Multi-Agent Planning

Michal Štolba

stolba@agents.fel.cvut.cz

AI CENTER
Department of Computer Science, CTU

PUI (Planning in Artificial Intelligence)

Coordination Schemes

Planning By	Planning For: Single Agent	Planning For: Multiple Agents
Multiple Agents	Distributed Planning	**Multi-Agent Planning**
Single Agent	Classical Planning	Classical Planning

Agents and Environment

Observ-ability	Actions	No Agents	Cooperative Agents	Adversarial Agents
Partial	Nondet.	POMDP	Dec-POMDP	POSG
	Det.	Conformant Planning		
Privacy	Nondet.	-	?	?
	Det.	-	**MA-STRIPS**	?
Full	Nondet.	MDP, Contingent Planning, Fault-tolerant Planning	MMDP, Factored MDP	Stochastic games
	Det.	**Classical/STRIPS**	Factored Planning	Perfect Information Games

Idea

MA-STRIPS

- Agents $\mathscr{A}$, $|\mathscr{A}| = n$
- Planning problem for each agent
  - $\{\Pi_i\}_{i=1}^{n}$
  - $\Pi_i = \langle P_i, A_i, I_i, G_i \rangle$

Idea

MA-STRIPS

- $\Pi_i = \langle P_i, A_i, I_i, G_i \rangle$
  - $P_i = P_i^{\text{priv}} \cup P^{\text{pub}}$
  - $A_i = A_i^{\text{priv}} \cup A_i^{\text{pub}}$ + projections
  - $I_i = I \cap P_i$
  - $G_i \subseteq P^{\text{pub}}$

Idea

MA-STRIPS

- $\Pi_i = \langle P_i, A_i, I_i, G_i \rangle$
- Action $a \in A_i$ is public if either
  - $\text{pre}(a) \cap P^{\text{pub}} \neq \emptyset$,
  - $\text{add}(a) \cap P^{\text{pub}} \neq \emptyset$, or
  - $\text{del}(a) \cap P^{\text{pub}} \neq \emptyset$

Projection

MA-STRIPS

- Public projection of action
  - $a \in A_i^{\text{pub}}$: $a = \langle \text{pre}(a), \text{add}(a), \text{del}(a) \rangle$
  - $a^{\triangleright} = \langle \text{pre}(a), \text{add}(a), \text{del}(a) \rangle$
    - $\text{pre}(a^{\triangleright}) = \text{pre}(a) \cap P^{\text{pub}}$
    - $\text{add}(a^{\triangleright}) = \text{add}(a) \cap P^{\text{pub}}$
    - $\text{del}(a^{\triangleright}) = \text{del}(a) \cap P^{\text{pub}}$
- *i*-projection of state
  - $s \subseteq \bigcap_{i=1}^{n} P_i$ ... $s^{\triangleright i} = s \cap P_i$

MA-MPT

- Similar to MA-STRIPS
- Private or public variables

Multi-Agent Forward Search

Principle

- MAD-A* instance of MAFS
- Each agent searches its own search space (no projections)
  - Asynchronous!
- Send states achieved by public actions
  - Encrypt private information
- Add received states to the open list

Multi-Agent Forward Search

Principle

- MAD-A* instance of MAFS
- Each agent searches its own search space (no projections)
  - Asynchronous!
- Send states achieved by public actions
  - Encrypt private information
- Add received states to the open list

Multi-Agent Forward Search

Principle

- MAD-A* instance of MAFS
- Each agent searches its own search space (no projections)
  - Asynchronous!
- Send states achieved by public actions
  - Encrypt private information
- Add received states to the open list

Multi-Agent Forward Search

Principle

- MAD-A* instance of MAFS
- Each agent searches its own search space (no projections)
  - Asynchronous!
- Send states achieved by public actions
  - Encrypt private information
- Add received states to the open list

Multi-Agent Forward Search

Heuristics

- Projected
  - Compute on Π_i (including projected actions)
  - Send with states
  - Take maximum $h_i(s)$ and $h_j(s)$ when s received by i from j

    + Fast, computed individually
    - Less informed

Multi-Agent Forward Search

Heuristics

- Projected
  - Compute on Π_i (including projected actions)
  - Send with states
  - Take maximum $h_i(s)$ and $h_j(s)$ when s received by i from j

+ Fast, computed individually
- Less informed

Multi-Agent Forward Search

Heuristics

- Distributed
  - Compute by all agents for each state
  - Relaxations/FF, LM-Cut, Potential heuristics

    + More informed
    - Slow, all agents must participate

Multi-Agent Forward Search

Heuristics

- Distributed
  - Compute by all agents for each state
  - Relaxations/FF, LM-Cut, Potential heuristics

  \+ More informed

  \- Slow, all agents must participate

Multi-Agent Forward Search

Heuristics

- Lazy-FF
  - i computes RP
  - Requests other agents for RP to solve private preconditions
- MA-Pot
  - Distributed LP computation
  - Potentials for P^{pub} and P_i^{priv} for each agent
  - Sent with the state and summed-up independently

Multi-Agent Forward Search

Heuristics

- Lazy-FF
  - i computes RP
  - Requests other agents for RP to solve private preconditions
- MA-Pot
  - Distributed LP computation
  - Potentials for P^{pub} and P_i^{priv} for each agent
  - Sent with the state and summed-up independently

Planning State Machine Planner

Idea

1. Each agent generates a set of plans $S_i = \{\pi_1^i, ..., \pi_k^i\}$
2. Public projection: $S_i^{\triangleright} = \{\pi_1^{i\triangleright}, ..., \pi_k^{i\triangleright}\}$
   - $\pi^{\triangleright}$ ... public actions replaced with projections, private actions removed
3. Find intersection: $\bigcup_{i=1}^{n} S_i^{\triangleright}$
   - if $\pi_l^{\triangleright} \in \bigcap_{i=1}^{n} S_i^{\triangleright}$ ... $\{\pi_l^1, ..., \pi_l^n\}$ is a plan
   - if $\bigcap_{i=1}^{n} S_i^{\triangleright} = \emptyset$ .. no solution, add more plans

(Note: Systematic generation necessary to avoid complete plan-space exploration)

Planning State Machine Planner

Idea

1. Each agent generates a set of plans $S_i = \{\pi_1^i, ..., \pi_k^i\}$
2. Public projection: $S_i^{\triangleright} = \{\pi_1^{i\triangleright}, ..., \pi_k^{i\triangleright}\}$
   - $\pi^{\triangleright}$ ... public actions replaced with projections, private actions removed
3. Find intersection: $\bigcup_{i=1}^{n} S_i^{\triangleright}$
   - if $\pi_l^{\triangleright} \in \bigcap_{i=1}^{n} S_i^{\triangleright}$ ... $\{\pi_l^1, ..., \pi_l^n\}$ is a plan
   - if $\bigcap_{i=1}^{n} S_i^{\triangleright} = \emptyset$ .. no solution, add more plans

(Note: Systematic generation necessary to avoid complete plan-space exploration)

Planning State Machine Planner

Idea

1. Each agent generates a set of plans $S_i = \{\pi^i_1, ..., \pi^i_k\}$
2. Public projection: $S_i^{\triangleright} = \{\pi_1^{i\triangleright}, ..., \pi_k^{i\triangleright}\}$
   - $\pi^{\triangleright}$ ... public actions replaced with projections, private actions removed
3. Find intersection: $\bigcup_{i=1}^{n} S_i^{\triangleright}$
   - if $\pi_l^{\triangleright} \in \bigcap_{i=1}^{n} S_i^{\triangleright}$ ... $\{\pi_l^1, ..., \pi_l^n\}$ is a plan
   - if $\bigcap_{i=1}^{n} S_i^{\triangleright} = \emptyset$ .. no solution, add more plans

(Note: Systematic generation necessary to avoid complete plan-space exploration)

Planning State Machine Planner

Idea

?

What about infinite number of plans (loops!)?

Planning State Machine Planner

Idea

- Planning State Machine (PSM)
  - Concise representation of (infinite) number of plans
  - Based on Finite Automata
  - Projection

What is MAP good for?

- Factorization
  - Solve more but smaller problems
  - How to factor the problem?
- Distributed/Parallelized computation
  - Search notoriously hard to parallelize
  - MAP-A* - better than generic techniques
- Privacy
  - The reason MAP cannot be solved centrally
  - What is that?

What is MAP good for?

- Factorization
  - Solve more but smaller problems
  - How to factor the problem?
- Distributed/Parallelized computation
  - Search notoriously hard to parallelize
  - MAP-A* - better than generic techniques
- Privacy
  - The reason MAP cannot be solved centrally
  - What is that?

What is MAP good for?

- Factorization
  - Solve more but smaller problems
  - How to factor the problem?
- Distributed/Parallelized computation
  - Search notoriously hard to parallelize
  - MAP-A* - better than generic techniques
- Privacy
  - The reason MAP cannot be solved centrally
  - What is that?

Privacy

Motivation

- Business cooperation/consortium
  - Need to cooperate but do not want to disclose data and processes
- Sensitive data
  - Medical computations
  - Private data on the cloud
- Military coalition operations
  - Need to cooperate but some data secret

Privacy

In Computer Science

Secure Multiparty Computation

- Secure multiparty computation (MPC) (Yao 1982)
- Subfield of cryptography
- Compute a function f by a set of n parties $p_1, ..., p_n$ such that each p_i knows part of the input of f.
- Compute f in a way that no party p_i learns more information about the inputs of other parties than what can be learned from the output of f.

Secure Multiparty Computation

Assumptions

- Other agents

  Semi-honest Attempts to get as much information as possible, but does not alter the protocol.

  Malicious Can do whatever it wants to deceive and get information.

- Computation

  Information-theoretic privacy no assumptions on computation power of agents.

  Computational privacy polynomial bound $\rightarrow$ factoring is hard, etc.

- Communication
  - Synchronous/Asynchronous
  - Retains order of messages (or not)
  - ...

Secure Multiparty Computation

Assumptions

- Other agents
  - Semi-honest: Attempts to get as much information as possible, but does not alter the protocol.
  - Malicious: Can do whatever it wants to deceive and get information.
- Computation
  - Information-theoretic privacy: no assumptions on computation power of agents.
  - Computational privacy: polynomial bound $\rightarrow$ factoring is hard, etc.
- Communication
  - Synchronous/Asynchronous
  - Retains order of messages (or not)
  - ...

Secure Multiparty Computation

Assumptions

- Other agents
  - **Semi-honest** Attempts to get as much information as possible, but does not alter the protocol.
  - **Malicious** Can do whatever it wants to deceive and get information.
- Computation
  - **Information-theoretic privacy** no assumptions on computation power of agents.
  - **Computational privacy** polynomial bound $\rightarrow$ factoring is hard, etc.
- Communication
  - Synchronous/Asynchronous
  - Retains order of messages (or not)
  - ...

Privacy

In Multi-Agent Planning

- What is the private information?
  - Existence and value(s) of private fact (or variable)
  - Existence of private actions
  - For a public action $a \in A_i^{\text{pub}}$, existence and value(s) of
    - private $\text{pre}(a) \cap P_i^{\text{priv}}$
    - private $\text{add}(a) \cap P_i^{\text{priv}}$
    - private $\text{del}(a) \cap P_i^{\text{priv}}$

Privacy-preserving Planner

!

Simply not sending private information is not enough!

- Private information may leak (be deduced)
  - Action is not applicable but the projection is ...
  - Heuristic values ...

Privacy-preserving Planner

!

Simply not sending private information is not enough!

- Private information may leak (be deduced)
  - Action is not applicable but the projection is ...
  - Heuristic values ...

Privacy-preserving Planner

Recall PSM - where might private information leak?

1. Each agent generates a set of plans $S_i = \{\pi_1^i, ..., \pi_k^i\}$
2. Public projection: $S_i^{\triangleright} = \{\pi_1^{i\triangleright}, ..., \pi_k^{i\triangleright}\}$
   - $\pi^{\triangleright}$ ... public actions replaced with projections, private actions removed
3. Find intersection: $\bigcap_{i=1}^{n} S_i^{\triangleright}$
   - if $\pi_l^{\triangleright} \in \bigcap_{i=1}^{n} S_i^{\triangleright}$ ... $\{\pi_l^1, ..., \pi_l^n\}$ is a plan
   - if $\bigcap_{i=1}^{n} S_i^{\triangleright} = \emptyset$ .. no solution, add more plans

Privacy-preserving Planner

Recall PSM - where might private information leak?

1. Each agent generates a set of plans $S_i = \{\pi^i_1, ..., \pi^i_k\}$
2. Public projection: $S_i^{\triangleright} = \{\pi_1^{i\triangleright}, ..., \pi_k^{i\triangleright}\}$
   - $\pi^{\triangleright}$ ... public actions replaced with projections, private actions removed
3. Find intersection: $\bigcap_{i=1}^{n} S_i^{\triangleright}$
   - if $\pi_l^{\triangleright} \in \bigcap_{i=1}^{n} S_i^{\triangleright}$ ... $\{\pi_l^1, ..., \pi_l^n\}$ is a plan
   - if $\bigcap_{i=1}^{n} S_i^{\triangleright} = \emptyset$ .. no solution, add more plans

Privacy-preserving Planner

Recall PSM - where might private information leak?

1. Each agent generates a set of plans $S_i = \{\pi^i_1, ..., \pi^i_k\}$
2. Public projection: $S^{\triangleright}_i = \{\pi^{i\triangleright}_1, ..., \pi^{i\triangleright}_k\}$
   - $\pi^{\triangleright}$ ... public actions replaced with projections, private actions removed
3. Find intersection: $\bigcap_{i=1}^{n} S^{\triangleright}_i$
   - if $\pi^{\triangleright}_l \in \bigcap_{i=1}^{n} S^{\triangleright}_i$ ... $\{\pi^1_l, ..., \pi^n_l\}$ is a plan
   - if $\bigcap_{i=1}^{n} S^{\triangleright}_i = \emptyset$ .. no solution, add more plans

Privacy-preserving Planner

PSM - solution

- Find intersection: $\bigcap_{i=1}^{n} S_i^{\triangleright}$ securely!
  - Information-theoretic secure set intersection (Li&Wu 2007)
  - Computationally secure DFA intersection (Guanciale et al. 2014)
- (+ Securely select a solution at random)

Privacy-preserving Planner

PSM - solution?

- But! What if no solution found?
- Recall: “If no solution, add more plans”
- Information leaks!
  - Assuming some systematic generation of plans (e.g. from shortest to longest)
  - In iteration k all plans of length $< k$ already generated by all agents
  - If not accepted - some private preconditions must exist

Privacy-preserving Planner

PSM - solution?

- But! What if no solution found?
- Recall: “If no solution, add more plans”
- Information leaks!
  - Assuming some systematic generation of plans (e.g. from shortest to longest)
  - In iteration k all plans of length $< k$ already generated by all agents
  - If not accepted - some private preconditions must exist

Privacy-preserving Planner

PSM - solution?

- But! What if no solution found?
- Recall: “If no solution, add more plans”
- Information leaks!
  - Assuming some systematic generation of plans (e.g. from shortest to longest)
  - In iteration k all plans of length $< k$ already generated by all agents
  - If not accepted - some private preconditions must exist

Privacy-preserving Planner

PSM - so can it be privacy-preserving?

Generate all plans up-front - not efficient!

Do just one iteration (with random plans) - not complete!

Iterate systematically - information leaks!

- Is the non-completeness practically a problem?
  - **Research question!**
- Does it always hold?
  - **Research question!**
- How to quantify leaked information (in general)
  - **Research question!**

Privacy-preserving Planner

PSM - so can it be privacy-preserving?

Generate all plans up-front - not efficient!

Do just one iteration (with random plans) - not complete!

Iterate systematically - information leaks!

- Is the non-completeness practically a problem?
  - **Research question!**
- Does it always hold?
  - **Research question!**
- How to quantify leaked information (in general)
  - **Research question!**

Privacy-preserving Planner

PSM - so can it be privacy-preserving?

Generate all plans up-front - not efficient!

Do just one iteration (with random plans) - not complete!

Iterate systematically - information leaks!

- Is the non-completeness practically a problem?
  - **Research question!**
- Does it always hold?
  - **Research question!**
- How to quantify leaked information (in general)
  - **Research question!**

Privacy-preserving Planner

PSM - so can it be privacy-preserving?

Generate all plans up-front - not efficient!

Do just one iteration (with random plans) - not complete!

Iterate systematically - information leaks!

- Is the non-completeness practically a problem?
  - **Research question!**
- Does it always hold?
  - **Research question!**
- How to quantify leaked information (in general)
  - **Research question!**

Privacy-preserving Planner

PSM - so can it be privacy-preserving?

Generate all plans up-front - not efficient!

Do just one iteration (with random plans) - not complete!

Iterate systematically - information leaks!

- Is the non-completeness practically a problem?
  - **Research question!**
- Does it always hold?
  - **Research question!**
- How to quantify leaked information (in general)
  - **Research question!**

Privacy-preserving Planner

PSM - so can it be privacy-preserving?

Generate all plans up-front - not efficient!

Do just one iteration (with random plans) - not complete!

Iterate systematically - information leaks!

- Is the non-completeness practically a problem?
  - **Research question!**
- Does it always hold?
  - **Research question!**
- How to quantify leaked information (in general)
  - **Research question!**