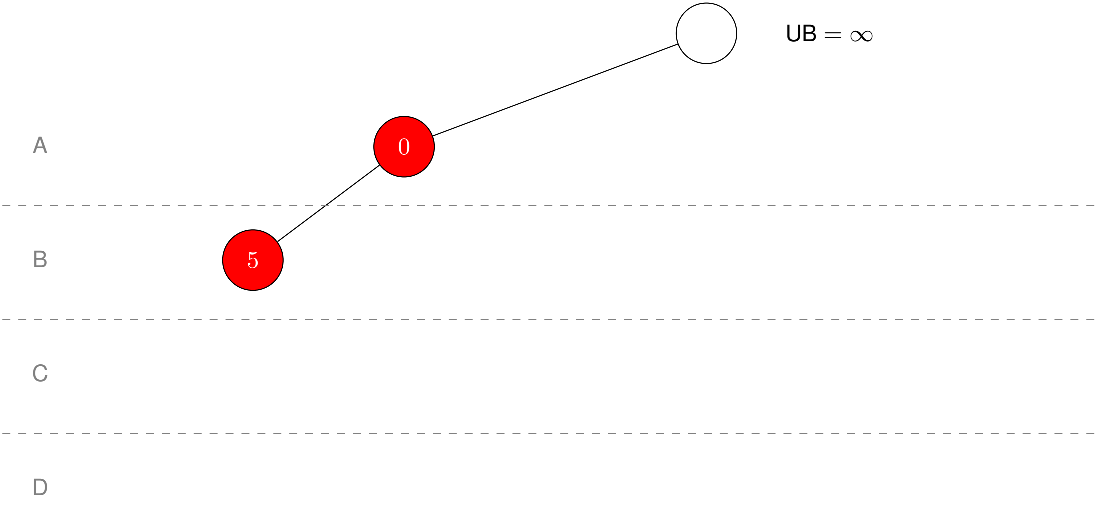
Synchronous Branch-and-Bound (SBB)

[Hirayama and Yokoo, 1997]



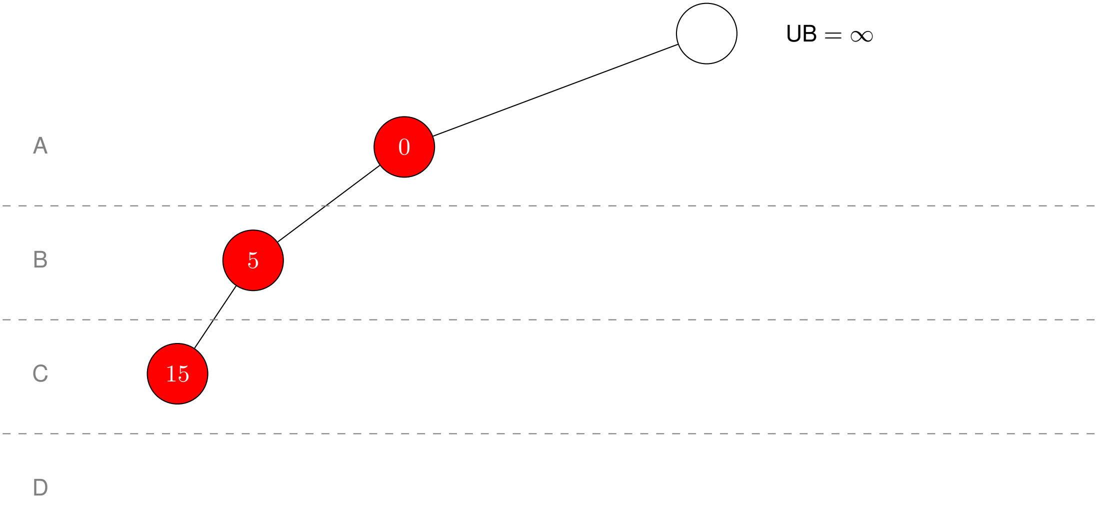
Synchronous Branch-and-Bound (SBB)

[Hirayama and Yokoo, 1997]



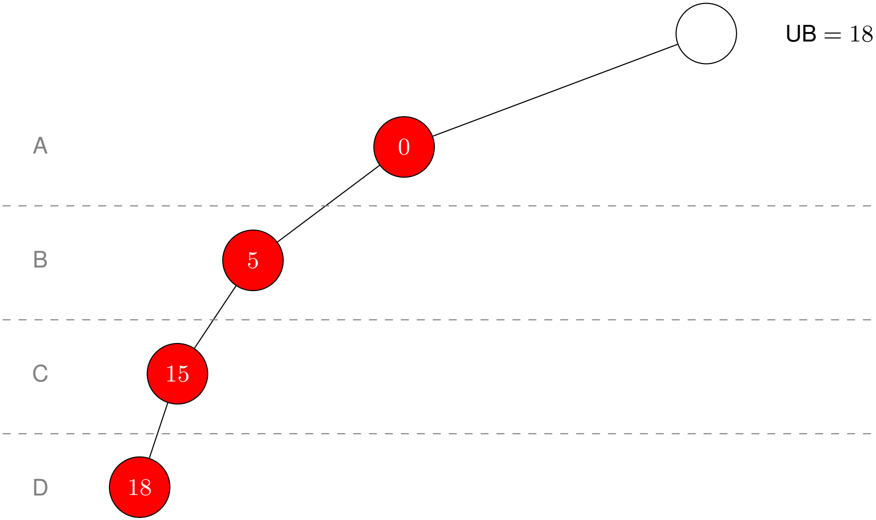
Synchronous Branch-and-Bound (SBB)

[Hirayama and Yokoo, 1997]



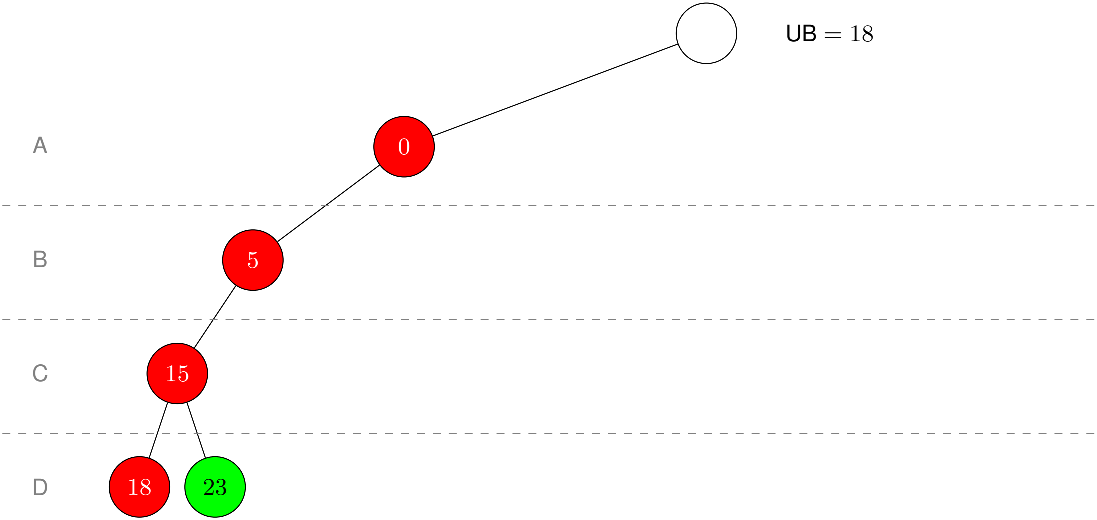
Synchronous Branch-and-Bound (SBB)

[Hirayama and Yokoo, 1997]



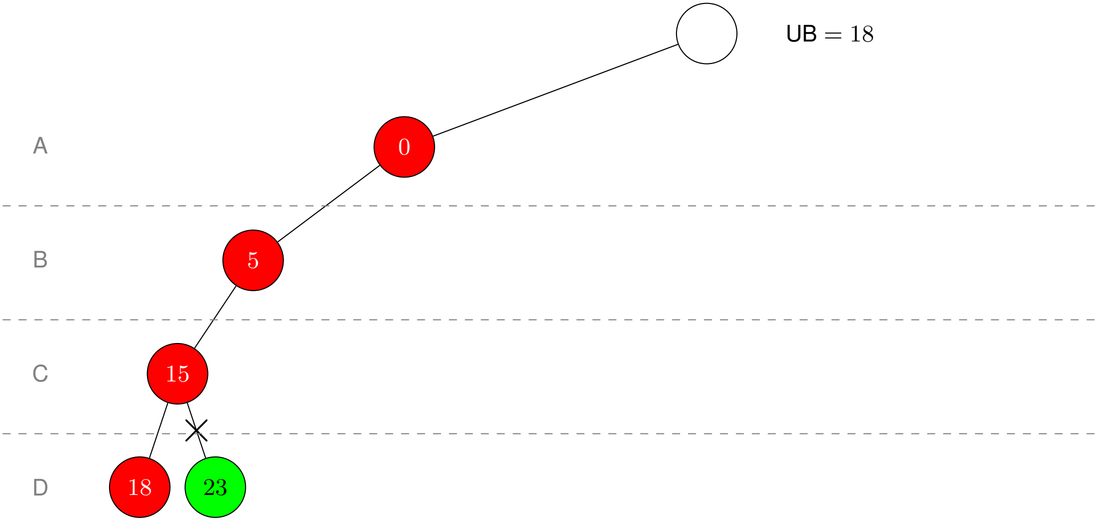
Synchronous Branch-and-Bound (SBB)

[Hirayama and Yokoo, 1997]



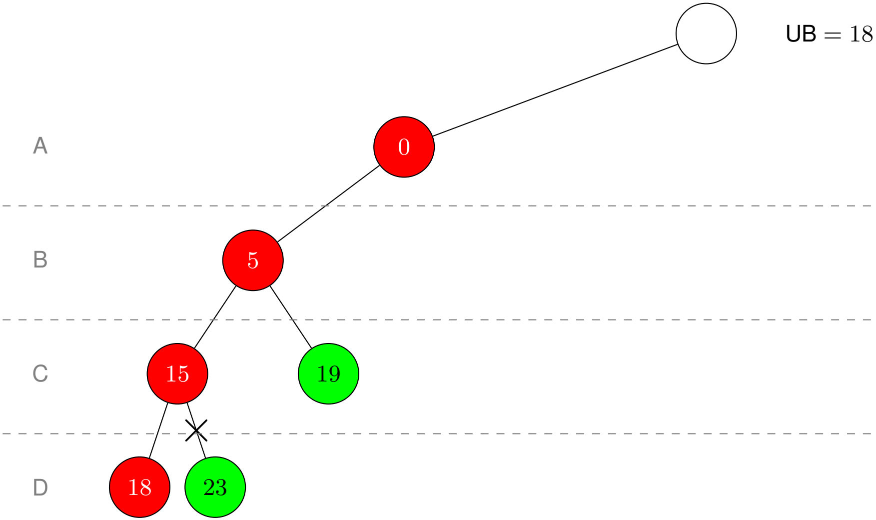
Synchronous Branch-and-Bound (SBB)

[Hirayama and Yokoo, 1997]



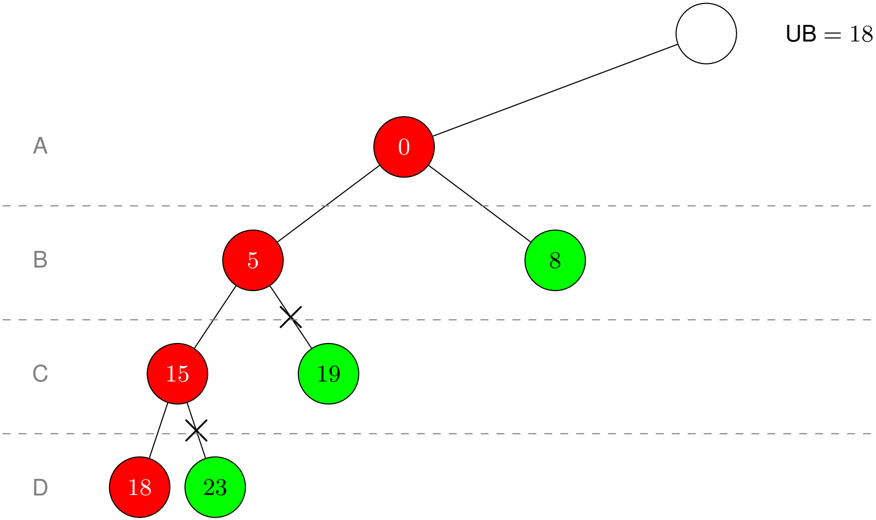
Synchronous Branch-and-Bound (SBB)

[Hirayama and Yokoo, 1997]



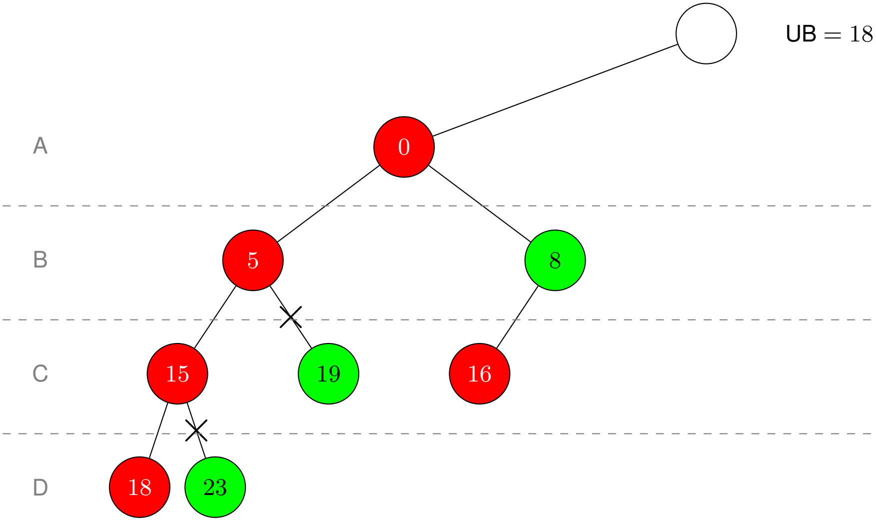
Synchronous Branch-and-Bound (SBB)

[Hirayama and Yokoo, 1997]



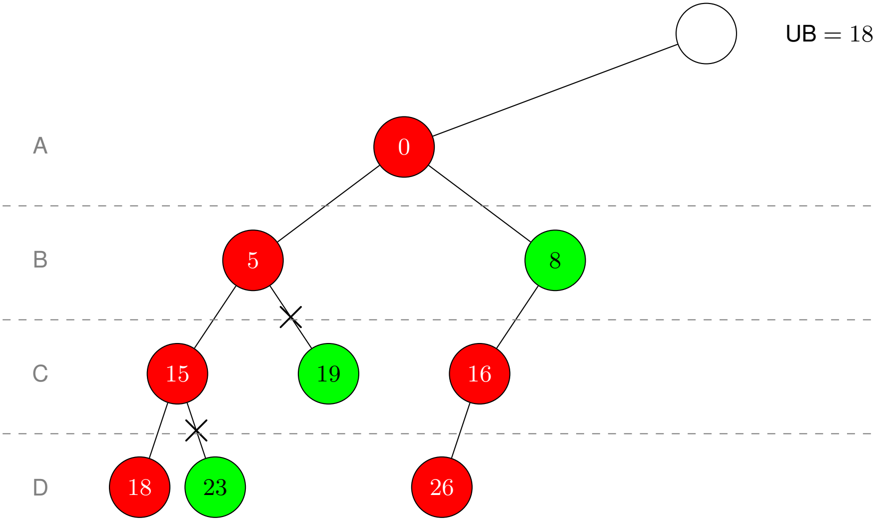
Synchronous Branch-and-Bound (SBB)

[HIRAYAMA and YOKOO, 1997]



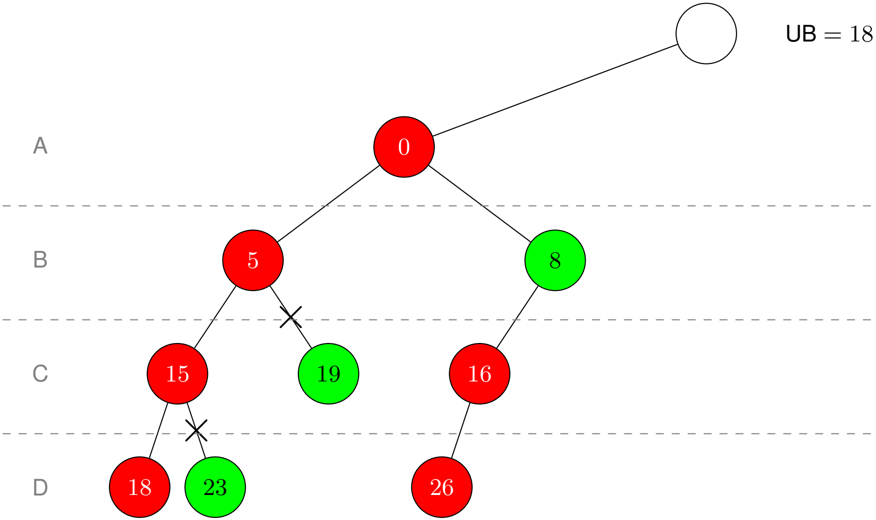
Synchronous Branch-and-Bound (SBB)

[Hirayama and Yokoo, 1997]



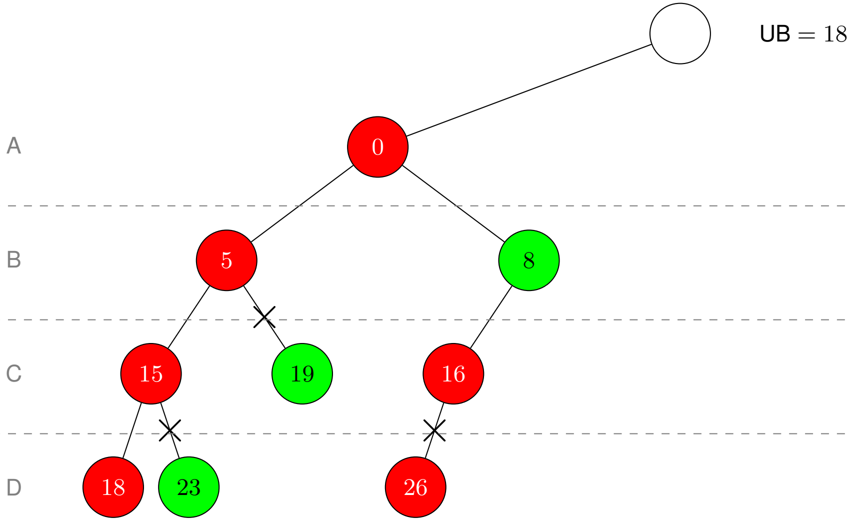
Synchronous Branch-and-Bound (SBB)

[Hirayama and Yokoo, 1997]



Synchronous Branch-and-Bound (SBB)

[Hirayama and Yokoo, 1997]



Synchronous Branch-and-Bound (SBB)

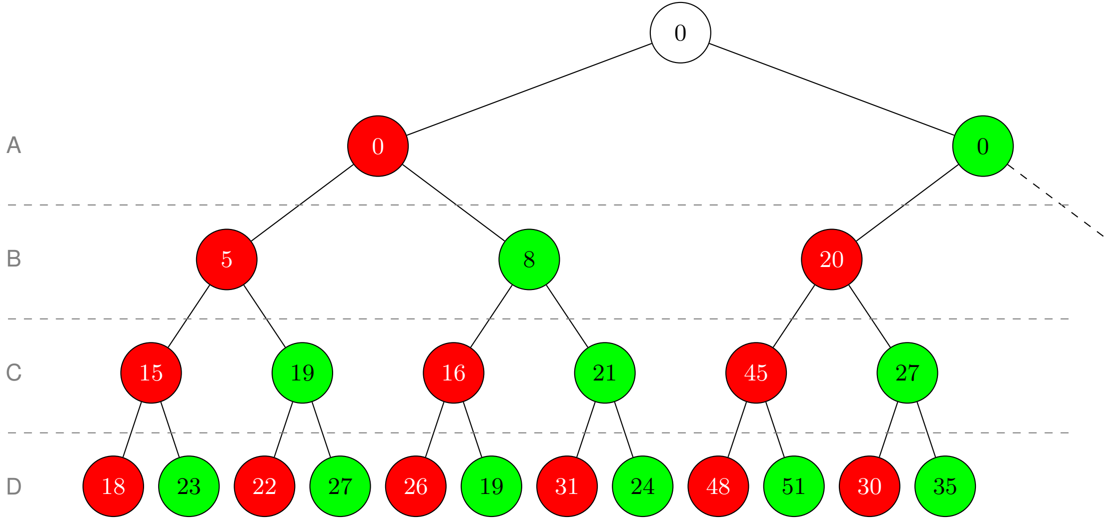
[Hirayama and Yokoo, 1997]

	SBB
Correct the solution it finds is optimal	Yes
Complete it terminates	Yes
Message complexity max size of messages	$\mathcal{O}(d)$
Network load max number of messages	$\mathcal{O}(b^d)$
Runtime how long it takes	$\mathcal{O}(b^d)$

branching factor = b
num variables = d

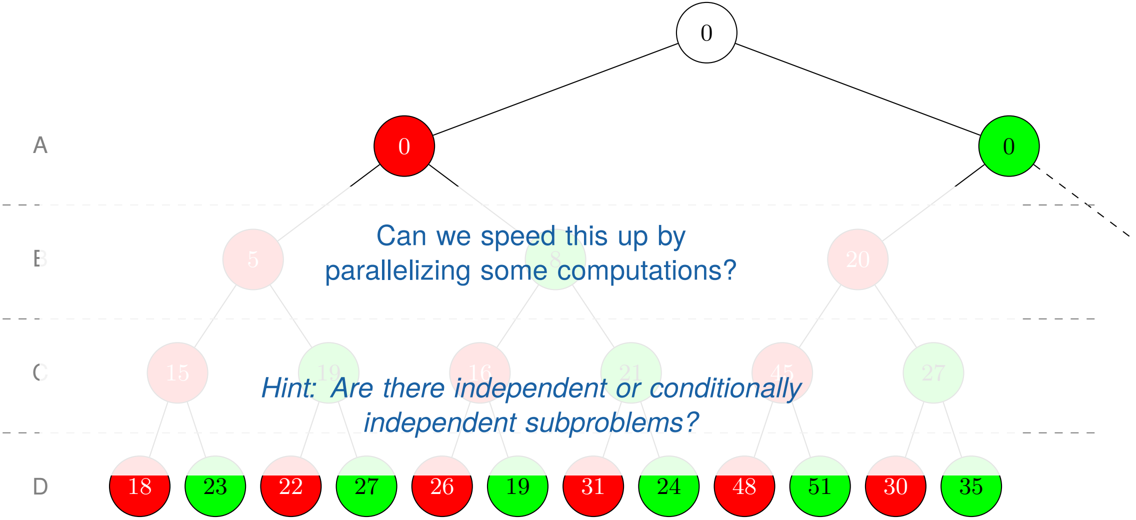
Synchronous Branch-and-Bound (SBB)

[Hirayama and Yokoo, 1997]



Synchronous Branch-and-Bound (SBB)

[Hirayama and Yokoo, 1997]

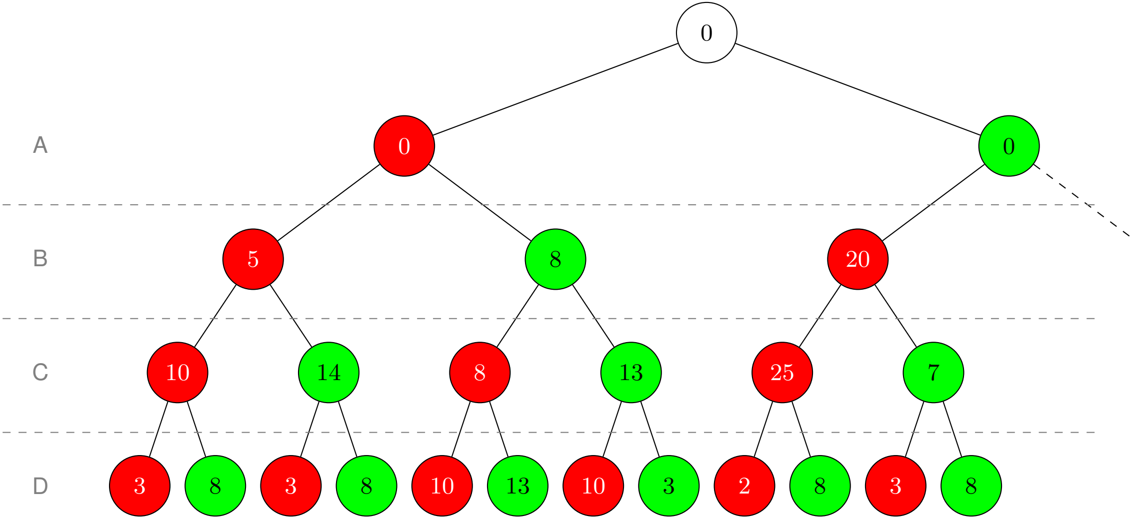


Can we speed this up by parallelizing some computations?

Hint: Are there independent or conditionally independent subproblems?

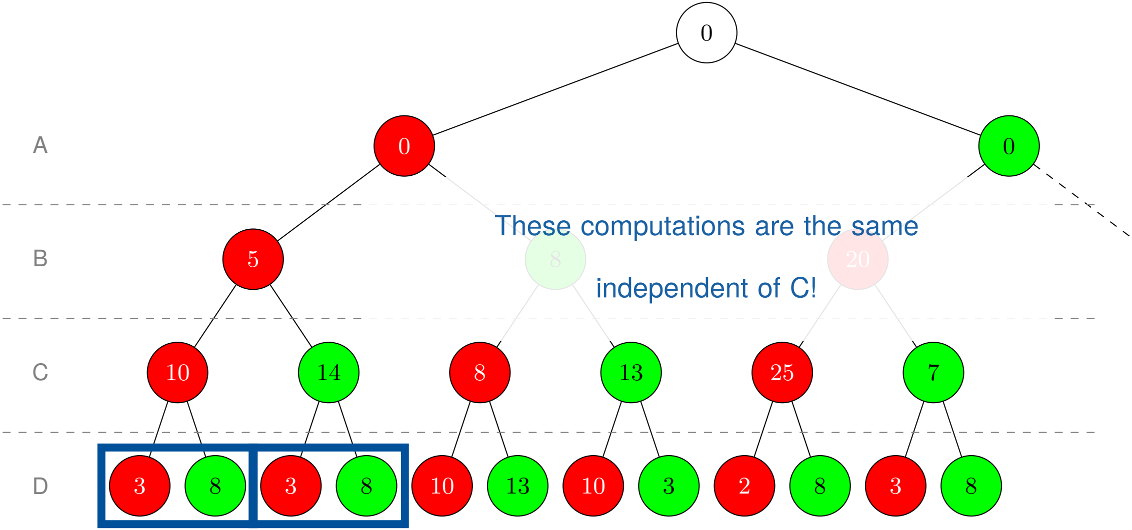
Synchronous Branch-and-Bound (SBB)

[Hirayama and Yokoo, 1997]

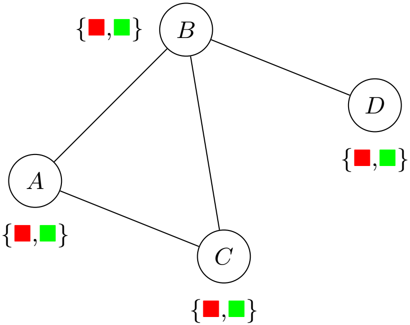


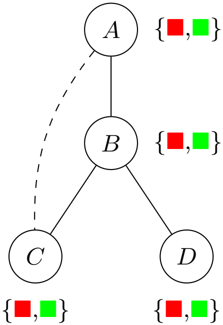
Synchronous Branch-and-Bound (SBB)

[Hirayama and Yokoo, 1997]

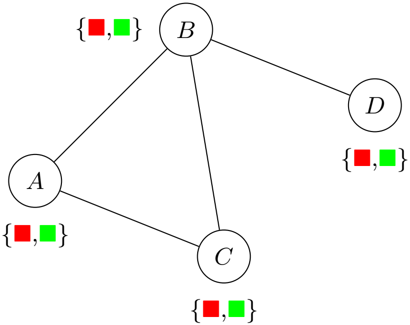


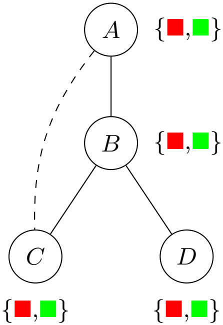
Pseudo-Tree





Pseudo-Tree



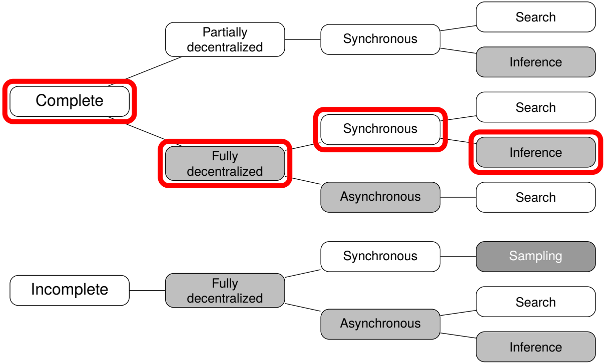


Definition (Pseudo-Tree)

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

DCOP Algorithms

See [Fioretto et al., 2018]



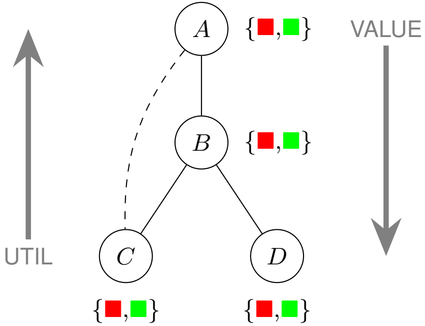
Distributed Pseudotree Optimization Procedure (DPOP)

[Adrian Petcu and Boi Faltings, 2005]

DPOP

[Adrian Petcu and Boi Faltings, 2005]

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



DPOP

[Adrian Petcu and Boi Faltings, 2005]

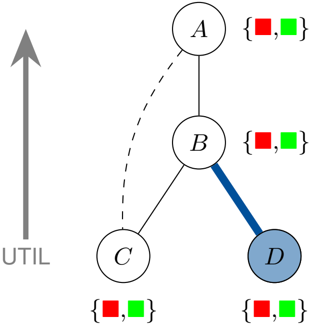
B	D	(B, D)
r	r	3
r	g	8
g	r	10
g	g	3

min$\{3, 8\} = 3$

min$\{10, 3\} = 3$

Message to B

B	cost
r	3
g	3



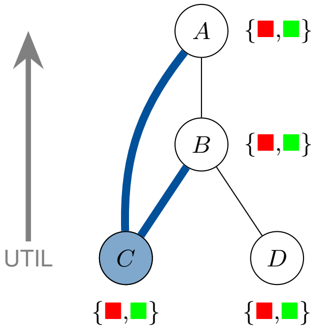
DPOP

[Adrian Petcu and Boi Faltings, 2005]

A	B	C	(B, C)	(A, C)	cost
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

Message to B

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



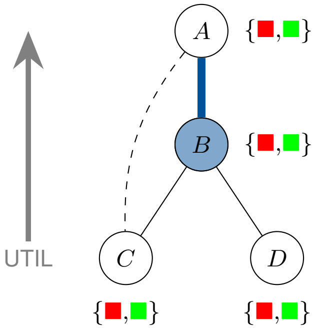
DPOP

[Adrian Petcu and Boi Faltings, 2005]

A	B	(A, B)	Util C	Util D	cost
r	r	5	10	53	18
r	g	8	8	3	19
g	r	20	7	3	30
g	g	3	6	3	12

Message to A

A	cost
r	18
g	12

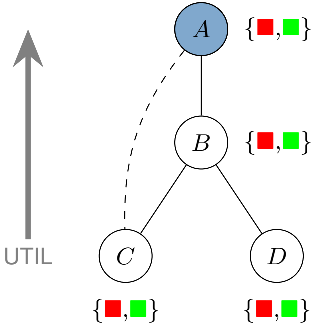


DPOP

[Adrian Petcu and Boi Faltings, 2005]

A	cost
r	18
g	12

optimal cost $= 12$

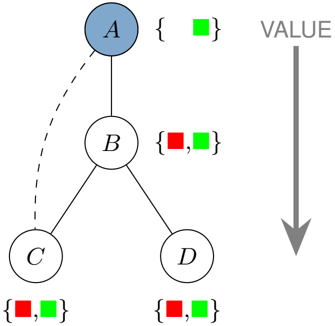


DPOP

[Adrian Petcu and Boi Faltings, 2005]

A	cost
r	18
g	12

- Select value for $A = g$
- Send MSG "$A = g$" to agents B and C

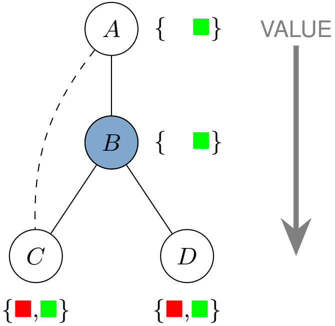


DPOP

[Adrian Petcu and Boi Faltings, 2005]

A	B	(A,B)	Util C	Util D	cost
r	r	5	10	53	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

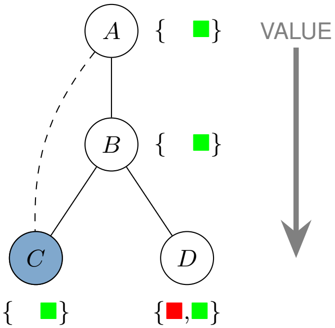


DPOP

[Adrian Petcu and Boi Faltings, 2005]

A	B	C	(B,C)	(A,C)	cost
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 $C = g$



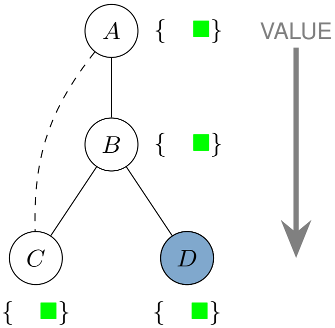
DPOP

[Adrian Petcu and Boi Faltings, 2005]

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

min$\{3,8\} = 3$

min$\{10,3\} = 3$



- Select value for $D = g$

DPOP

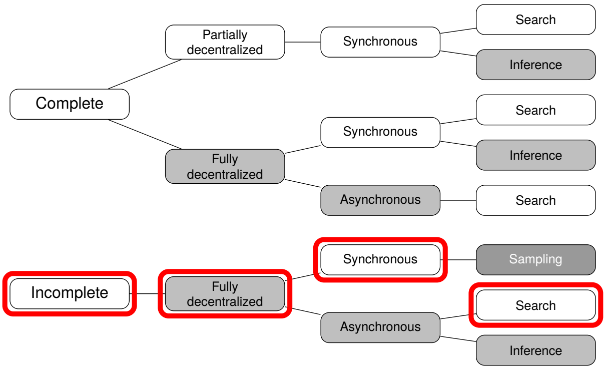
[Adrian Petcu and Boi Faltings, 2005]

	SBB	DPOP
Correct the solution it finds is optimal	Yes	Yes
Complete it terminates	Yes	Yes
Message complexity max size of messages	$\mathcal{O}(d)$	$\mathcal{O}(b^d)$
Network load max number of messages	$\mathcal{O}(b^d)$	$\mathcal{O}(d)$
Runtime how long it takes	$\mathcal{O}(b^d)$	$\mathcal{O}(b^d)$

branching factor = b
num variables = d

DCOP Algorithms

See [Fioretto et al., 2018]



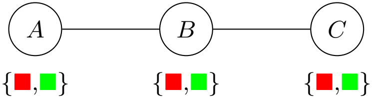
Distributed Local Search

[Maheswaran et al., 2004; Weixiong Zhang et al., 2003]

Local Search Algorithms

- DSA: Distributed Stochastic Search [W. Zhang et al., 2005]
- MGM: Maximum Gain Messages Algorithm [Maheswaran et al., 2004]
- Note: we now maximize utilities
- Every agent individually decides whether to change its value or not
- Decision involves
  - knowing neighbors' values
  - calculation of utility gain by changing values
  - probabilities

x_i	x_j	(A, B)	(B, C)
🟥	🟥	5	5
🟥	🟩	5	0
🟩	🟥	0	0
🟩	🟩	8	8



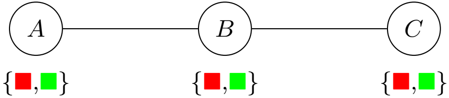
DSA Algorithm

[W. Zhang et al., 2005]

- 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

DSA Algorithm

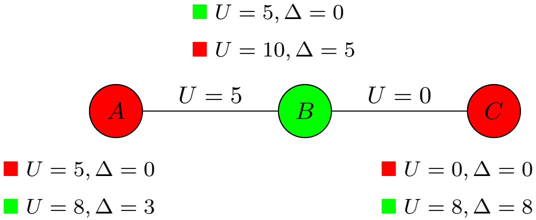
[W. Zhang et al., 2005]



x_i	x_j	(A, B)	(B, C)
🟥	🟥	5	5
🟥	🟩	5	0
🟩	🟥	0	0
🟩	🟩	8	8

DSA Algorithm

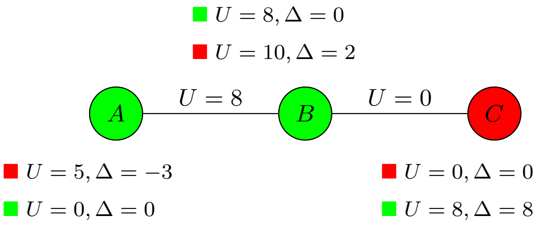
[W. Zhang et al., 2005]



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

DSA Algorithm

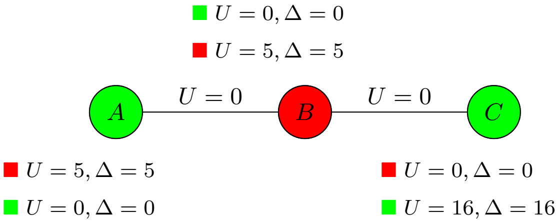
[W. ZHANG et al., 2005]



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

DSA Algorithm

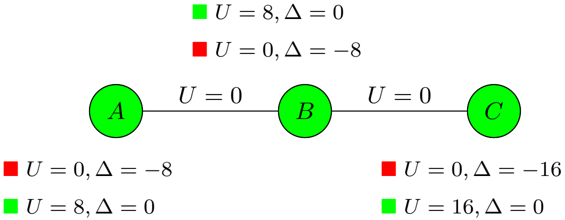
[W. ZHANG et al., 2005]



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

DSA Algorithm

[W. Zhang et al., 2005]



x_i	x_j	(A, B)	(B, C)
🟥	🟥	5	5
🟥	🟩	5	0
🟩	🟥	0	0
🟩	🟩	8	8

MGM Algorithm

[Maheswaran et al., 2004]

- 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

MGM Algorithm

[Maheswaran et al., 2004]

- 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

Large Great if you need an anytime algorithm!

MGM vs DSA

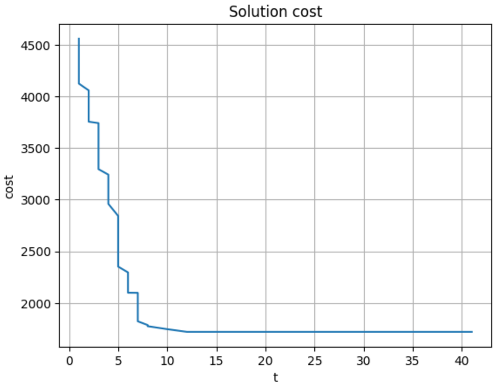


Figure: MGM

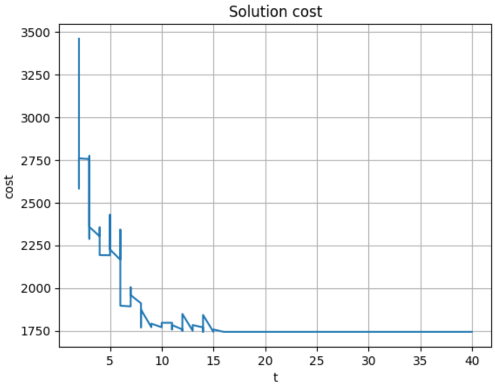


Figure: DSA

Extensions to the DCOP Framework

- Dynamic DCOPs
  - SDPOP [A. Petcu and B. Faltings, 2005], I-ADOPT and I-BnB-ADOPT [Yeoh et al., 2011], FMS [Ramchurn et al., 2010]
- Multi-Objective DCOPs
  - MO-SBB [Medi et al., 2014], Pseudo-tree Based Algorithm [Matsui et al., 2012], B-MOMS [Delle Fave et al., 2011], DP-AOF [Okimoto et al., 2013]
- Asymetric DCOPs
  - SyncABB-2ph, SyncABB-1ph, ACLS, MCS-MGM [Grinshpoun et al., 2013]
- Probabilistic DCOPs
  - $\mathbb{E}$[DPOP] and SD-DPOP [Léauté and B. Faltings, 2011; Nguyen et al., 2012], U-GDL [Stranders et al., 2011]
- Continuous DCOPs
  - CMS [Stranders et al., 2009], HCMS [Voice et al., 2010], PFD [Choudhury et al., 2020], EC-DPOP, AC-DPOP, CAC-DPOP, C-DSA [Hoang et al., 2020], C-CoCoA [Sarker et al., 2021]
- ...

Today's Menu

Coordinating using Auctions

What are Auctions?

- Competitive bidding processes for **allocating** goods or services
- Buyers submit **bids**, highest bid wins
- Different auction schemes exist (e.g., English, Dutch, sealed-bid)

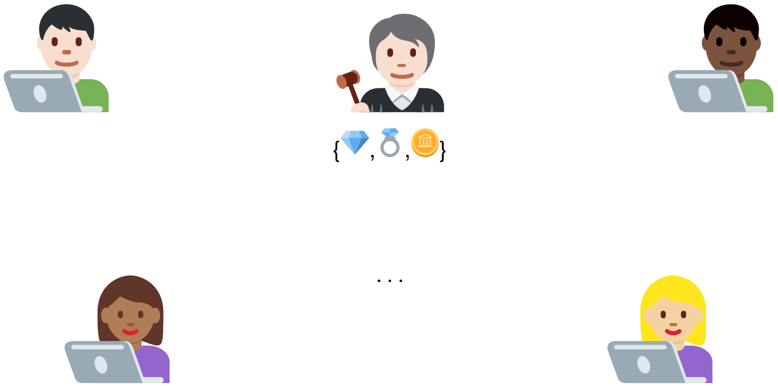
Coordinating using Auctions

What are Auctions?

- Competitive bidding processes for **allocating** goods or services
- Buyers submit **bids**, highest bid wins
- Different auction schemes exist (e.g., English, Dutch, sealed-bid)
- Single item vs. Multiple items

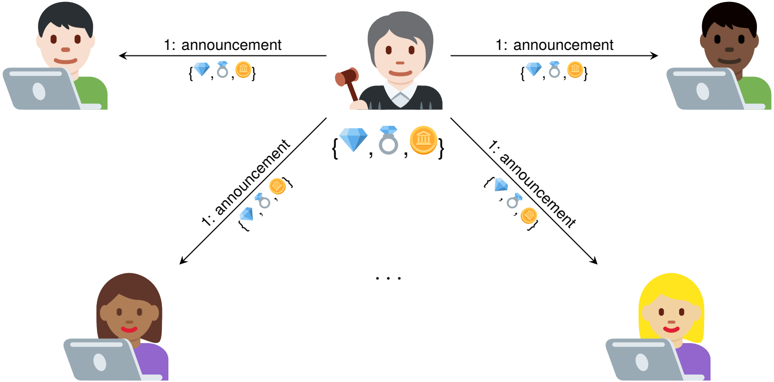
Coordinating using Auctions

Classical Protocol



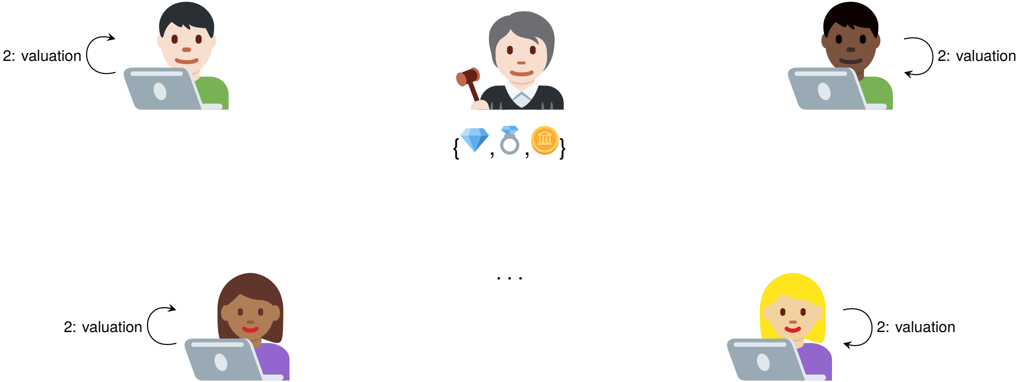
Coordinating using Auctions

Classical Protocol



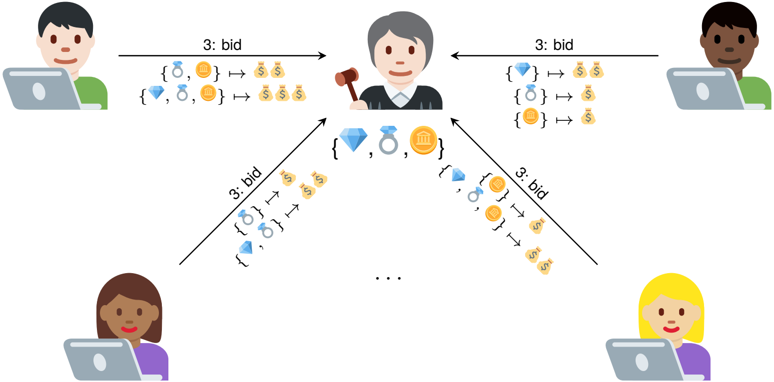
Coordinating using Auctions

Classical Protocol



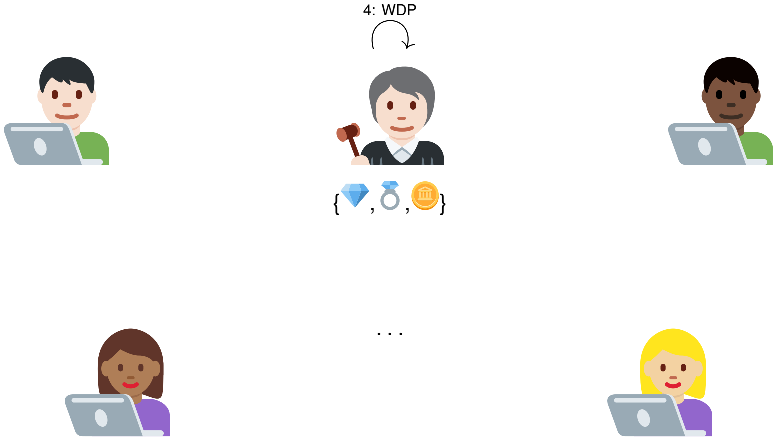
Coordinating using Auctions

Classical Protocol



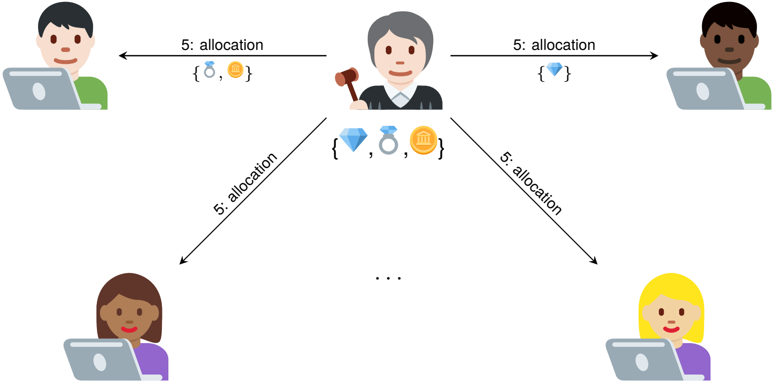
Coordinating using Auctions

Classical Protocol



Coordinating using Auctions

Classical Protocol



Coordinating using Auctions

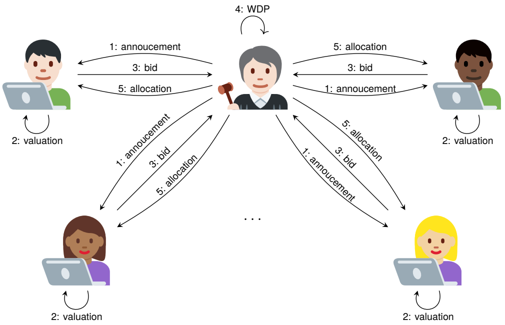
Simple Formulation of Winner Determination Problem (WDP)

- $\mathcal{T} = \{t_1, t_2, ..., t_m\}$ the set of goods to be auctioned
- $\mathcal{A} = \{a_1, a_2, ..., a_n\}$ the set of bidders
- $\mathcal{B} = \{b_1, b_2, ..., b_k\}$ the set of bid combinations (bundles)
- $y_{ik} \in \{0, 1\}$ indicates whether bundle b_k is allocated to bidder a_i
- c_{ik} the price offered by bidder a_i for bundle s_k

$$
\begin{aligned}
\max \quad & \sum_{a_i \in \mathcal{A}} \sum_{b_k \in \mathcal{S}} c_{ik} y_{ik} \\
\text{s.t.} \quad & \sum_{a_i \in \mathcal{A}} \sum_{b_k \subseteq \mathcal{T}, t_j \in b_k} y_{ik} \leq 1, && \forall t_j \in \mathcal{T} \\
& \sum_{b_k \subseteq \mathcal{T}} y_{ik} \leq 1, && \forall a_i \in \mathcal{A}
\end{aligned}
$$

Coordinating using Auctions

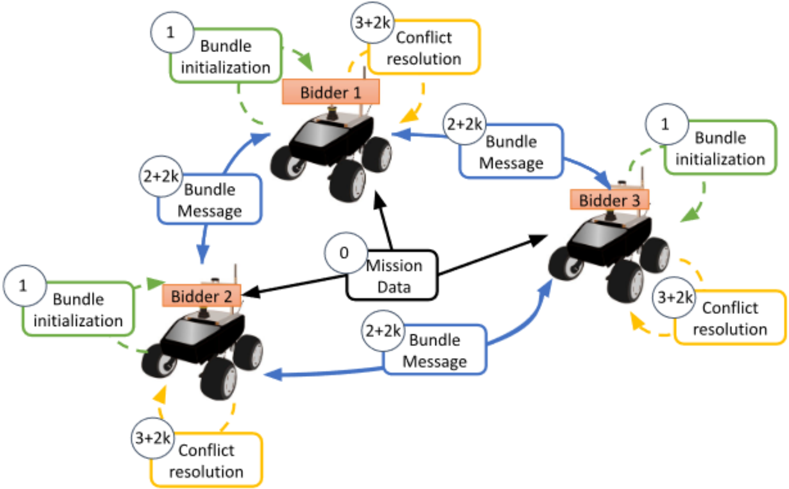
Many auction schemes [Parsons et al., 2011]



- **Combinatorial Auctions** (CA)
  [Cramton et al., 2010]
- **Parallel Single Item Auctions** (PSI)
  [Koenig et al., 2006]
  - Each agent bids on the *whole set of items* in parallel
- **Sequential Single Item Auctions** (SSI)
  [Lagoudakis et al., 2005]
  - Each agent *sequentially* bids on a *single item* wrt to the already allocated items
- **Consensus-based Bundle Auction** (CBBA)
  [Choi et al., 2009]
  - WDP decentralized as a *consensus* on bundles

Consensus-based Bundle Auction (CBBA)

[Choi et al., 2009]



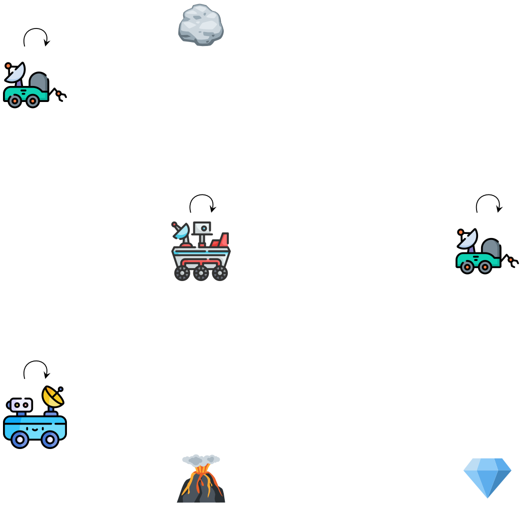
CBBA Algorithm

How does it work?

CBBA Algorithm

How does it work?

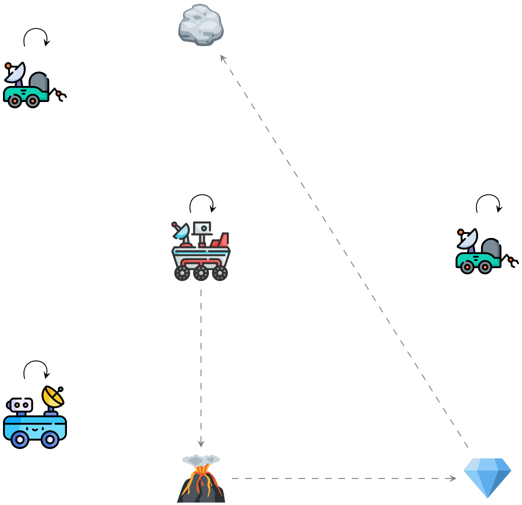
- **Bundle Construction**
  - Each agent creates bundles of tasks it can complete
  - May include dependent tasks



CBBA Algorithm

How does it work?

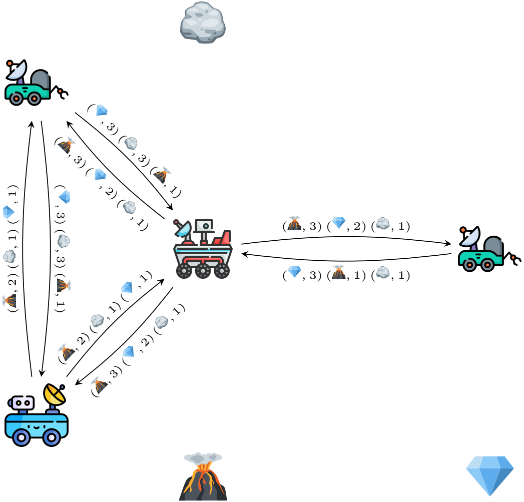
- **Bundle Construction**
  - Each agent creates bundles of tasks it can complete
  - May include dependent tasks



CBBA Algorithm

How does it work?

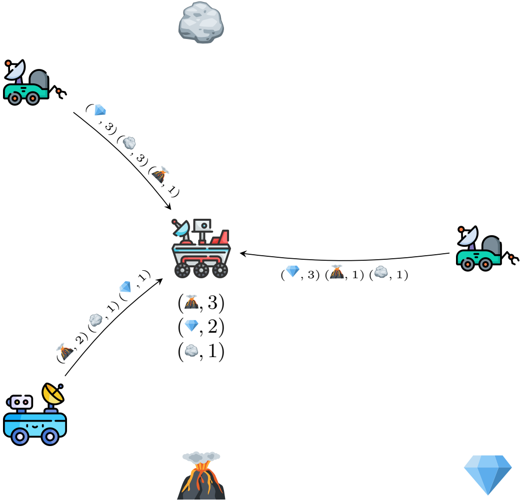
- **Bundle Construction**
  - Each agent creates bundles of tasks it can complete
  - May include dependent tasks
- **Bidding**
  - Agents bid on bundles based on their utility
  - Messages sent to neighbors
  - e.g. completion time, preferences



CBBA Algorithm

How does it work?

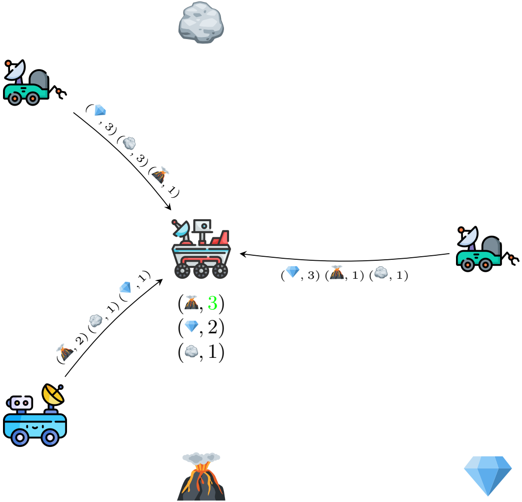
- **Bundle Construction**
  - Each agent creates bundles of tasks it can complete
  - May include dependent tasks
- **Bidding**
  - Agents bid on bundles based on their utility
  - Messages sent to neighbors
  - e.g. completion time, preferences
- **Conflict Resolution**
  - Conflicting bids are adjusted/removed
  - e.g. valuation and time stamps



CBBA Algorithm

How does it work?

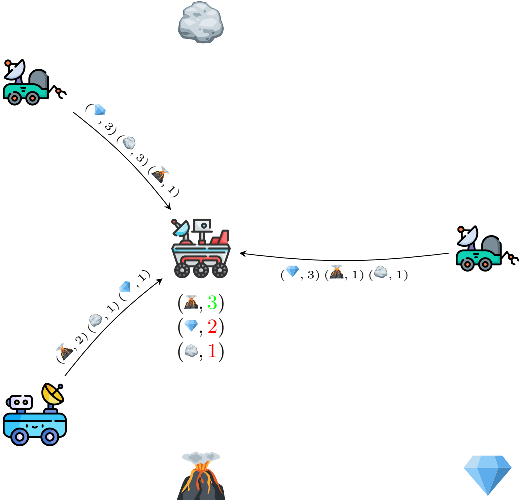
- **Bundle Construction**
  - Each agent creates bundles of tasks it can complete
  - May include dependent tasks
- **Bidding**
  - Agents bid on bundles based on their utility
  - Messages sent to neighbors
  - e.g. completion time, preferences
- **Conflict Resolution**
  - Conflicting bids are adjusted/removed
  - e.g. valuation and time stamps



CBBA Algorithm

How does it work?

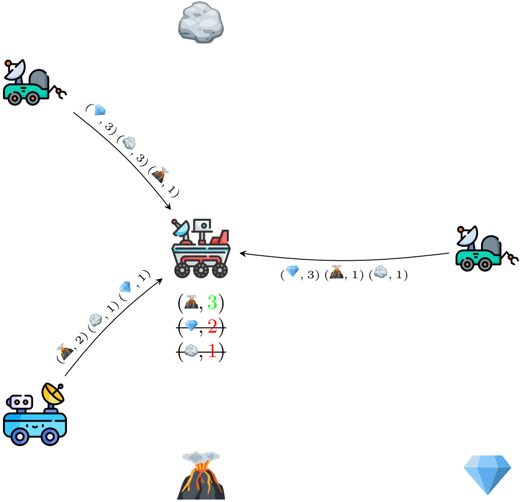
- **Bundle Construction**
  - Each agent creates bundles of tasks it can complete
  - May include dependent tasks
- **Bidding**
  - Agents bid on bundles based on their utility
  - Messages sent to neighbors
  - e.g. completion time, preferences
- **Conflict Resolution**
  - Conflicting bids are adjusted/removed
  - e.g. valuation and time stamps



CBBA Algorithm

How does it work?

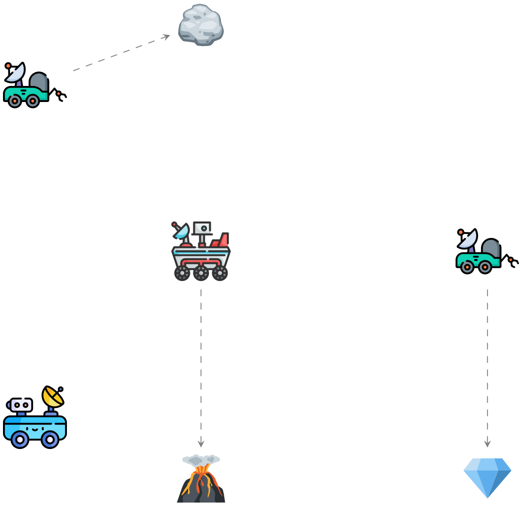
- **Bundle Construction**
  - Each agent creates bundles of tasks it can complete
  - May include dependent tasks
- **Bidding**
  - Agents bid on bundles based on their utility
  - Messages sent to neighbors
  - e.g. completion time, preferences
- **Conflict Resolution**
  - Conflicting bids are adjusted/removed
  - e.g. valuation and time stamps



CBBA Algorithm

How does it work?

- **Bundle Construction**
  - Each agent creates bundles of tasks it can complete
  - May include dependent tasks
- **Bidding**
  - Agents bid on bundles based on their utility
  - Messages sent to neighbors
  - e.g. completion time, preferences
- **Conflict Resolution**
  - Conflicting bids are adjusted/removed
  - e.g. valuation and time stamps
- **Allocation**
  - Winning bundles are allocated
  - Agents execute the tasks in their assigned bundles



Consensus-based Bundle Auction (CBBA)

Advantages of CBBA

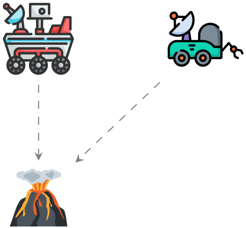
- **Decentralized:** No central authority required, enabling robust operation in dynamic environments
- **Scalable:** Efficiently handles large numbers of agents and tasks
- **Flexible:** Can be adapted to different task allocation problems and objective functions

Consensus-based Bundle Auction (CBBA)

Advantages of CBBA

- **Decentralized:** No central authority required, enabling robust operation in dynamic environments
- **Scalable:** Efficiently handles large numbers of agents and tasks
- **Flexible:** Can be adapted to different task allocation problems and objective functions

- How to handle tasks requiring multiple agents?

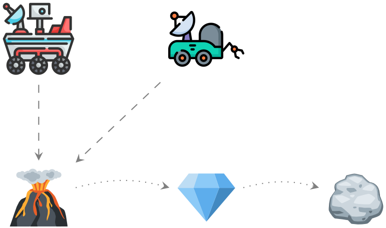


Consensus-based Bundle Auction (CBBA)

Advantages of CBBA

- **Decentralized:** No central authority required, enabling robust operation in dynamic environments
- **Scalable:** Efficiently handles large numbers of agents and tasks
- **Flexible:** Can be adapted to different task allocation problems and objective functions

- How to handle tasks requiring multiple agents?
- How to handle composite/sequenced tasks?

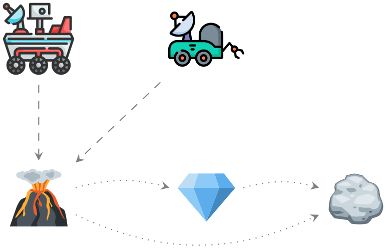


Consensus-based Bundle Auction (CBBA)

Advantages of CBBA

- **Decentralized:** No central authority required, enabling robust operation in dynamic environments
- **Scalable:** Efficiently handles large numbers of agents and tasks
- **Flexible:** Can be adapted to different task allocation problems and objective functions

- How to handle tasks requiring multiple agents?
- How to handle composite/sequenced tasks?
- How to handle alternative sequences/modes?



Today's Menu

Illustration 1: Constellation Management

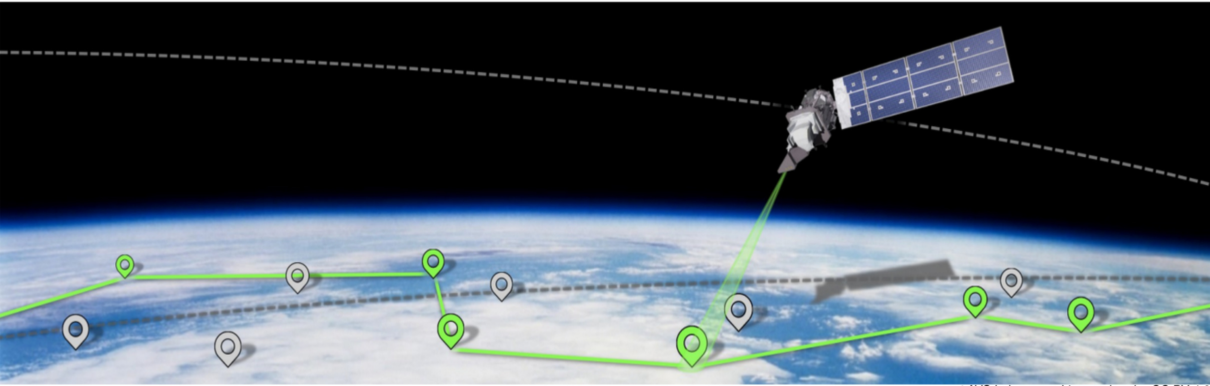
Sample system: Constellation of Agile Earth Observation Satellites (EOS)

©Airbus

- Multiple satellites, potentially operated by multiple partners
- Heterogenous orbits and sensors

Illustration 1: Constellation Management

Observing Earth using Agile Satellites

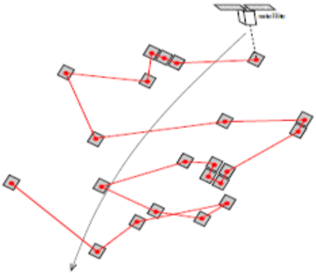


- **Agile satellites**: can image targets about-track and along-track
- Equipped with imaging instrument(s) to gather data about **ground targets**

Given a set of obervation tasks, **select** and optimally **schedule** a subset of tasks to perform under the constraints given by the **position** and the **agility** of the EOS

Illustration 1: Constellation Management

Single Satellite Problem



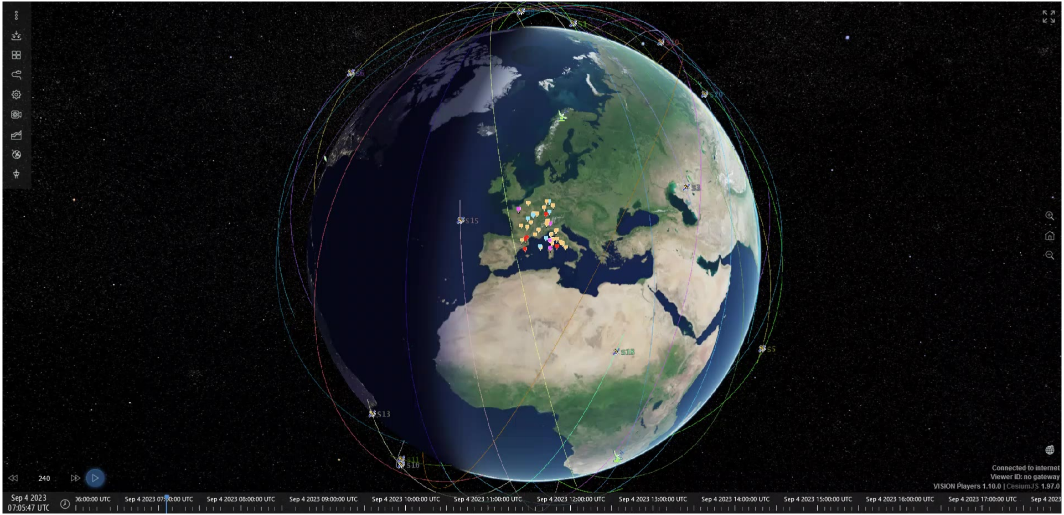
The *Earth Observation Scheduling Problem* (or EOSP) consists in finding a sequence of observations $\sigma = [\sigma_1, \dots, \sigma_K]$ such that:

- each candidate observation at most once in σ
- the successive observations can be performed during the allowed time windows; formally, the earliest start time of the first observation is $s_{\sigma_1} = S_{\sigma_1}$, the earliest start time of the kth observation is given by $s_{\sigma_k} = \max(S_{\sigma_k}, s_{\sigma_{k-1}} + tt(\sigma_{k-1}, \sigma_k, s_{\sigma_{k-1}}))$, and condition $s_{\sigma_k} \leq E_{\sigma_k}$ must be satisfied for every observation σ_k involved in σ
- the total reward collected ($\sum_{i \in \sigma} Rw_i$) is maximized

- Agile EOS scheduling problem can be mapped to TD-OP-TW [Schmid and Ehmke, 2017]
- TD-OP-TW is NP-hard [Golden et al., 1987]
- Common solution methods : ant colony optimization [Verbeeck et al., 2017], iterated local search [Garcia et al., 2010], or large neighborhood search (LNS) [Schmid and Ehmke, 2017]

Illustration 1: Constellation Management

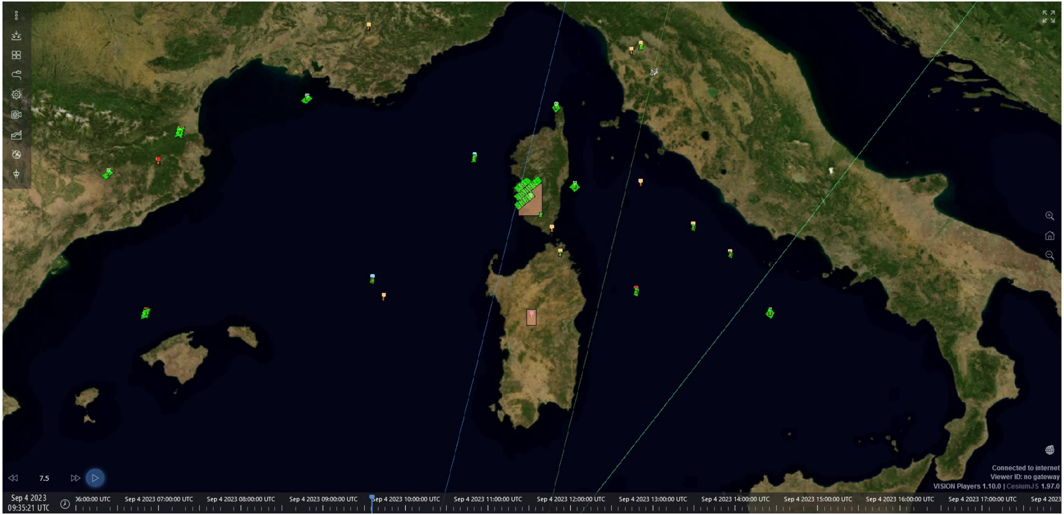
Multi-Satellite Problems [Pralet, 2025]



Click for video

Illustration 1: Constellation Management

Multi-Satellite Problems [Pralet, 2025]

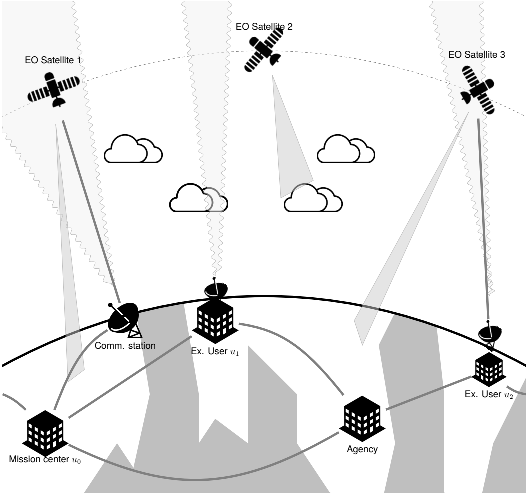


Click for video

Inter-Exclusive Coordinated Scheduling

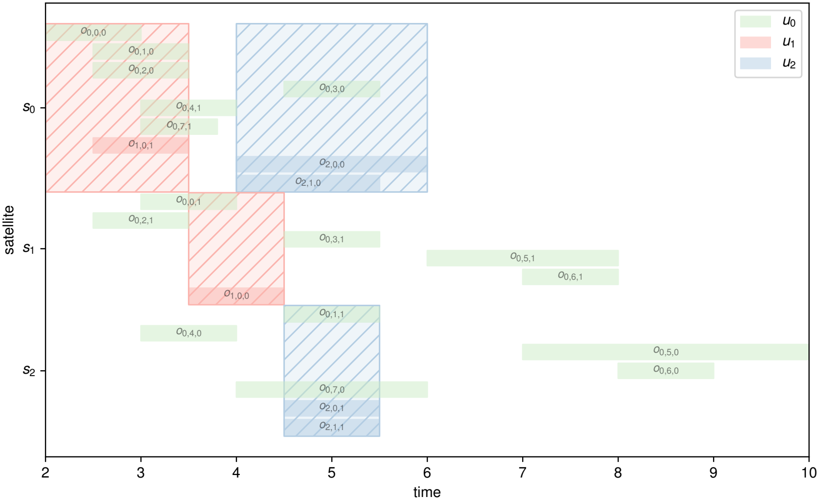
- We focus here on collective observation scheduling on a constellation where some users have exclusive access to some orbit portions

⇒ Answer to strong user expectations to benefit both from a shared system (to reduce costs) and a proprietary system (total control and confidentiality)



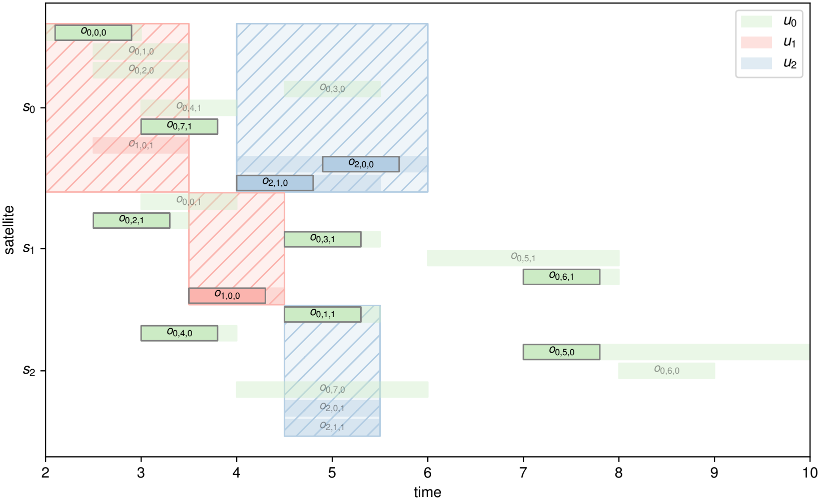
Scheduling Observations on an EOS Constellation

Illustrative Example



Scheduling Observations on an EOS Constellation

Illustrative Example



The Problems Behind

- How to **coordinate** exclusive user plans, **without disclosing private plans**, whilst meeting system constraints (memory, energy, etc.)
- How to couple private and non-private observations as to **maximize the system cost-efficiency**?

EOSCSP Model [Picard, 2022a]

Earth Observation Satellite Constellation Scheduling with Exclusives Problem is a tuple

$$P = \langle \mathcal{S}, \mathcal{U}, \mathcal{R}, \mathcal{O} \rangle$$

- $\mathcal{S} = \{s = \langle t_s^{\text{start}}, t_s^{\text{end}}, \kappa_s, \tau_s \rangle\}$ is a set of satellites
- $\mathcal{U} = \{u = \langle e_u, p_u \rangle\}$ is a set of users
- $\mathcal{R} = \{r = \langle t_r^{\text{start}}, t_r^{\text{end}}, \Delta_r, \rho_r, p_r, u_r, \theta_r \rangle\}$ is a set of requests
- $\mathcal{O} = \{o = \langle t_o^{\text{start}}, t_o^{\text{end}}, \Delta_o, r_o, \rho_o, s_o, u_o, p_o \rangle\}$ is a set of observation opportunities

A *solution* to an EOSCSP is a mapping $\mathcal{M} = \{(o, t) \mid o \in \mathcal{O}, t \in [t_o^{\text{start}}, t_o^{\text{end}}]\}$
s.t. the overall reward is maximized (sum of the rewards of the scheduled observations):
$\textbf{argmax}_{\mathcal{M}} \sum_{(o,t)\in\mathcal{M}} \rho_o$

How to Solve EOSCSPs?

How to Solve EOSCSPs?

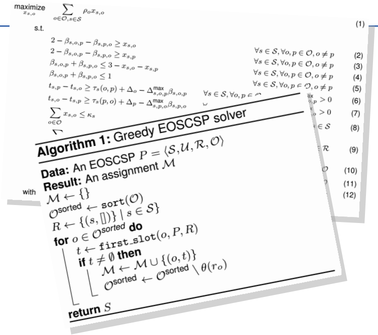
- Centralized allocation

How to Solve EOSCSPs?

$$
\begin{aligned}
\underset{x_{s,o}}{\text{maximize}} \quad & \sum_{o\in\mathcal{O},s\in\mathcal{S}} \rho_o x_{s,o} && (1)\\
\text{s.t.} \quad & 2-\beta_{s,o,p}-\beta_{s,p,o} \geq x_{s,o} && \forall s\in\mathcal{S},\forall o,p\in\mathcal{O},o\neq p \quad (2)\\
& 2-\beta_{s,o,p}-\beta_{s,p,o} \geq x_{s,p} && \forall s\in\mathcal{S},\forall o,p\in\mathcal{O},o\neq p \quad (3)\\
& \beta_{s,o,p}+\beta_{s,p,o} \leq 3-x_{s,o}-x_{s,p} && \forall s\in\mathcal{S},\forall o,p\in\mathcal{O},o\neq p \quad (4)\\
& \beta_{s,o,p}+\beta_{s,p,o} \leq 1 && \forall s\in\mathcal{S},\forall o,p\in\mathcal{O},o\neq p \quad (5)\\
& t_{s,p}-t_{s,o} \geq \tau_s(o,p)+\Delta_o-\Delta^{\text{max}}_{s,o,p}\beta_{s,o,p} && \forall s\in\mathcal{S},\forall o,p\in\mathcal{O},o\neq p,\text{s.t.}\Delta^{\text{max}}_{s,o,p}>0 \quad (6)\\
& t_{s,o}-t_{s,p} \geq \tau_s(p,o)+\Delta_p-\Delta^{\text{max}}_{s,p,o}\beta_{s,p,o} && \forall s\in\mathcal{S},\forall o,p\in\mathcal{O},o\neq p,\text{s.t.}\Delta^{\text{max}}_{s,p,o}>0 \quad (7)\\
& \sum_{o\in\mathcal{O}} x_{s,o} \leq \kappa_s && \forall s\in\mathcal{S} \quad (8)\\
& \sum_{o\in\theta(r)} x_{s,o} \leq 1 && \forall r\in\mathcal{R} \quad (9)\\
& x_{s,o}\in\{0,1\} && \forall s\in\mathcal{S},\forall o\in\mathcal{O} \quad (10)\\
& t_{s,o}\in[t_o^{\text{start}},t_o^{\text{end}}]\subset\mathbb{R} && \forall s\in\mathcal{S},\forall o\in\mathcal{O} \quad (11)\\
& \beta_{s,o,p}\in\{0,1\} && \forall s\in\mathcal{S},\forall o,p\in\mathcal{O},o\neq p \quad (12)\\
\text{with} \quad & \Delta^{\text{max}}_{s,o,p}=t_o^{\text{end}}-t_p^{\text{start}}+\Delta_o+\tau^s(o,p)
\end{aligned}
$$

- Centralized allocation
  - Exact solving (e.g. MILP), but won’t scale-up

How to Solve EOSCSPs?



Algorithm 1: Greedy EOSCSP solver

Data: An EOSCSP $P = \langle \mathcal{S}, \mathcal{U}, \mathcal{R}, \mathcal{O} \rangle$

Result: An assignment $\mathcal{M}$

$\mathcal{M} \leftarrow \{\}$

$\mathcal{O}^{\text{sorted}} \leftarrow \texttt{sort}(\mathcal{O})$

$R \leftarrow \{(s, []) \mid s \in \mathcal{S}\}$

for $o \in \mathcal{O}^{\text{sorted}}$ **do**

$t \leftarrow \texttt{first\_slot}(o, P, R)$

if $t \neq \emptyset$ **then**

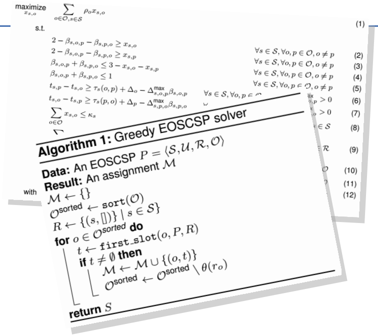
$\mathcal{M} \leftarrow \mathcal{M} \cup \{(o, t)\}$

$\mathcal{O}^{\text{sorted}} \leftarrow \mathcal{O}^{\text{sorted}} \setminus \theta(r_o)$

return S

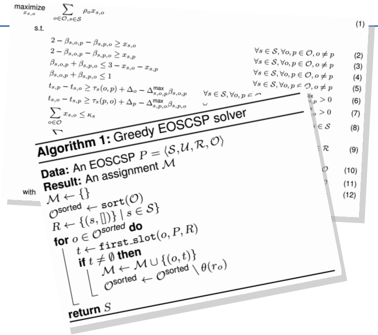
- Centralized allocation
  - Exact solving (e.g. MILP), but won't scale-up
  - Heuristic solving (e.g. greedy)

How to Solve EOSCSPs?



- Centralized allocation
  - Exact solving (e.g. MILP), but won't scale-up
  - Heuristic solving (e.g. greedy)
  - ✗ private plan disclosure

How to Solve EOSCSPs?



maximize$_{x_{s,o}}$ $\sum_{o\in\mathcal{O}, s\in\mathcal{S}} \rho_o x_{s,o}$ (1)

s.t.

$2 - \beta_{s,o,p} - \beta_{s,p,o} \geq x_{s,o}$ $\quad \forall s \in \mathcal{S}, \forall o,p \in \mathcal{O}, o \neq p$ (2)

$2 - \beta_{s,o,p} - \beta_{s,p,o} \geq x_{s,p}$ $\quad \forall s \in \mathcal{S}, \forall o,p \in \mathcal{O}, o \neq p$ (3)

$\beta_{s,o,p} + \beta_{s,p,o} \leq 3 - x_{s,o} - x_{s,p}$ $\quad \forall s \in \mathcal{S}, \forall o,p \in \mathcal{O}, o \neq p$ (4)

$\beta_{s,o,p} + \beta_{s,p,o} \leq 1$ $\quad \forall s \in \mathcal{S}, \forall o,p \in \mathcal{O}, o \neq p$ (5)

$t_{s,p} - t_{s,o} \geq \tau_s(o,p) + \Delta_o - \Delta^{\max}_{s,o,p}\beta_{s,o,p}$ (6)

$t_{s,o} - t_{s,p} \geq \tau_s(p,o) + \Delta_p - \Delta^{\max}_{s,p,o}\beta_{s,p,o}$ (7)

$\sum_{o\in\mathcal{O}} x_{s,o} \leq \kappa_s$ (8)

(9) (10) (11) (12)

with

Algorithm 1: Greedy EOSCSP solver

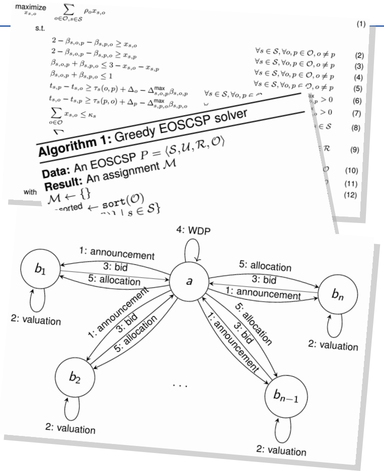
Data: An EOSCSP $P = \langle \mathcal{S}, \mathcal{U}, \mathcal{R}, \mathcal{O} \rangle$
Result: An assignment $\mathcal{M}$

$\mathcal{M} \leftarrow \{\}$
$\mathcal{O}^{\text{sorted}} \leftarrow \texttt{sort}(\mathcal{O})$
$R \leftarrow \{(s, []) \mid s \in \mathcal{S}\}$
for $o \in \mathcal{O}^{\text{sorted}}$ **do**
  $t \leftarrow \texttt{first\_slot}(o, P, R)$
  if $t \neq \emptyset$ **then**
    $\mathcal{M} \leftarrow \mathcal{M} \cup \{(o,t)\}$
    $\mathcal{O}^{\text{sorted}} \leftarrow \mathcal{O}^{\text{sorted}} \setminus \theta(r_o)$
return S

- Centralized allocation
  - Exact solving (e.g. MILP), but won’t scale-up
  - Heuristic solving (e.g. greedy)
  - ✗ private plan disclosure
- Distributed allocation

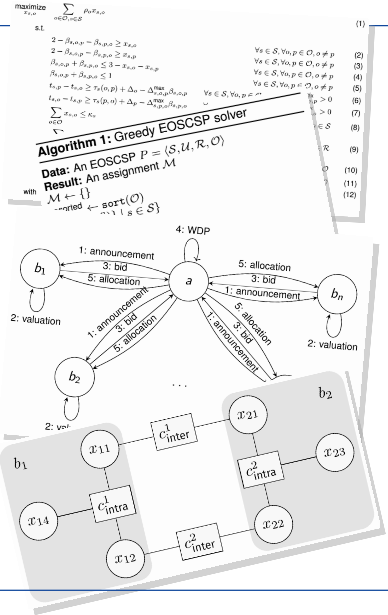
How to Solve EOSCSPs?

- Centralized allocation
  - Exact solving (e.g. MILP), but won't scale-up
  - Heuristic solving (e.g. greedy)
  - ✗ private plan disclosure
- Distributed allocation
  - Auctions (e.g. PSI, SSI, CBBA)



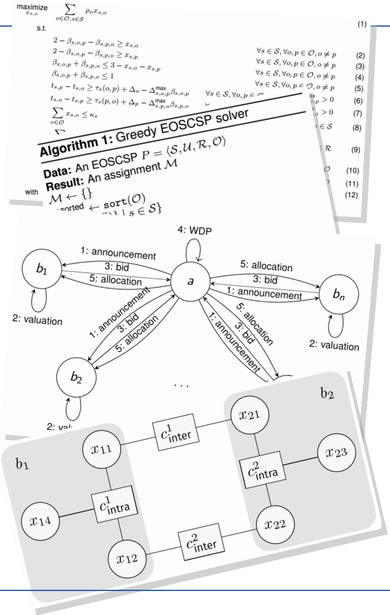
How to Solve EOSCSPs?

- Centralized allocation
  - Exact solving (e.g. MILP), but won't scale-up
  - Heuristic solving (e.g. greedy)
  - ✗ private plan disclosure
- Distributed allocation
  - Auctions (e.g. PSI, SSI, CBBA)
  - Distributed optimization (e.g. DCOPs)



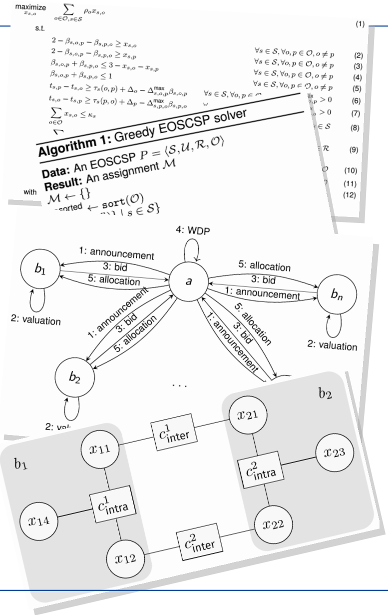
How to Solve EOSCSPs?

- Centralized allocation
  - Exact solving (e.g. MILP), but won't scale-up
  - Heuristic solving (e.g. greedy)
  - ✗ private plan disclosure
- Distributed allocation
  - Auctions (e.g. PSI, SSI, CBBA)
  - Distributed optimization (e.g. DCOPs)
  - ✓ plans remain private



How to Solve EOSCSPs?

- Centralized allocation
  - Exact solving (e.g. MILP), but won't scale-up
  - Heuristic solving (e.g. greedy)
  - ✗ private plan disclosure
- Distributed allocation
  - Auctions (e.g. PSI, SSI, CBBA)
  - Distributed optimization (e.g. DCOPs)
  - ✓ plans remain private
  - ⚠ requires some coordination/communication



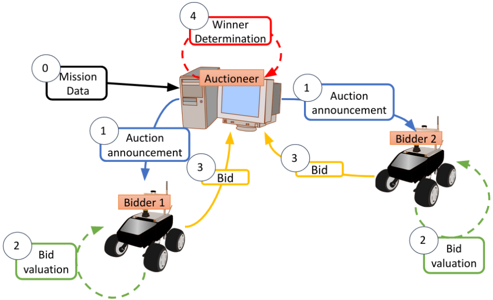
Auction-based Coordination for EOSCSP

Focus on Resource/Task Allocation

Many application fields, as Collective Robotics, make use of market-based approach to allocate tasks/resources to robots

- A set of **resources** (robots, satellites, etc.), $R = \{r_1, \ldots, r_{|R|}\}$
- A set of **tasks**, $T = \{t_1, \ldots, t_{|T|}\}$, each having a time-related and operation constraints
- Find an allocation of tasks to resources, wrt. some consistency constraints

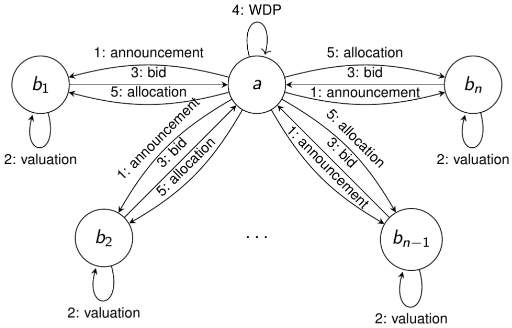
≈ **multi-item allocation**: each resource is allocated several tasks (bundle)



Auction-based Coordination for EOSCSP

Allocating non exclusive observations to best exclusive portions

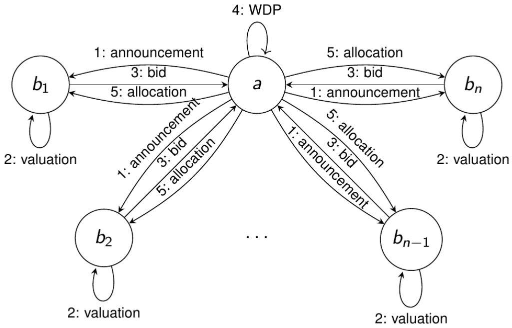
Auction-based approches are relevant for satellite task allocation [Phillips and Parra, 2021]



Auction-based Coordination for EOSCSP

Allocating non exclusive observations to best exclusive portions

Auction-based approches are relevant for satellite task allocation [Phillips and Parra, 2021]

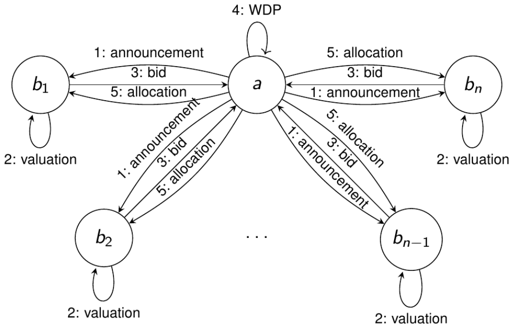


- **Combinatorial Auctions** (CA) [Cramton et al., 2010]

Auction-based Coordination for EOSCSP

Allocating non exclusive observations to best exclusive portions

Auction-based approches are relevant for satellite task allocation [Phillips and Parra, 2021]

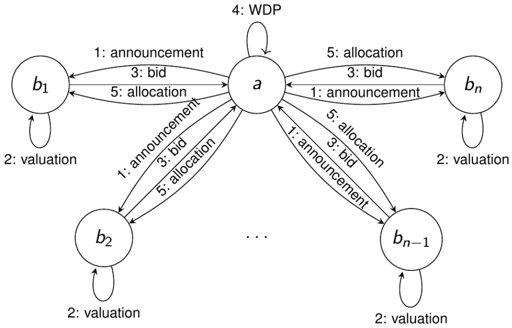


- **Combinatorial Auctions** (CA) [Cramton et al., 2010]
- **Parallel Single Item Auctions** (PSI) [Koenig et al., 2006]

Auction-based Coordination for EOSCSP

Allocating non exclusive observations to best exclusive portions

Auction-based approches are relevant for satellite task allocation [Phillips and Parra, 2021]

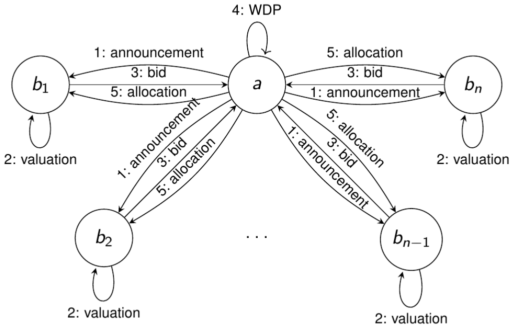


- **Combinatorial Auctions** (CA) [Cramton et al., 2010]
- **Parallel Single Item Auctions** (PSI) [Koenig et al., 2006]
  - Each agent bids on the *whole set of tasks* in parallel

Auction-based Coordination for EOSCSP

Allocating non exclusive observations to best exclusive portions

Auction-based approches are relevant for satellite task allocation [Phillips and Parra, 2021]

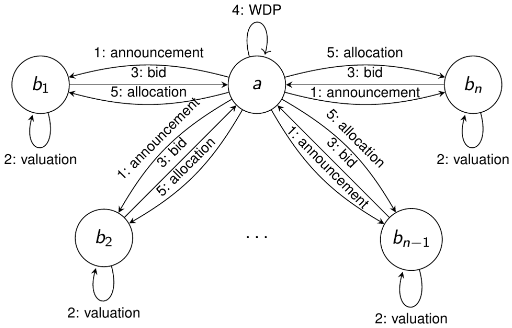


- **Combinatorial Auctions** (CA) [Cramton et al., 2010]
- **Parallel Single Item Auctions** (PSI) [Koenig et al., 2006]
  - Each agent bids on the *whole set of tasks* in parallel
- **Sequential Single Item Auctions** (SSI) [Lagoudakis et al., 2005]

Auction-based Coordination for EOSCSP

Allocating non exclusive observations to best exclusive portions

Auction-based approches are relevant for satellite task allocation [Phillips and Parra, 2021]

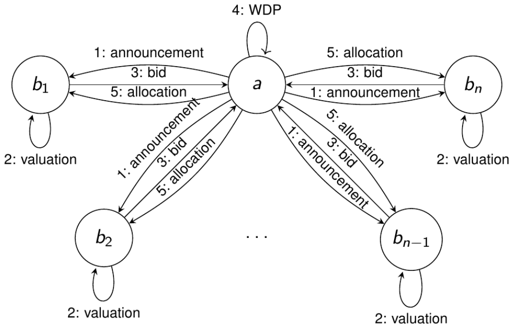


- **Combinatorial Auctions** (CA) [Cramton et al., 2010]
- **Parallel Single Item Auctions** (PSI) [Koenig et al., 2006]
  - Each agent bids on the *whole set of tasks* in parallel
- **Sequential Single Item Auctions** (SSI) [Lagoudakis et al., 2005]
  - Each agent *sequentially* bids on a *single task* wrt to the already allocated tasks

Auction-based Coordination for EOSCSP

Allocating non exclusive observations to best exclusive portions

Auction-based approches are relevant for satellite task allocation [Phillips and Parra, 2021]

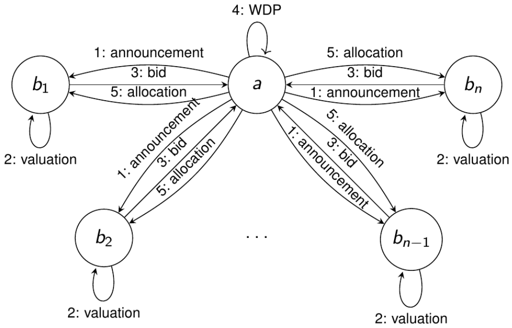


- **Combinatorial Auctions** (CA) [Cramton et al., 2010]
- **Parallel Single Item Auctions** (PSI) [Koenig et al., 2006]
  - Each agent bids on the *whole set of tasks* in parallel
- **Sequential Single Item Auctions** (SSI) [Lagoudakis et al., 2005]
  - Each agent *sequentially* bids on a *single task* wrt to the already allocated tasks
- **Consensus-based Bundle Auction** (CBBA) [Choi et al., 2009]

Auction-based Coordination for EOSCSP

Allocating non exclusive observations to best exclusive portions

Auction-based approches are relevant for satellite task allocation [Phillips and Parra, 2021]

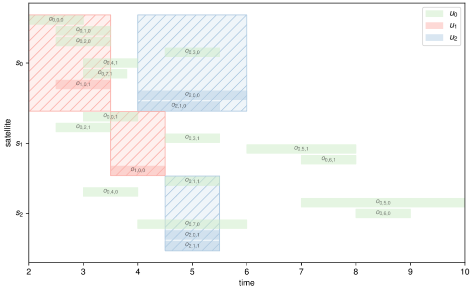


- **Combinatorial Auctions** (CA) [Cramton et al., 2010]
- **Parallel Single Item Auctions** (PSI) [Koenig et al., 2006]
  - Each agent bids on the *whole set of tasks* in parallel
- **Sequential Single Item Auctions** (SSI) [Lagoudakis et al., 2005]
  - Each agent *sequentially* bids on a *single task* wrt to the already allocated tasks
- **Consensus-based Bundle Auction** (CBBA) [Choi et al., 2009]
  - Each agent bids on some *bundle of tasks* and *converge to a consensus* with other agents

Applying Auction-based Allocation to EOSCSP

General Scheme

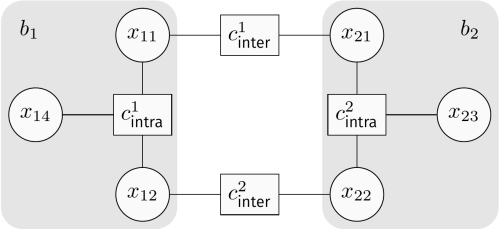
1. Identify non exclusive requests possibly fulfilled in exclusive portions
2. Send identified requests to exclusive users
3. Solve the allocation problem using PSI, SSI or CBBA
   - Bids are computed as the **best marginal costs** of integrating requests in their current plans (which amounts to solve scheduling problems...)
4. Allocate as many remaining requests outside exclusive windows



DCOP-based Coordination for EOSCSP

Allocating non exclusive observations to best exclusive portions

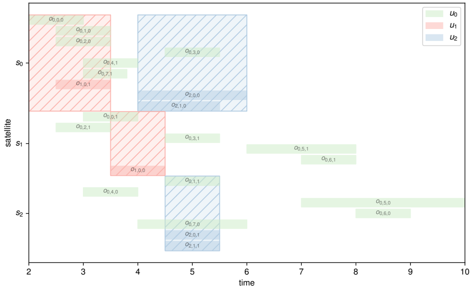
- Consider the **collective decision** for allocating non exclusive tasks to exclusive windows
- Collective decision to coordinate exclusive users' decisions modeled as a **distributed constraint optimization problem** (DCOP)
- As for auctions, exclusive users aim to **minimizing the marginal cost** of integrating non exclusive tasks in their schedule, while meeting some operational constraints



DCOP-based Coordination for EOSCSP

General Scheme

1. Identify non exclusive requests possibly fulfilled in exclusive windows
2. Send each identified request r to exclusives users, one by one
3. Solve the problem of r using a DCOP solution method (e.g. DPOP [Adrian Petcu and Boi Faltings, 2005])
   - Costs are computed as the **best marginal cost** of integrating requests in their current plan (which amounts to solve a scheduling problem...)
4. Allocate as many remaining requests outside exclusive windows



A DCOP $\langle \mathcal{A}, \mathcal{X}, \mathcal{D}, \mathcal{C}, \mu \rangle$ is defined for a given request r, and a current scheduling

A DCOP $\langle \mathcal{A}, \mathcal{X}, \mathcal{D}, \mathcal{C}, \mu \rangle$ is defined for a given request r, and a current scheduling

- The agents are the exclusive users which can potentially schedule r:

$$\mathcal{A} = \{u \in \mathcal{U}^{\text{ex}} | \exists (s, (t_u^{\text{start}}, t_u^{\text{end}})) \in e_u, \exists o \in \theta_r \ \text{ s.t. } s_o = s, [t_u^{\text{start}}, t_u^{\text{end}}] \cap [t_o^{\text{start}}, t_o^{\text{end}}] \neq \emptyset\} \quad (1)$$

A DCOP $\langle \mathcal{A}, \mathcal{X}, \mathcal{D}, \mathcal{C}, \mu \rangle$ is defined for a given request r, and a current scheduling

- The agents are the exclusive users which can potentially schedule r:

$$\mathcal{A} = \{u \in \mathcal{U}^{\text{ex}} | \exists (s, (t_u^{\text{start}}, t_u^{\text{end}})) \in e_u, \exists o \in \theta_r \text{ s.t. } s_o = s, [t_u^{\text{start}}, t_u^{\text{end}}] \cap [t_o^{\text{start}}, t_o^{\text{end}}] \neq \emptyset\} \quad (1)$$

- Each agent u owns binary decision variables, one for each observation $o \in \mathcal{O}[u]^r$ and exclusive e in its exclusives e_u, stating whether it schedules o in e or not:

$$\mathcal{X} = \{x_{e,o} | e \in \textstyle\bigcup_{u \in \mathcal{A}} e_u, o \in \mathcal{O}[u]^r\} \quad (2)$$

$$\mathcal{D} = \{\mathcal{D}_{x_{e,o}} = \{0, 1\} | x_{e,o} \in \mathcal{X}\} \quad (3)$$

A DCOP $\langle \mathcal{A}, \mathcal{X}, \mathcal{D}, \mathcal{C}, \mu \rangle$ is defined for a given request r, and a current scheduling

- The agents are the exclusive users which can potentially schedule r:

$$\mathcal{A} = \{u \in \mathcal{U}^{\text{ex}} | \exists (s, (t_u^{\text{start}}, t_u^{\text{end}})) \in e_u, \exists o \in \theta_r \text{ s.t. } s_o = s, [t_u^{\text{start}}, t_u^{\text{end}}] \cap [t_o^{\text{start}}, t_o^{\text{end}}] \neq \emptyset\} \quad (1)$$

- Each agent u owns binary decision variables, one for each observation $o \in \mathcal{O}[u]^r$ and exclusive e in its exclusives e_u, stating whether it schedules o in e or not:

$$\mathcal{X} = \{x_{e,o} | e \in \bigcup_{u \in \mathcal{A}} e_u, o \in \mathcal{O}[u]^r\} \quad (2)$$

$$\mathcal{D} = \{\mathcal{D}_{x_{e,o}} = \{0, 1\} | x_{e,o} \in \mathcal{X}\} \quad (3)$$

with $\mathcal{O}[u]^r = \{o \in \theta r | \exists (s, (t_u^{\text{start}}, t_u^{\text{end}})) \in e_u, \text{ s.t. } s_o = s, [t_u^{\text{start}}, t_u^{\text{end}}] \cap [t_o^{\text{start}}, t_o^{\text{end}}] \neq \emptyset\}$ are observations related to request r that can be scheduled on u's exclusives

A DCOP $\langle \mathcal{A}, \mathcal{X}, \mathcal{D}, \mathcal{C}, \mu \rangle$ is defined for a given request r, and a current scheduling

- The agents are the exclusive users which can potentially schedule r:

$$\mathcal{A} = \{u \in \mathcal{U}^{\text{ex}} | \exists (s, (t_u^{\text{start}}, t_u^{\text{end}})) \in e_u, \exists o \in \theta_r \text{ s.t. } s_o = s, [t_u^{\text{start}}, t_u^{\text{end}}] \cap [t_o^{\text{start}}, t_o^{\text{end}}] \neq \emptyset\} \quad (1)$$

- Each agent u owns binary decision variables, one for each observation $o \in \mathcal{O}[u]^r$ and exclusive e in its exclusives e_u, stating whether it schedules o in e or not:

$$\mathcal{X} = \{x_{e,o} | e \in \bigcup_{u \in \mathcal{A}} e_u, o \in \mathcal{O}[u]^r\} \quad (2)$$

$$\mathcal{D} = \{\mathcal{D}_{x_{e,o}} = \{0,1\} | x_{e,o} \in \mathcal{X}\} \quad (3)$$

with $\mathcal{O}[u]^r = \{o \in \theta r | \exists (s, (t_u^{\text{start}}, t_u^{\text{end}})) \in e_u, \text{ s.t. } s_o = s, [t_u^{\text{start}}, t_u^{\text{end}}] \cap [t_o^{\text{start}}, t_o^{\text{end}}] \neq \emptyset\}$ are observations related to request r that can be scheduled on u's exclusives

- μ associates each variable $x_{e,o}$ to e's owner

DCOP-based Coordination for EOSCSP (cont.)

DCOP Model

- Constraints should check that at most one observation is scheduled per request (4), that satellites are not overloaded (5), that at most one agent serves the same observation (6)

$$\sum\nolimits_{e \in \bigcup_{u \in \mathcal{A}} e_u} x_{e,o} \leq 1, \quad \forall u \in \mathcal{X}, \forall o \in \mathcal{O}[u]^r \tag{4}$$

$$\sum\nolimits_{o \in \{o \in \mathcal{O}[u]^r \mid u \in \mathcal{A}, s_o = s\}, e \in \bigcup_{u \in \mathcal{A}} e_u} x_{e,o} \leq \kappa_s^*, \ \forall s \in \mathcal{S} \tag{5}$$

$$\sum\nolimits_{e \in \bigcup_{u \in \mathcal{A}} e_u} x_{e,o} \leq 1, \quad \forall o \in \mathcal{O} \tag{6}$$

DCOP-based Coordination for EOSCSP (cont.)

DCOP Model

- Constraints should check that at most one observation is scheduled per request (4), that satellites are not overloaded (5), that at most one agent serves the same observation (6)

$$\sum_{e \in \bigcup_{u \in \mathcal{A}} e_u} x_{e,o} \leq 1, \quad \forall u \in \mathcal{X}, \forall o \in \mathcal{O}[u]^r \tag{4}$$

$$\sum_{o \in \{o \in \mathcal{O}[u]^r | u \in \mathcal{A}, s_o = s\}, e \in \bigcup_{u \in \mathcal{A}} e_u} x_{e,o} \leq \kappa_s^*, \ \forall s \in \mathcal{S} \tag{5}$$

$$\sum_{e \in \bigcup_{u \in \mathcal{A}} e_u} x_{e,o} \leq 1, \quad \forall o \in \mathcal{O} \tag{6}$$

- The cost to integrate an observation in the current user's schedule should be assessed to guide the optimization process

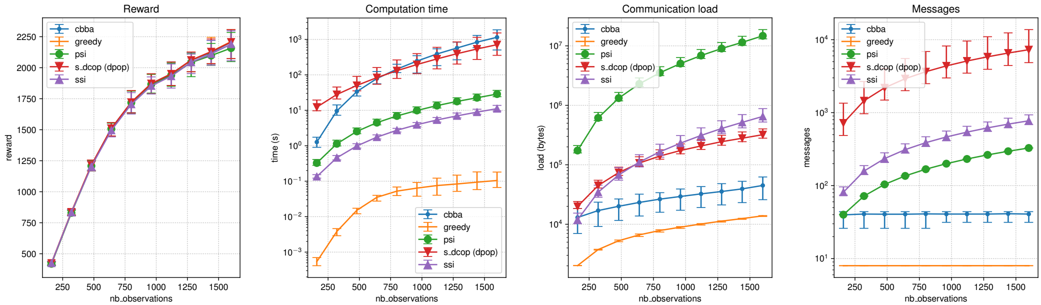
$$c(x_{e,o}) = \pi(o, \mathcal{M}_{u_o}), \quad \forall x_{e,o} \in \mathcal{X} \tag{7}$$

where π evaluates the best cost obtained when scheduling o and any combination of observations from $\mathcal{M}_{u_o}$, as to consider all possible revisions of u_o's current schedule

$$\mathcal{C} = \{(4), (5), (6), (7)\} \tag{8}$$

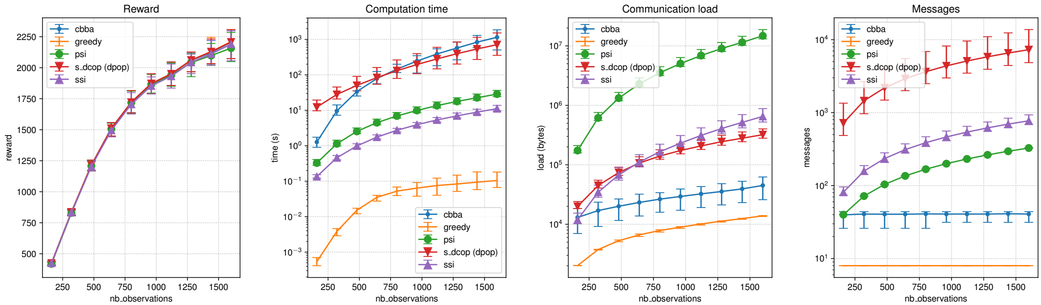
Highly conflicting randomly generated problems

5-min horizon with overlapping requests and limited capacity



Highly conflicting randomly generated problems

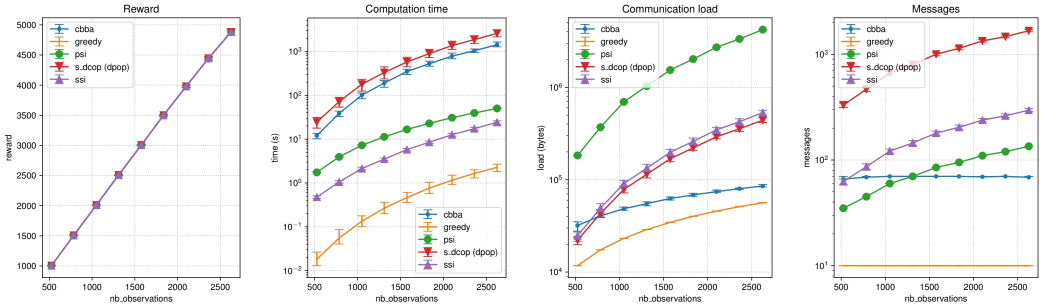
5-min horizon with overlapping requests and limited capacity



- ✗ cbba and s_dcop requires extra-computation time ($\approx$ 1000s)
- ✓ cbba and s_dcop provide the best solutions wrt. reward
- ✓ cbba exchanges fewer messages of small size
- ✓ ssi remains the best compromise wrt. solution quality, computation time and communication load

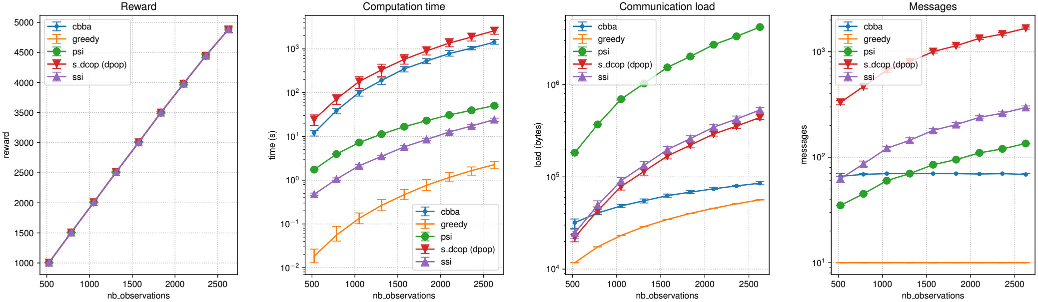
Realistic randomly generated problems

6-hour horizon with numerous requests and large capacity



Realistic randomly generated problems

6-hour horizon with numerous requests and large capacity



✓ cbba does require less time to compute than s_dcop

✓ s_dcop and cbba can perform many computation concurrently

⇒ There is room for computation speedup in real distributed settings

Where to find detailed info?

- Initial model definition [Picard, 2022a]
- Auction-based and DCOP-based solution methods [ibid.]
- More complex requests and decentralized auctions [Picard, 2023a]
- Some data [Picard, 2023b]

Today's Menu

On-demand Transport

Mines Saint-Etienne [Daoud et al., 2021a, 2020, 2021b,c,d, 2023], Renault Innovation

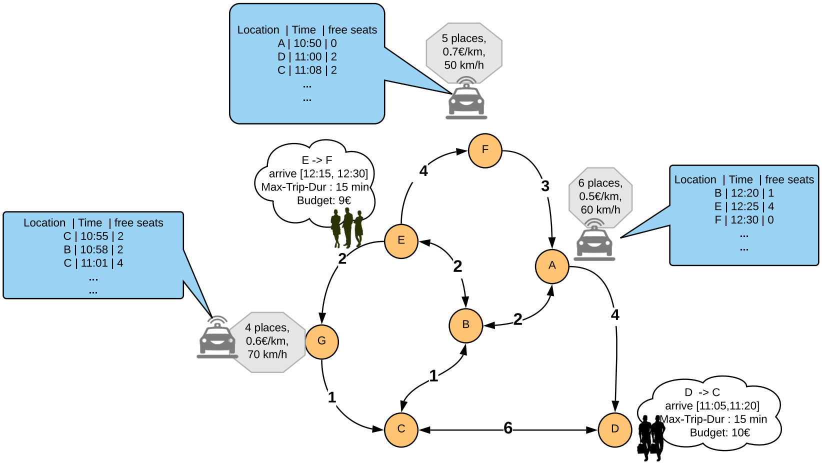


Figure: Dial A Ride Problem (DARP)

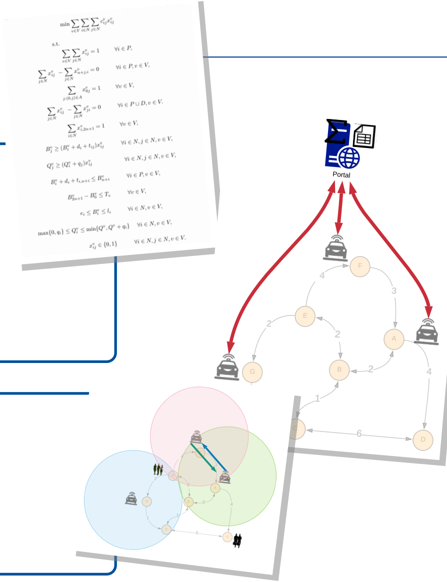
Existing Approaches

Centralized dispatch (conventional)

- Requests are gathered from a central portal
- Integer Linear Programming (ILP)
  - $\Rightarrow$ NP-hard, lack of scalability
- Requires continuous access to the portal
  - $\Rightarrow$ costly, bottleneck, single point of failure

Decentralized dispatch (experimental)

- Decentralized autonomous decisions
  - $\Rightarrow$ Requires conflict detection and resolution protocols
- P2P communication
  - $\Rightarrow$ Requires communication model to ensure information sharing



Contributions and Core Concepts

Generic Autonomous Agent Model

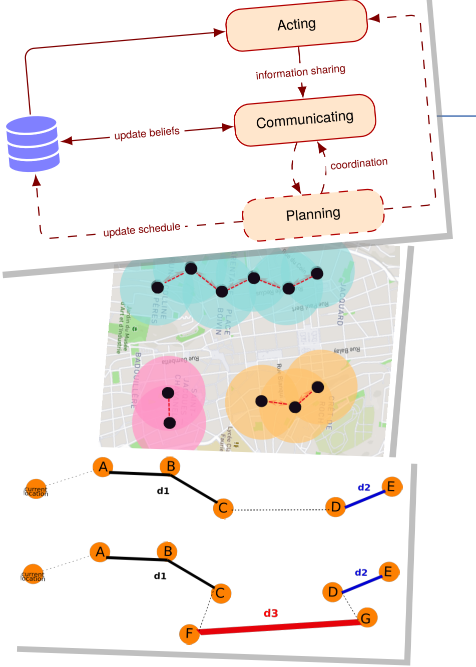
- Adjustable autonomy level
- Adjustable cooperation level
- Adjustable and dynamic allocation scheme

Communication Model

- Transitive V2V
- Dynamic

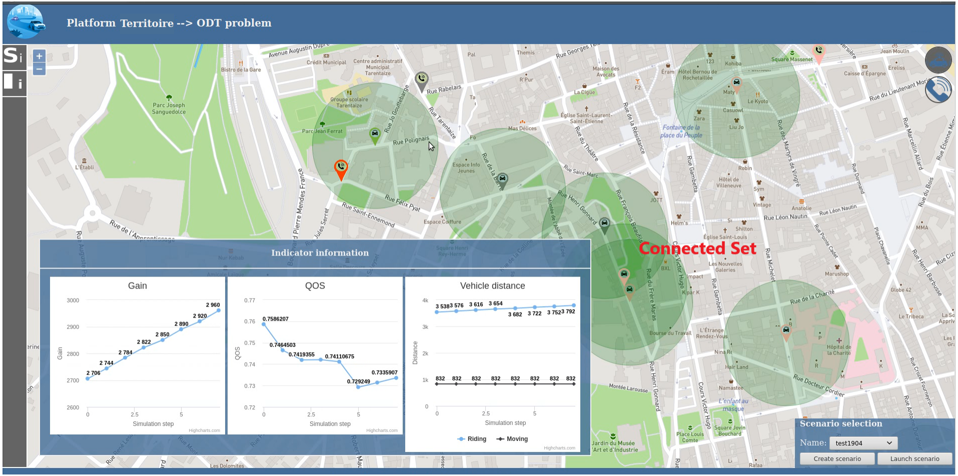
Insertion-cost Heuristic

- Marginal cost of inserting request
- Re-assessed when neighbors change



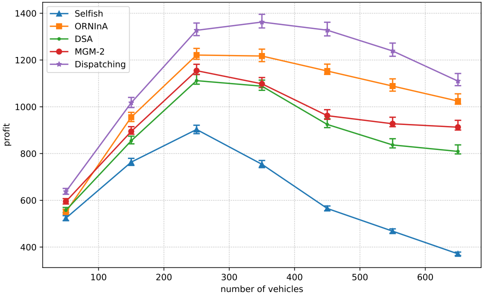
Experimental Evaluation

Simulation with synthetic (Saint-Étienne) and real data (NYC)

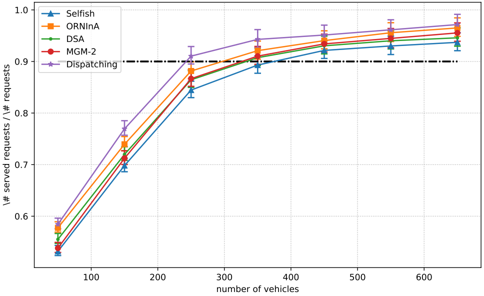


Sample Results

NYC Dataset



(a) QoB evolution with fleet size

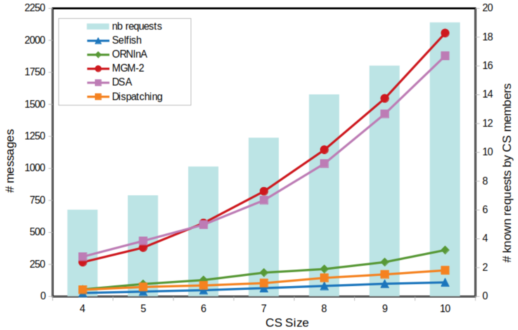


(b) QoS evolution with fleet size

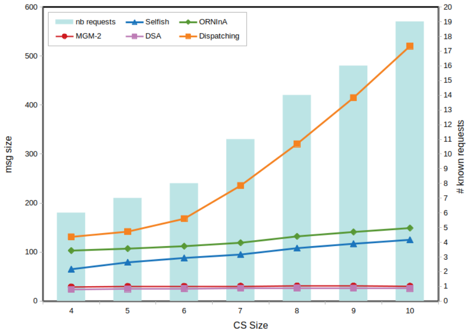
Figure: Solution quality evolution with fleet size

Sample Results (cont.)

NYC Dataset



(a) Average number of messages received by a vehicle in connected set



(b) Average message size received by a vehicle in connected set

Figure: Communication load evolution.

Today’s Menu

Illustration 3: Unmanned Aircraft System Traffic Management

Example: Urban UTM Scenario

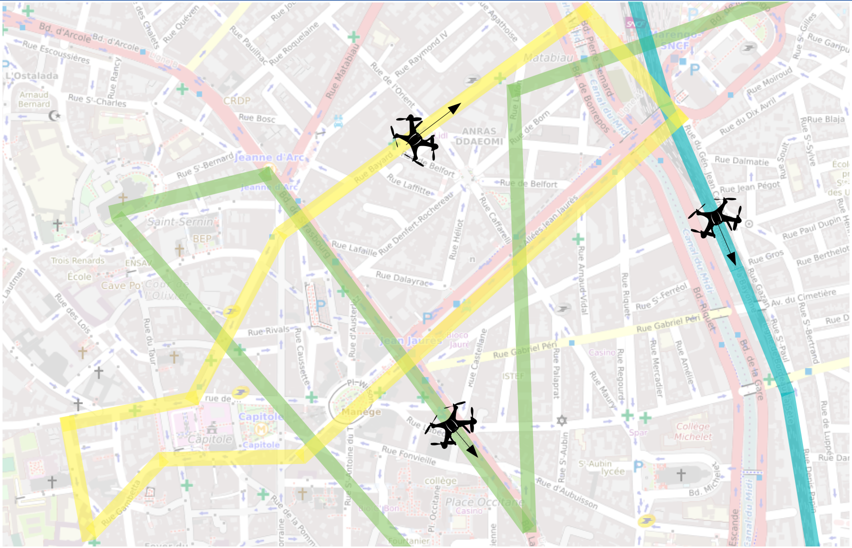


Illustration 3: Unmanned Aircraft System Traffic Management

Example: Urban UTM Scenario

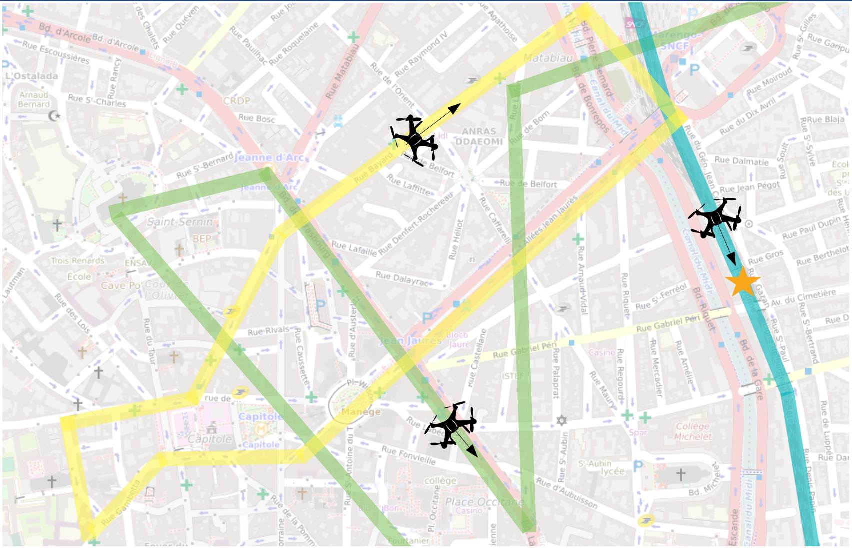


Illustration 3: Unmanned Aircraft System Traffic Management

Example: Urban UTM Scenario

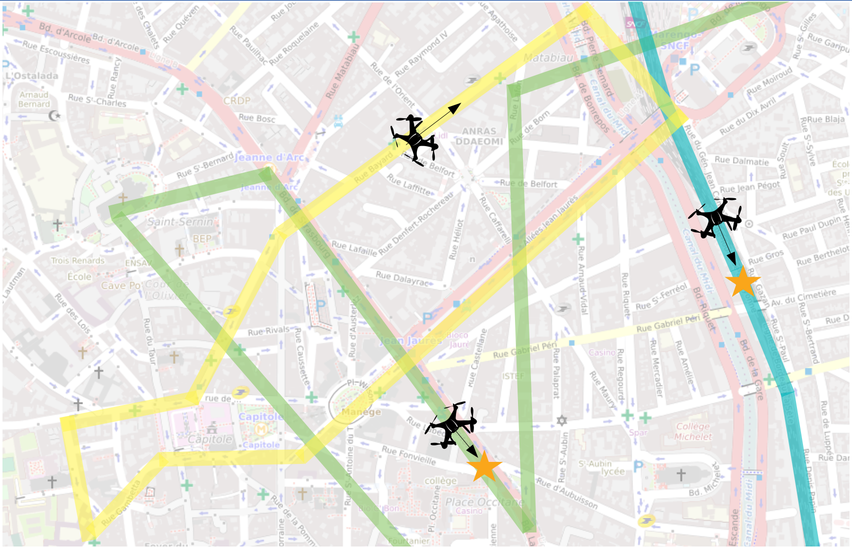


Illustration 3: Unmanned Aircraft System Traffic Management

Example: Urban UTM Scenario

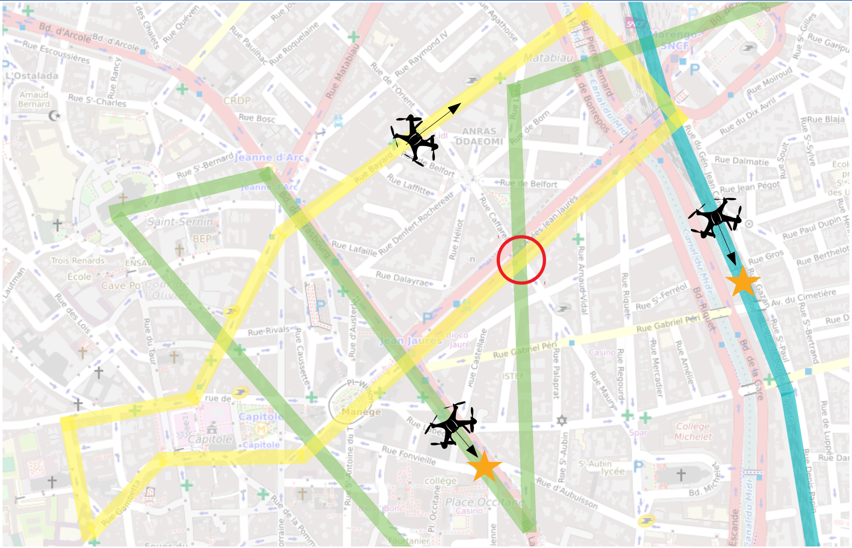


Illustration 3: Unmanned Aircraft System Traffic Management

Example: Urban UTM Scenario

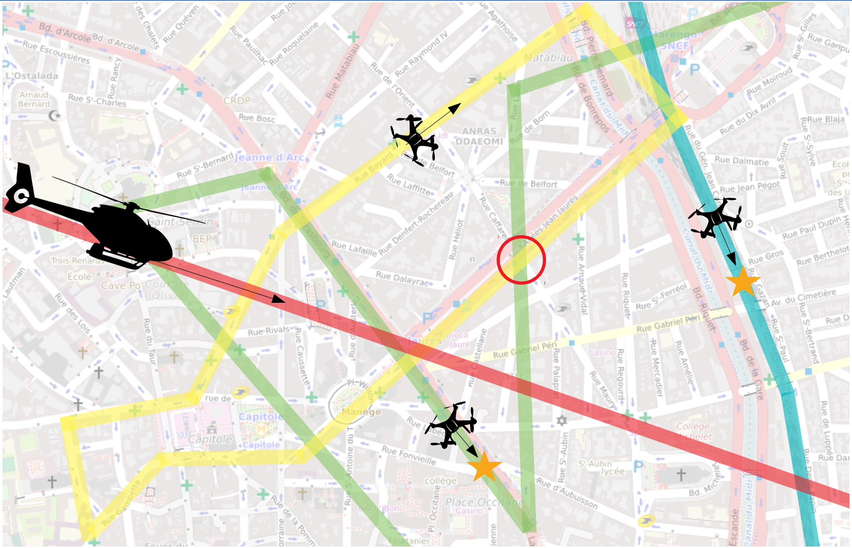


Illustration 3: Unmanned Aircraft System Traffic Management

Example: Urban UTM Scenario

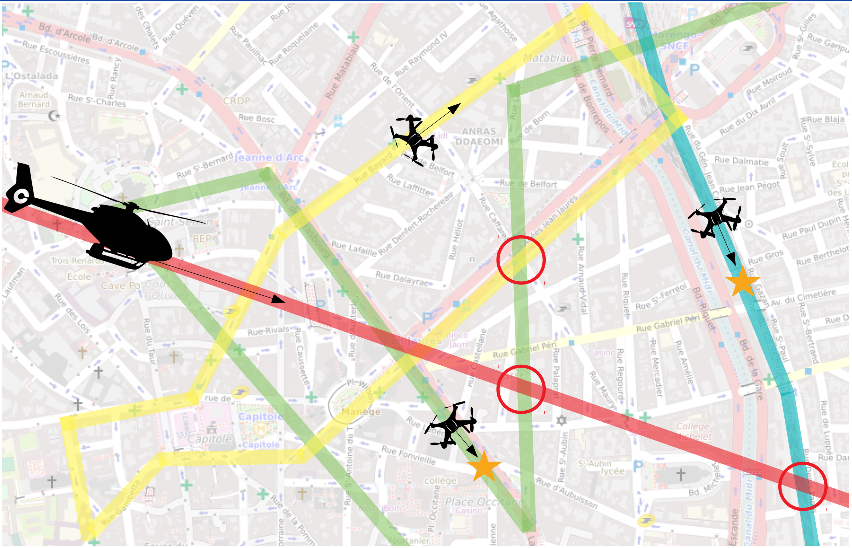


Illustration 3: Unmanned Aircraft System Traffic Management

Context and Vision

- **Concepts of operations** are still work in progress
  [Federal Aviation Agency, 2020; SESAR, 2019]
- Several **challenging optimization problems** identified [Hamadi, 2020]

Illustration 3: Unmanned Aircraft System Traffic Management

Context and Vision

- **Concepts of operations** are still work in progress [Federal Aviation Agency, 2020; SESAR, 2019]
- Several **challenging optimization problems** identified [Hamadi, 2020]

Our focus: 4D trajectory repair

- **Free Route** Airspace
- Decisions at the **UAS level**
- UAVs can directly exchange information via **V2V communication**
- Tactical and reactive **coordination mechanisms** between several (semi-)autonomous UAS
- Focus on small UAVs able to perform **stationary flight** and operating at low altitude (between 0m and 300m)

Contributions and Core Concepts

[Hamadi and Picard, 2024; Picard, 2022b]

Generic Autonomous UAV Model

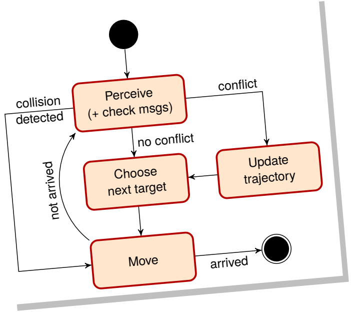
- Adjustable autonomy level
- Pluggable at UAS level
- Adjustable deconfliction protocol

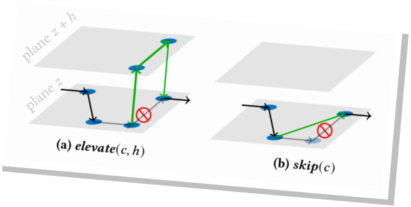
Corrective Actions

- 4D contract update
- Postpone, elevate, skip

Multi-criteria Valuation

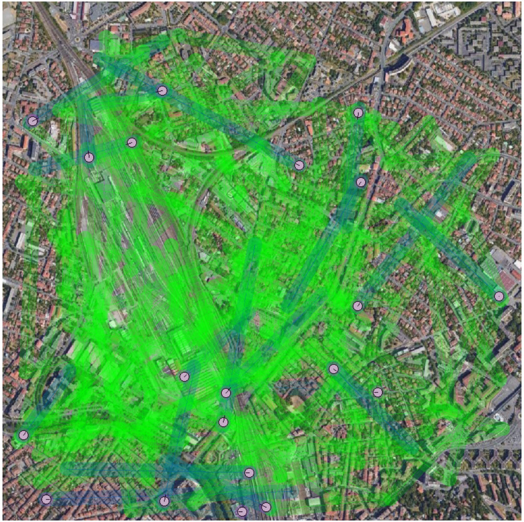
- Impact of a corrective action
- Safety, QoS, QoB, etc.





(a) *elevate*(c, h) (b) *skip*(c)

Experimental Evaluation

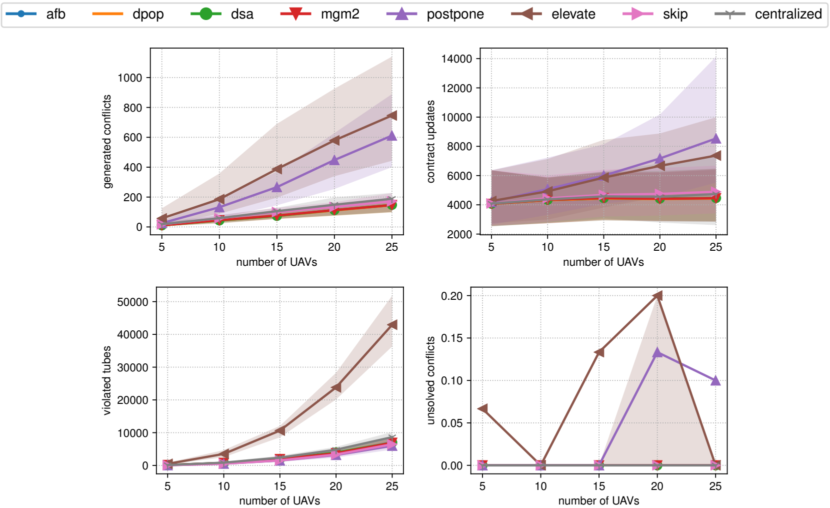


Environment

- An area of 2km by 2km, with vertical airspace planes at 20m, 40m and 60m
- $h_{max} = 18m.s^{-1}$, $v_{max} = 6m.s^{-1}$, $a_{max} = \Pi/2\text{rad}.s^{-1}$, $\Delta h_{\text{max}} = \Delta v_{\text{max}} = 6m.s^{-2}$, $\Delta a_{\text{max}} = \Pi/2\text{rad}.s^{-2}$
- Initial speed is set to $(0, 0, 0)$
- Initial UAV trajectories are randomly generated with 60 way-points
- Safety tubes are defined by $(h, v, t) = (30, 15, 1)$
- Number of UAVs in $\{5, 10, 15, 20, 25\}$

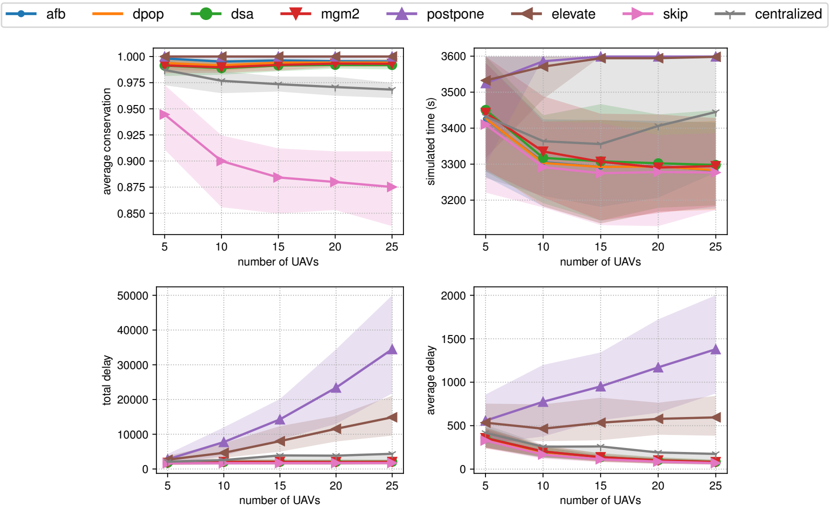
Result Analysis

Without coordination, numerous conflicts and/or some violations



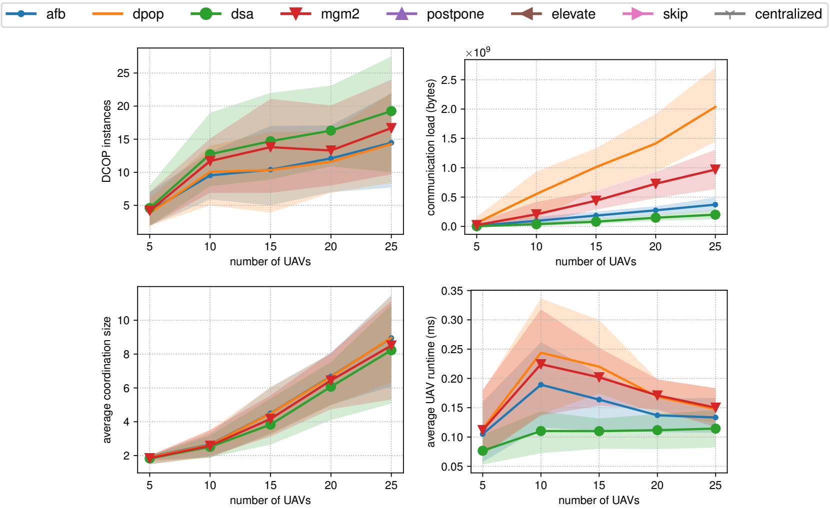
Result Analysis (cont.)

Without coordination, increased delays or reduced QoS



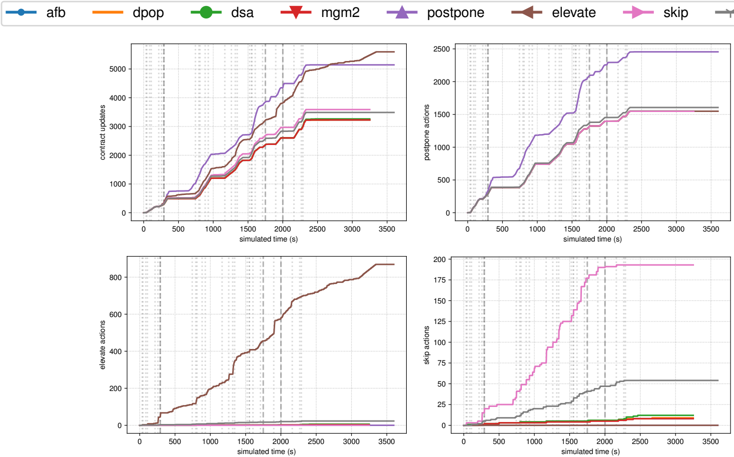
Result Analysis (cont.)

Coordination group size are small ⇒ communication/computation overload are limited



Result Analysis (cont.)

Focus on a specific instance



What About Auctions? And other Decision Criteria?

[Hamadi and Picard, 2024]

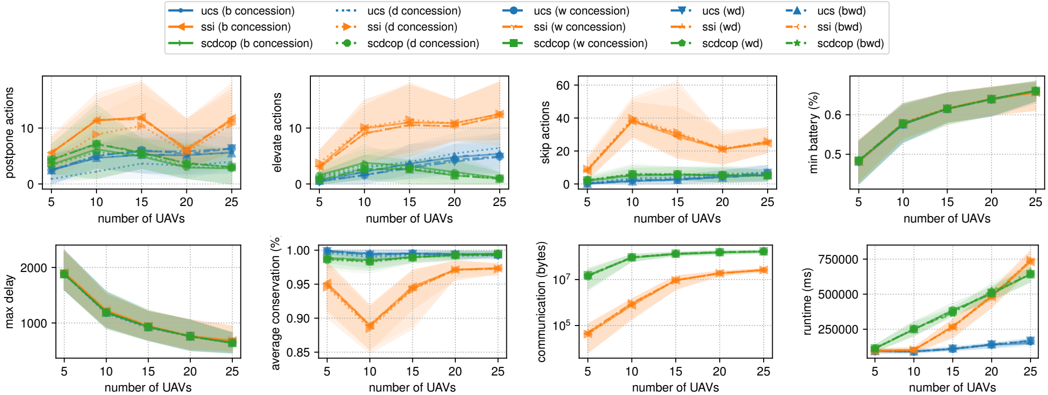


Figure: Average values over 20 instances for several performance metrics with increasing number of UAVs.

What About Auctions? And other Decision Criteria? (cont.)

[Hamadi and Picard, 2024]

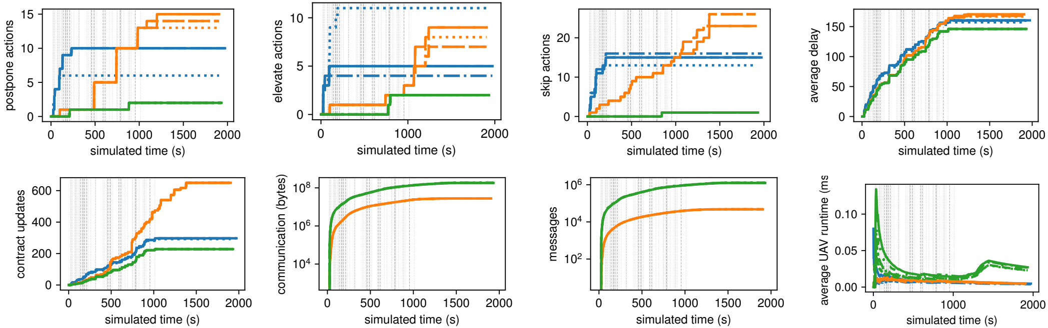


Figure: Results for one simulation with 25 UAVs and 10 emergency procedures (gray dashed) and 46 incidents (gray dotted).

Today's Menu

Conclusions

To sum up...

- Auctions and DCOPs are powerful tools to install **coordination in cooperative collectives**
- Many potential **applications**
  - On-demand transport, UTM, Satellite constellation management, IoT, Smart grids, ...
- **Agency** as a way to install encapsulation and **explanability**

To go beyond...

- **Non cooperative** settings
- Hybrid AI: **learning** and approximating costs
- **Security** of coordination protocols


Thank you for your attention!
Any question?
ONERA AI LAB

References

Chakraa, Hamza, François Guérin, Edouard Leclercq, and Dimitri Lefebvre (2023). "Optimization techniques for Multi-Robot Task Allocation problems: Review on the state-of-the-art". In: *Robotics and Autonomous Systems* 168, p. 104492. issn: 0921-8890. doi: https://doi.org/10.1016/j.robot.2023.104492. url: https://www.sciencedirect.com/science/article/pii/S0921889023001318.

Choi, Han-Lim, Luc Brunet, and Jonathan P. How (2009). "Consensus-Based Decentralized Auctions for Robust Task Allocation". In: *IEEE Trans. Robotics* 25.4, pp. 912–926. doi: 10.1109/TRO.2009.2022423.

Choudhury, Moumita, Saaduddin Mahmud, and Md. Mosaddek Khan (2020). "A Particle Swarm Based Algorithm for Functional Distributed Constraint Optimization Problems". In: *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7-12, 2020*. AAAI Press, pp. 7111–7118. url: https://ojs.aaai.org/index.php/AAAI/article/view/6198.

Cramton, Peter, Yoav Shoham, and Richard Steinberg, eds. (2010). *"Combinatorial Auctions"*. MIT Press.

Daoud, Alaa, Hiba Alqasir, Yazan Mualla, Amro Najjar, Gauthier Picard, and Flavien Balbo (2021a). "Towards Explainable Recommendations of Resource Allocation Mechanisms in On-Demand Transport Fleets". In: *Explainable and Transparent AI and Multi-Agent Systems, Third International Workshop (EXTRAAMAS 2021)*. Vol. 12688. Lecture Notes on Artificial Intelligence (LNAI). Springer International Publishing, pp. 95–117. doi: 10.1007/978-3-030-82017-6_7.

Daoud, Alaa, Flavien Balbo, Paolo Gianessi, and Gauthier Picard (2020). "Decentralized Insertion Heuristic with Runtime Optimization for On-demand Transport Scheduling". In: *ATT2020 (11th International Workshop on Agents in Traffic and Transportation)*. Ed. by Ivana Dusparic, Marin Lujak, Franziska Klügl, and Giuseppe Vizzari. CEUR Workshop Proceedings, pp. 9–15. url: https://sites.google.com/unimib.it/att2020/.

References (cont.)

Daoud, Alaa, Flavien Balbo, Paolo Gianessi, and Gauthier Picard (May 2021b). "A Generic Agent Model Towards Comparing Resource Allocation Approaches to On-demand Transport with Autonomous Vehicles". In: *The 12th Workshop on Optimization and Learning in Multiagent Systems (OptLearnMAS-21)*. at AAMAS 2021 (virtual) London, United Kingdom. URL: https://hal.archives-ouvertes.fr/hal-03201325.

— (2021c). "A Generic Multi-Agent Model for Resource Allocation Strategies in Online On-Demand Transport with Autonomous Vehicles". In: *Proceedings of the 20th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2021)*. Ed. by U. Endriss, A. Nowé, F. Dignum, and A. Lomuscio. Extended abstract. International Foundation for Autonomous Agents and Multiagent Systems, pp. 1489–1491. DOI: https://dl.acm.org/doi/10.5555/3463952.3464135. URL: https://dl.acm.org/doi/10.5555/3463952.3464135. AR=40%.

— (Feb. 2021d). "ORNInA: A Decentralized, Auction-based Multi-agent Coordination in ODT Systems". In: *AI Communications* 34.1, pp. 37–53. DOI: 10.3233/AIC-201579. URL: https://content.iospress.com/articles/ai-communications/aic201579.

Daoud, Alaa, Gauthier Picard, Hiba Alqasir, Paolo Gianessi, and Flavien Balbo (2023). "Communication-wise Comparison of the Online Resource Allocation Methods in CAV Fleets". In: *International Conference on Ambient Systems, Networks and Technologies (ANT-23)*. Vol. 220. Elsevier, pp. 299–306. DOI: 10.1016/j.procs.2023.03.039. AR=31%.

Delle Fave, F.M., R. Stranders, A. Rogers, and N.R. Jennings (2011). "Bounded Decentralised Coordination over Multiple Objectives". In: *The 10th International Conference on Autonomous Agents and Multiagent Systems (AAMAS '11)*, pp. 371–378.

Federal Aviation Agency (2020). *UTM concept of operations version 2.0.* Technical Report. Federal Aviation Agency. URL: https://www.faa.gov/uas/research_development/traffic_management/media/UTM_ConOps_v2.pdf.

Fioretto, F., E. Pontelli, and W. Yeoh (2018). "Distributed Constraint Optimization Problems and Applications: A Survey". In: *Journal of Artificial Intelligence Research* 61, pp. 623–698.

References (cont.)

Garcia, Ander, Olatz Arbelaitz, Pieter Vansteenwegen, Wouter Souffriau, and Maria Teresa Linaza (2010). "Hybrid Approach for the Public Transportation Time Dependent Orienteering Problem with Time Windows". In: *Hybrid Artificial Intelligence Systems*. Ed. by Emilio Corchado, Manuel Graña Romay, and Alexandre Manhaes Savio. Berlin, Heidelberg: Springer Berlin Heidelberg, pp. 151–158. ISBN: 978-3-642-13803-4.

Golden, Bruce L, Larry Levy, and Rakesh Vohra (1987). "The orienteering problem". In: *Naval Research Logistics (NRL)* 34.3, pp. 307–318.

Grinshpoun, T., A. Grubshtein, R. Zivan, A. Netzer, and A. Meisels (2013). "Asymmetric Distributed Constraint Optimization Problems". In: *J. Artif. Int. Res.* 47.1, pp. 613–647. ISSN: 1076-9757. URL: `http://dl.acm.org/citation.cfm?id=2566972.2566988`.

Hamadi, Youssef (2020). "Optimization for Urban Air Mobility". In: *Learning and Intelligent Optimization*. Ed. by Ilias S. Kotsireas and Panos M. Pardalos. Cham: Springer International Publishing, pp. 1–8. ISBN: 978-3-030-53552-0.

Hamadi, Youssef and Gauthier Picard (2024). "Towards Socially-Acceptable Multi-Criteria Resolution of the 4D-Contracts Repair Problem". In: *International Conference on Autonomous Agents and Multiagent Systems (AAMAS-24)*. IFAAMAS.

Hirayama, K. and M. Yokoo (1997). "Distributed partial constraint satisfaction problem". In: *Principles and Practice of Constraint Programming-CP97*. Springer, pp. 222–236. ISBN: 978-3-540-69642-1.

Hoang, Khoi D., William Yeoh, Makoto Yokoo, and Zinovi Rabinovich (2020). "New Algorithms for Continuous Distributed Constraint Optimization Problems". In: *Proceedings of the 19th International Conference on Autonomous Agents and MultiAgent Systems*. AAMAS '20. Auckland, New Zealand: International Foundation for Autonomous Agents and Multiagent Systems, p. 502510. ISBN: 9781450375184.

Koenig, Sven, Craig A. Tovey, Michail G. Lagoudakis, Evangelos Markakis, David Kempe, Pinar Keskinocak, Anton J. Kleywegt, Adam Meyerson, and Sonal Jain (2006). "The Power of Sequential Single-Item Auctions for Agent Coordination". In: *Proceedings, The Twenty-First National Conference on Artificial Intelligence and the Eighteenth Innovative Applications of Artificial Intelligence Conference, July 16-20, 2006, Boston, Massachusetts, USA*. AAAI Press, pp. 1625–1629. URL: `http://www.aaai.org/Library/AAAI/2006/aaai06-266.php`.

References (cont.)

LAGOUDAKIS, Michail G., Evangelos MARKAKIS, David KEMPE, Pinar KESKINOCAK, Anton J. KLEYWEGT, Sven KOENIG, Craig A. TOVEY, Adam MEYERSON, and Sonal JAIN (2005). "Auction-Based Multi-Robot Routing". In: *Robotics: Science and Systems I, June 8-11, 2005, Massachusetts Institute of Technology, Cambridge, Massachusetts, USA*. Ed. by Sebastian THRUN, Gaurav S. SUKHATME, and Stefan SCHAAL. The MIT Press, pp. 343–350. DOI: 10.15607/RSS.2005.I.045. URL: http://www.roboticsproceedings.org/rss01/p45.html.

LÉAUTÉ, T. and B. FALTINGS (2011). "Distributed Constraint Optimization Under Stochastic Uncertainty". In: *Proceedings of the Twenty-Fifth AAAI Conference on Artificial Intelligence (AAAI'11)*. AAAI Press, pp. 68–73. URL: http://dl.acm.org/citation.cfm?id=2900423.2900434.

MAHESWARAN, R.T., J.P. PEARCE, and M. TAMBE (2004). "Distributed Algorithms for DCOP: A Graphical-Game-Based Approach". In: *Proceedings of the 17th International Conference on Parallel and Distributed Computing Systems (PDCS), San Francisco, CA*, pp. 432–439.

MATSUI, T., M. SILAGHI, K. HIRAYAMA, M. YOKOO, and H. MATSUO (2012). "Distributed Search Method with Bounded Cost Vectors on Multiple Objective DCOPs". In: *PRIMA 2012: Principles and Practice of Multi-Agent Systems*. Springer, pp. 137–152. ISBN: 978-3-642-32729-2.

MEDI, A., T. OKIMOTO, and K. INOUE (July 2014). "A two-phase complete algorithm for multi-objective distributed constraint optimization". In: 18, pp. 573–580.

MODI, P. J., W. SHEN, M. TAMBE, and M. YOKOO (2005). "ADOPT: Asynchronous Distributed Constraint Optimization with Quality Guarantees". In: *Artificial Intelligence* 161.2, pp. 149–180.

NGUYEN, D.T., W. YEOH, and H.C. LAU (2012). "Stochastic Dominance in Stochastic DCOPs for Risk-sensitive Applications". In: *Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems (AAMAS '12)*. Valencia, Spain, pp. 257–264. ISBN: 0-9817381-1-7, 978-0-9817381-1-6. URL: http://dl.acm.org/citation.cfm?id=2343576.2343613.

OKIMOTO, T., M. CLEMENT, and K. INOUE (2013). "AOF-Based Algorithm for Dynamic Multi-Objective Distributed Constraint Optimization". In: *Proceedings of the 7th International Workshop on Multi-disciplinary Trends in Artificial Intelligence (MIWAI'13)*. Springer, pp. 175–186. ISBN: 978-3-642-44948-2. DOI: 10.1007/978-3-642-44949-9_17. URL: http://dx.doi.org/10.1007/978-3-642-44949-9_17.

References (cont.)

Parsons, Simon, Juan A. Rodríguez-Aguilar, and Mark Klein (2011). “Auctions and bidding: A guide for computer scientists”. In: *ACM Comput. Surv.* 43.2, 10:1–10:59. doi: `10.1145/1883612.1883617`.

Petcu, A. and B. Faltings (2005). “Superstabilizing, Fault-containing Distributed Combinatorial Optimization”. In: *Proceedings of the 20th National Conference on Artificial Intelligence (AAAI’05)*. AAAI Press, pp. 449–454. isbn: 1-57735-236-x. url: `http://dl.acm.org/citation.cfm?id=1619332.1619405`.

Petcu, Adrian and Boi Faltings (2005). “A scalable method for multiagent constraint optimization”. In: *IJCAI International Joint Conference on Artificial Intelligence*, pp. 266–271. isbn: 1045-0823.

Phillips, Sean and Fernando Parra (2021). “A Case Study on Auction-Based Task Allocation Algorithms in Multi-Satellite Systems”. In: *AIAA Scitech 2021 Forum*. Reston, VA: AIAA. doi: `10.2514/6.2021-0185`.

Picard, Gauthier (2022a). “Auction-based and Distributed Optimization Approaches for Scheduling Observations in Satellite Constellations with Exclusive Orbit Portions”. In: *International Conference on Autonomous Agents and Multiagent Systems (AAMAS-22)*. IFAAMAS.

— (2022b). “Trajectory Coordination based on Distributed Constraint Optimization Techniques in Unmanned Air Traffic Management”. In: *International Conference on Autonomous Agents and Multiagent Systems (AAMAS-22)*. IFAAMAS, pp. 1065–1073. doi: `https://dl.acm.org/doi/10.5555/3535850.3535969`.

— (2023a). “Multi-Agent Consensus-based Bundle Allocation for Multi-Mode Composite Tasks”. In: *International Conference on Autonomous Agents and Multiagent Systems (AAMAS-23)*. IFAAMAS, pp. 504–512. doi: `10.5555/3545946.3598677`. url: `https://dl.acm.org/doi/10.5555/3545946.3598677`.

— (2023b). *Multi-agent multi-mode composite task allocation problem (MACTA) instances*. Version 1.0. Dataset. doi: `10.5281/zenodo.7550677`. url: `https://doi.org/10.5281/zenodo.7550677`.

References (cont.)

Pralet, Cédric (2025). "Continuous Planning and Execution for a Mission Planning System Managing a Constellation of Earth Observation Satellites". In: *International Workshop on Planning and Scheduling for Space (IWPSS-25)*.

Quinton, Félix, Christophe Grand, and Charles Lesire (Feb. 2023). "Market Approaches to the Multi-Robot Task Allocation Problem: a Survey". In: *J. Intell. Robotics Syst.* 107.2. issn: 0921-0296. doi: 10.1007/s10846-022-01803-0. url: https://doi.org/10.1007/s10846-022-01803-0.

Ramchurn, S. D., A. Farinelli, K. S. Macarthur, and N. R. Jennings (2010). "Decentralized Coordination in RoboCup Rescue". In: *Comput. J.* 53.9, pp. 1447–1461. issn: 0010-4620. doi: 10.1093/comjnl/bxq022. url: http://dx.doi.org/10.1093/comjnl/bxq022.

Sarker, Amit, Moumita Choudhury, and Md. Mosaddek Khan (2021). "A Local Search Based Approach to Solve Continuous DCOPs". In: *Proceedings of the 20th International Conference on Autonomous Agents and MultiAgent Systems*. AAMAS '21. Virtual Event, United Kingdom: International Foundation for Autonomous Agents and Multiagent Systems, p. 11271135. isbn: 9781450383073.

Schmid, Verena and Jan Fabian Ehmke (2017). "An Effective Large Neighborhood Search for the Team Orienteering Problem with Time Windows". In: *Computational Logistics*. Ed. by Tolga Bektaş, Stefano Coniglio, Antonio Martinez-Sykora, and Stefan Voss. Cham: Springer International Publishing, pp. 3–18. isbn: 978-3-319-68496-3.

SESAR (2019). *U-Space: Concept of Operations*. Tech. rep. SESAR Joint Undertaking.

Shelkamy, Mohamed, Catherine M. Elias, Dalia M. Mahfouz, and Omar M. Shehata (2020). "Comparative Analysis of Various Optimization Techniques for Solving Multi-Robot Task Allocation Problem". In: *2020 2nd Novel Intelligent and Leading Emerging Sciences Conference (NILES)*, pp. 538–543. doi: 10.1109/NILES50944.2020.9257967.

Shiroma, Pedro M. and Mario F. M. Campos (2009). "CoMutaR: A framework for multi-robot coordination and task allocation". In: *2009 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 4817–4824. doi: 10.1109/IROS.2009.5354166.

References (cont.)

STRANDERS, R., F.M. DELLE FAVE, A. ROGERS, and N.R. JENNINGS (May 2011). "U-GDL: A decentralised algorithm for DCOPs with Uncertainty". Project Report. URL: https://eprints.soton.ac.uk/273037/.

STRANDERS, R., A. FARINELLI, A. ROGERS, and N. R. JENNINGS (2009). "Decentralised Coordination of Continuously Valued Control Parameters Using the Max-Sum Algorithm". In: *Proceedings of The 8th International Conference on Autonomous Agents and Multiagent Systems - Volume 1*. AAMAS '09. Budapest, Hungary: International Foundation for Autonomous Agents and Multiagent Systems, p. 601608. ISBN: 9780981738161.

VERBEECK, Cédric, Pieter VANSTEENWEGEN, and El-Houssaine AGHEZZAF (July 2017). "The time-dependent orienteering problem with time windows: a fast ant colony system". In: *Annals of Operations Research* 254.1, pp. 481–505. DOI: 10.1007/s10479-017-2409-3.

VOICE, Thomas, Ruben STRANDERS, Alex ROGERS, and Nicholas R. JENNINGS (2010). "A Hybrid Continuous Max-Sum Algorithm for Decentralised Coordination". In: *Proceedings of the 2010 Conference on ECAI 2010: 19th European Conference on Artificial Intelligence*. NLD: IOS Press, p. 6166. ISBN: 9781607506058.

YEOH, W., P. VARAKANTHAM, X. SUN, and S. KOENIG (2011). "Incremental DCOP Search Algorithms for Solving Dynamic DCOPs". In: *The 10th International Conference on Autonomous Agents and Multiagent Systems (AAMAS '11)*, pp. 1069–1070. ISBN: 0-9826571-7-X, 978-0-9826571-7-1.

ZHANG, W., G. WANG, Z. XING, and L. WITTENBURG (2005). "Distributed stochastic search and distributed breakout: properties, comparison and applications to constraint optimization problems in sensor networks.". In: *Journal of Artificial Intelligence Research (JAIR)* 161.1-2, pp. 55–87.

ZHANG, Weixiong, Guandong WANG, Zhao XING, and Lars WITTENBURG (2003). "A Comparative Study of Distributed Constraint Algorithms". In: *Distributed Sensor Networks: A Multiagent Perspective*. Ed. by Victor LESSER, Charles L. ORTIZ, and Milind TAMBE. Boston, MA: Springer US, pp. 319–338.