From one to many: Multiagent Planning

Michal Štolba, Antonín Komenda, Carmel Domshlak, Raz Nissim

Multiagent systems

Overview

- systems consisting of agents
- an agent is a bounded entity
- the entities interact with each other
- generally no limitations on what an agent is (robots, humans, programs, ...)

Multiagent systems

Overview

- systems consisting of agents
- an agent is a bounded entity
- the entities interact with each other
- generally no limitations on what an agent is (robots, humans, programs, ...)

For scope of this lecture

- agents $\sim$ intelligent programs
- interaction $\sim$ message passing

Multiagent systems

Technically

- agents $\sim$ a computational thread or process running ideally on its own machine or processor (core)
- interaction $\sim$ inter-process sending of messages (potentially over network)

Agent taxonomy

Reactive agent

Reacts only on immediate stimuli.

Agent taxonomy

Reactive agent

Reacts only on immediate stimuli.

Proactive agent

Pro actively tries to satisfy its needs and goals

Agent taxonomy

Reactive agent

Reacts only on immediate stimuli.

Proactive agent

Pro actively tries to satisfy its needs and goals

Planning agent

Plans its actions in advance.

Planning in MAS

How an agent sees other agents?

- Adversary - Game Theory
- As part of environment (indifferent) - POMDP
- Cooperative - This lecture

Planning in MAS

How an agent sees other agents?

- Adversary - Game Theory
- As part of environment (indifferent) - POMDP
- Cooperative - This lecture

Expressivity

- Concurrent actions (yes)
- Joint actions (no)
  - Combinatorial explosion
- Time, resources, uncertainty, ... (not now)

Multi-Agent Planning Problem

MAP

$\mathcal{M} = \langle \mathcal{A}, \{\Pi_i\}_{i=1}^{n} \rangle$

- $\mathcal{A}$ - set of n agents
- $\{\Pi_i\}_{i=1}^{n}$ set of n agent planning problems

Multi-Agent Planning Problem

Agent Problem

$\Pi_i = \langle V_i, O_i, s_I^i, s_\star \rangle$

- $V_i = V^{\text{pub}} \cup V_i^{\text{priv}}$ - set of public and private variables
- $O_i = O_i^{\text{pub}} \cup O_i^{\text{priv}}$ - set of public and private operators
- s_I^i - initial state of agent i (full valuation of V_i)
- $s_\star$ - public (WLOG) goal state (partial valuation over V^{pub})

Multi-Agent Planning Problem

Agent Problem

$$\Pi_i = \left\langle V_i = V^{\text{pub}} \cup V_i^{\text{priv}}, O_i = O_i^{\text{pub}} \cup O_i^{\text{priv}}, s_I^i, s_\star \right\rangle$$

Properties

- V^{pub} - public variables shared among all agents
- $V^{\text{pub}} \cap V_i^{\text{priv}} = \emptyset$, $V_i^{\text{priv}} \cap V_j^{\text{priv}} = \emptyset$, $O_i^{\text{pub}} \cap O_i^{\text{priv}} = \emptyset$, $O_i \cap O_j = \emptyset$ for all i, j
- Operator $o \in O_i$ is public iff precondition or effect defined over a public variable

Multi-Agent Planning Problem

Solution

$$
\begin{array}{lclcccccccl}
\pi_1 & = & (& o_1, & \epsilon, & \epsilon, & \ldots, & o_2, & o_3 &) & o_1, o_2, o_3 \in O_1 \\
 & \ldots & & & & & & & & & \\
\pi_n & = & (& \epsilon, & o_4, & \epsilon, & \ldots, & o_5, & \epsilon &) & o_4, o_5 \in O_n
\end{array}
$$

Multi-Agent Planning Problem

Solution

$$\begin{array}{lllllllllll} \pi_1 & = & (& o_1, & \epsilon, & \epsilon, & \ldots, & o_2, & o_3 &) & o_1, o_2, o_3 \in O_1 \\ & \ldots & & & & & & & & & \\ \pi_n & = & (& \epsilon, & o_4, & \epsilon, & \ldots, & o_5, & \epsilon &) & o_4, o_5 \in O_n \end{array}$$

Properties

- ϵ is a no-op action (empty precondition and effect), i.e., a placeholder
- The actions in each time-step **must not** be **mutually exclusive** (mutex)
- The actions in subsequent time-steps follow the same rules as in classical planning

Projections

Global problem

$\Pi^{G} = \langle V = \bigcup_{i=1}^{n} V_i, O = \bigcup_{i=1}^{n} O_i, s_I, s_\star \rangle$

- All variables
- All operators
- s_I - global initial state (valuation over V)
- $s_\star$ - public goal state

Projections

Global problem

$\Pi^G = \langle V = \bigcup_{i=1}^{n} V_i, O = \bigcup_{i=1}^{n} O_i, s_I, s_\star \rangle$

- All variables
- All operators
- s_I - global initial state (valuation over V)
- $s_\star$ - public goal state

Properties

- Equal to the MAP problem $\mathcal{M}$.
- Solution to one corresponds to a solution of the other.

Projections

Why we cannot simply solve Π^G centrally?

- Cost of communication
- Cost of formalization
- Speed improvement (factorization)
- Privacy

Projections

Public projection of operator

For an operator $o \in O_i^{\mathsf{pub}}$, public projection is operator $o^{\triangleright}$ such that

- Precondition of o is restricted to V^{pub}
- Effect of o is restricted to V^{pub}

Projections

Public projection of operator

For an operator $o \in O_i^{\mathrm{pub}}$, public projection is operator $o^{\triangleright}$ such that

- Precondition of o is restricted to V^{pub}
- Effect of o is restricted to V^{pub}

Public projection

$\Pi^{\triangleright} = \left\langle V^{\mathrm{pub}}, O^{\triangleright}, s_I^{\triangleright}, s_{\star} \right\rangle$

- $O^{\triangleright} = \{o^{\triangleright} | o \in \bigcup_{i=1}^{n} O_i^{\mathrm{pub}}\}$
- $s_I^{\triangleright}$ is s_I restricted to V^{pub} (valuation over V^{pub})

Projections

Agent projection for agent i

$\Pi^{\triangleright i} = \langle V_i, O^{\triangleright i}, s_I^i, s_\star \rangle$

- $O^{\triangleright i} = O_i \cup \{o^{\triangleright} | o \in \bigcup_{j=1}^{n} O_j^{\text{pub}}, j \neq i\}$

Projections

Properties

- Agent problem is **not an abstraction**[a] of the global problem
- Agent projection is **an abstraction** of the global problem
- Public projection is **an abstraction** of the global problem

[a] see lecture 4, abstractions

What is privacy?

Agents do not want to share their private information

- The set V_i^{priv} of private variables and their domains
- The set O_i^{priv} of private actions (their pre, eff, cost)
- The private preconditions and effects of all $o \in O_i^{\text{pub}}$

What is privacy?

Agents do not want to share their private information

- The set V_i^{priv} of private variables and their domains
- The set O_i^{priv} of private actions (their pre, eff, cost)
- The private preconditions and effects of all $o \in O_i^{\text{pub}}$

Agents share

- The set V^{pub} of public variables and their domains
- The set $O^{\triangleright}$ of public projections of public actions (i.e., public interface of the agent)
- Public projection of the final plan

Privacy

Weak privacy

Do not openly share private formation (i.e., unecrypted).

Privacy

Weak privacy

Do not openly share private formation (i.e., unecrypted).

But!

Private information can be inferred from the public information and the execution of the algorithm (e.g., the messages passed)

Privacy

Strong privacy

No private information can be inferred from the public information and the execution of the algorithm.

Privacy

Strong privacy

No private information can be inferred from the public information and the execution of the algorithm.

Assumptions

- **Computational Strong Privacy:** Using a limited resources (typically P)
- **Unconditional Strong Privacy:** No limits (even harder!)
- **Semi-honest agents:** Do not alter the computation protocol
- **Malicious agents:** Can do anything (harder!)

Privacy

But!

- Strong privacy is hard to achieve!
- Secure protocols exist for some problems
  - Set intersection, secure sum, linear programming, ...
- General Secure Multiparty Computation methods
  - Cryptographic circuits [Yao, 1986]
  - Typically exponential in the input size (the transition system!)

MAP Plannig

Planning paradigms

- Plan-based coordination
- State-based coordination

General idea of plan-based coordination

Extensibility

Let $\pi^{\triangleright} = (o_1^{\triangleright}, \ldots, o_k^{\triangleright})$ s.t. $o_1, \ldots, o_k \in O^{\triangleright}$ be a public plan (i.e., solution to the public projection).

- We say that $\pi^{\triangleright}$ is extensible by agent i
  - if by inserting some $o_i, \ldots, o_j \in O_i^{\text{priv}}$ into the respective $\pi = (o_1, \ldots, o_k)$
  - we obtain π' which is a solution to the agent projection $\Pi^{\triangleright i}$.

General idea of plan-based coordination

Extensibility

Let $\pi^\triangleright = (o_1^\triangleright, ..., o_k^\triangleright)$ s.t. $o_1, ..., o_k \in O^\triangleright$ be a public plan (i.e., solution to the public projection).

- We say that $\pi^\triangleright$ is extensible by agent i
  - if by inserting some $o_i, ..., o_j \in O_i^{\text{priv}}$ into the respective $\pi = (o_1, ..., o_k)$
  - we obtain π' which is a solution to the agent projection $\Pi^{\triangleright i}$.

Theorem [Tožička et al., 2015]

Let $\pi^\triangleright$ be a public plan. If $\pi^\triangleright$ is extensible by all agents in $\mathcal{A}$ then $\pi^\triangleright$ is a public projection of a solution of $\mathcal{M}$.

General idea of plan-based coordination

1. Find a public plan extensible by all agents.
2. Find the respective extending plan for each agent.

Planning First
[Nissim&Brafman, 2010]

Use Distributed CSP to find extensible public plan

- DisCSP: One variable per agent
- Values: All possible sequences of public actions from $O^{\triangleright}$ and placeholders for private actions
- Constraints:
  - Binary: The sequences must match on public actions (coordination points)
  - Unary: The value (public plan) must be extensible by the respective agent

Planning First
[Nissim&Brafman, 2010]

Properties

- Trivial extension
- Known complexity results [Brafman&Domshlak, 2008]
  - Not exponentially dependent on number of agents

PSM

[Tožička, Jakubův, Komenda & Pěchouček, 2015]

PSM Planner

- Each agent i:
  1. Generate plans for $\Pi^{\triangleright i}$
  2. Create public projections
  3. Share and find intersection
- If intersection nonempty, solution found, else iterate

PSM
[Tožička, Jakubův, Komenda & Pěchouček, 2015]

Properties

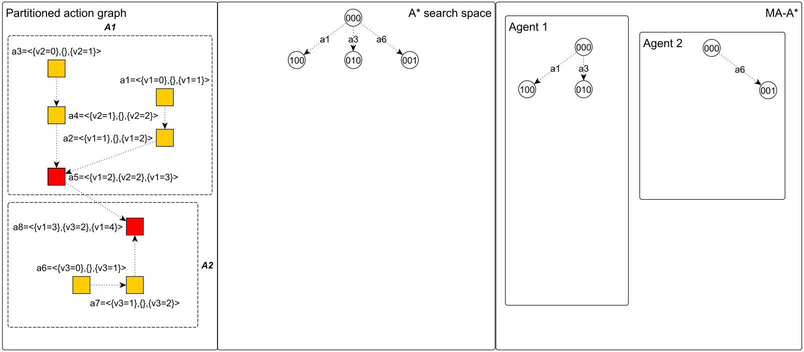
- Use FSM-like compact representation of plans (Planning State Machine)
- Strong privacy preserving if
  - using secure set (or FSM) intersection [Pinkas, 2015] and
  - only one iteration [Tožička, Štolba & Komenda 2017]
- Winner of the last MAP competition (2015)

Multi-Agent Forward Search
[Nissim&Brafman, 2014]

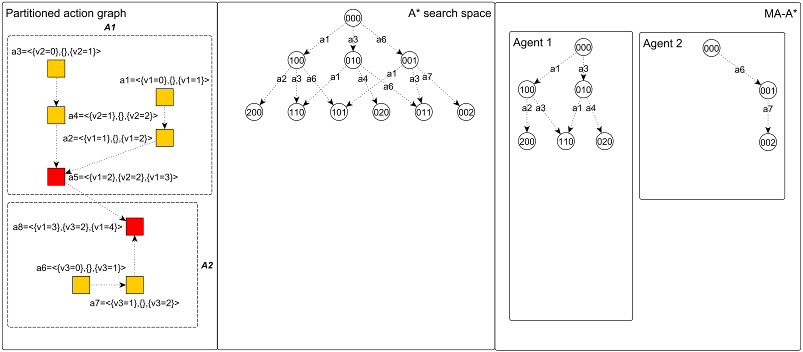
MAFS Principle

- Separate search spaces
- Each agent search using operators in O_i (no projections)
- If a state s was reached by a public operator $o \in O_i^{pub}$, send s to all other agents
  - Encrypt private part of the state
  - Enough to send only to agents which have applicable public operator

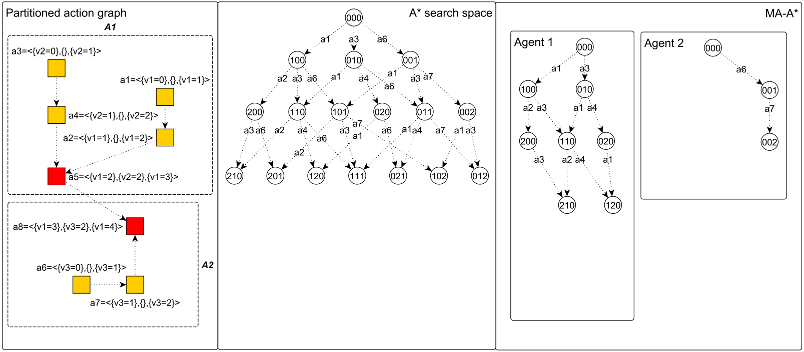
Multi-Agent Forward Search
[Nissim&Brafman, 2014]



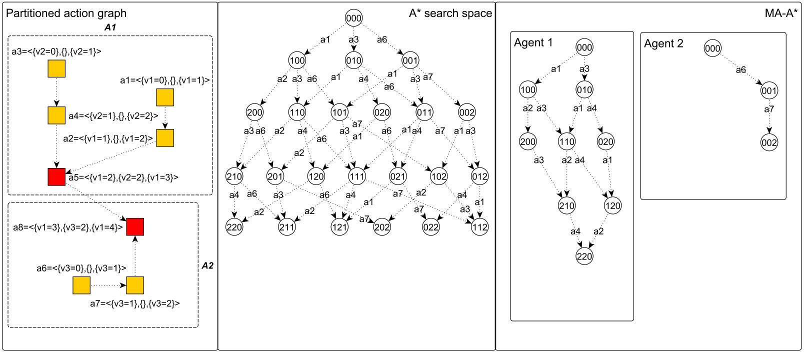
Multi-Agent Forward Search
[Nissim&Brafman, 2014]



Multi-Agent Forward Search
[Nissim&Brafman, 2014]

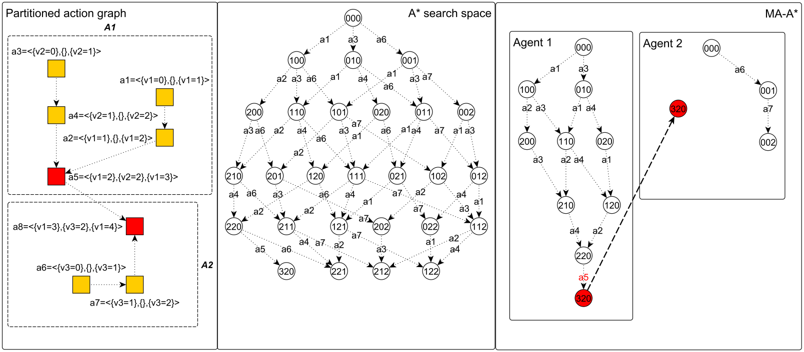


Multi-Agent Forward Search
[Nissim&Brafman, 2014]

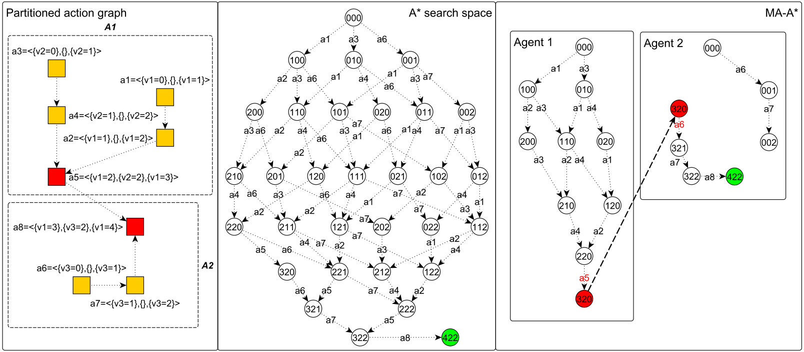


Multi-Agent Forward Search

[Nissim&Brafman, 2014]



Multi-Agent Forward Search
[Nissim&Brafman, 2014]



MAD-A*
[Nissim&Brafman, 2014]

Multi-agent distributed A*

- How to ensure global optimality of solution?
  - Better state might be in other agent's open list
  - We need to check a global state of the distributed system
  - A state might even be currently being send (that is, in no open list)
- Distributed snapshot algorithms [Chandy&Lamport, 1985]
  - Uses start and end message markers to denote states sent in between
  - Thus makes sure states currently in communication channels are also considered (exactly once)

Secure-MAFS
[Brafman, 2015; Maliah, Shani & Brafman, 2016]

How to reduce private information leakage?

- Send less states but maintain completeness.

Secure-MAFS
[Brafman, 2015; Maliah, Shani & Brafman, 2016]

How to reduce private information leakage?

- Send less states but maintain completeness.

i-parent

State s:

public part	private part of agent 1	...	private part of agent n
V^{pub}	V_1	...	V_n

- **parent:** State s_j in the open list of j
- i-**parent:** State s_i sent from i to j such that

Secure-MAFS

[Brafman, 2015; Maliah, Shani & Brafman, 2016]

Macro-MAFS

State s:

public part	private part of agent 1	...	private part of agent n
V^{pub}	V_1	...	V_n

- When i receives state s from j
  - i-parent s_i
  - Record macro operator[a] $o_{j,s_i} : s_i \to s$
- When a state s' equals with s_i on the public par and part of agent j
  - Use the macro operator o_{j,s_i} instead of sending s'

[a] See Lecture 8, modeling and reformulation

Heuristics in MAP

Projected heuristic

Computed on the agent projected problem and estimating only the agents problem solution cost.

Heuristics in MAP

Projected heuristic

Computed on the agent projected problem and estimating only the agents problem solution cost.

Distributed heuristic

Computed in a distributed way and estimating the global problem solution cost.

Projected heuristic

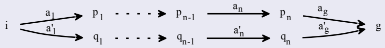
Properties

- Maintains admissibility
  - Hint: Agent projection is an abstraction.
  - Taking $\max(h_i(s), h_j(s))$ is admissible.
- Strong privacy preserving
  - No information exchanged
  - Depends on how the heuristic is used (heuristic value itself might leak information)!

Projected heuristic

Properties

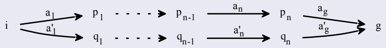
- Maintains admissibility
  - Hint: Agent projection is an abstraction.
  - Taking $\max(h_i(s), h_j(s))$ is admissible.
- Strong privacy preserving
  - No information exchanged
  - Depends on how the heuristic is used (heuristic value itself might leak information)!
- May be arbitrarily bad

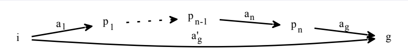


Projected heuristic

Properties

- Maintains admissibility
  - Hint: Agent projection is an abstraction.
  - Taking $\max(h_i(s), h_j(s))$ is admissible.
- Strong privacy preserving
  - No information exchanged
  - Depends on how the heuristic is used (heuristic value itself might leak information)!
- May be arbitrarily bad





Distributed heuristic

Properties

- Better informed
- Admissibility!
- Privacy!

Distributed heuristic

Properties

- Better informed
- Admissibility!
- Privacy!

Examples

- DTG-based [Torreno, Onaindia & Sapena, 2014]
- LM-Cut [Štolba, Fišer & Komenda, 2015]
- Potential heuristics [Štolba, Fišer & Komenda, 2016]

Distributed potential heuristic

Potential heuristic

$$h_{\mathrm{pot}}(s) = \sum_{v \in V} \mathrm{pot}(\langle v, s[v] \rangle)$$

Distributed potential heuristic

Potential heuristic

$$h_{\text{pot}}(s) = \sum_{v \in V} \text{pot}(\langle v, s[v] \rangle)$$

Potential heuristic in MAP

$$h_{\text{pot}}^{\text{pub}}(s) = \sum_{v \in V^{\textbf{pub}}} \text{pot}(\langle v, s[v] \rangle)$$

$$h_{\text{pot}}^{\text{priv}_i}(s) = \sum_{v \in V_i^{\textbf{priv}}} \text{pot}(\langle v, s[v] \rangle)$$

Distributed potential heuristic

Potential heuristic

$$h_{\text{pot}}(s) = \sum_{v \in V} \text{pot}(\langle v, s[v] \rangle)$$

Potential heuristic in MAP

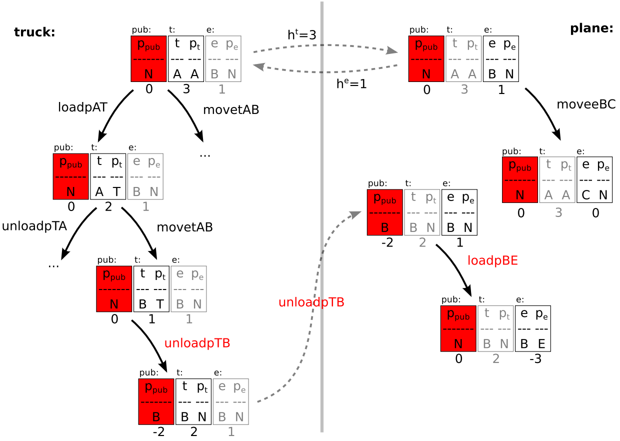
$$h^{\text{pub}}_{\text{pot}}(s) = \sum_{v \in V^{\textbf{pub}}} \text{pot}(\langle v, s[v] \rangle)$$

$$h^{\text{priv}_i}_{\text{pot}}(s) = \sum_{v \in V^{\textbf{priv}}_i} \text{pot}(\langle v, s[v] \rangle)$$

MA potential heuristic

$$h^{\text{G}}_{\text{pot}}(s) = h^{\text{pub}}_{\text{pot}}(s) + \sum_{i \in \mathcal{A}} h^{\text{priv}_i}_{\text{pot}}(s)$$

Distributed potential heuristic



Distributed potential heuristic

Properties

- No additional communication
  - Private parts $h_{\text{pot}}^{\text{priv}_i}(s)$ of other agents $i \in \mathcal{A} \setminus \{j\}$ are not changed by agent j
- Privacy preserving (not completely strong)
  - Secure sum [Sheikh, Kumar & Mishra, 2010]
  - Secure LP computation [Mangasarian, 2011; Dreier&Kerschbaum, 2011]

Distributed potential heuristic

Secure LP computation principle

- LP solved by one of the agents (e.g. i)
  1. All $j \neq i$ encrypt their parts of the LP
  2. All $j \neq i$ send their encrypted parts to i
  3. i combines the encrypted parts and computes the LP
  4. i sends results to all other agents $j \neq i$
  5. Agents $j \neq i$ decrypt their parts of the solution (potentials)