Scaling Up Multi-Agent Planning - A Best-Response Approach

Anders Jonsson, Universitat Pompeu Fabra
Michael Rovatsos, University of Edinburgh

Motivation

- Planning with multiple agents is hard
  - Joint action space is exponential in the number of agents
  - Agents may be self-interested

Motivation

- Planning with multiple agents is hard
  - Joint action space is exponential in the number of agents
  - Agents may be self-interested
- Proposed solution: let each agent compute its best response to other agents

Motivation

- Planning with multiple agents is hard
  - Joint action space is exponential in the number of agents
  - Agents may be self-interested
- Proposed solution: let each agent compute its best response to other agents
- Best response: plan that minimizes the cost to the agent, while satisfying its goals

Motivation

- Planning with multiple agents is hard
  - Joint action space is exponential in the number of agents
  - Agents may be self-interested
- Proposed solution: let each agent compute its best response to other agents
- Best response: plan that minimizes the cost to the agent, while satisfying its goals
- Plan for one agent at a time $\Rightarrow$ use single-agent planners

Notation

- A multi-agent problem (MAP) is a tuple $\Pi = \langle N, F, I, G, A, \Psi, c \rangle$, where
  - $N = \{1, \ldots, n\}$: set of agents
  - F: set of fluents
  - $I \subseteq F$: initial state
  - $G = G_1 \cup \ldots \cup G_n$: goal state
  - $A = A_1 \times \ldots \times A_n$: set of actions
  - $\Psi : A \rightarrow \{0, 1\}$: admissibility function
  - $c = (c_1, \ldots, c_n)$, where $c_i : A \rightarrow \mathbb{R}$ is the cost function of agent i

Notation

- A multi-agent problem (MAP) is a tuple $\Pi = \langle N, F, I, G, A, \Psi, c \rangle$, where
  - $N = \{1, \ldots, n\}$: set of agents
  - F: set of fluents
  - $I \subseteq F$: initial state
  - $G = G_1 \cup \ldots \cup G_n$: goal state
  - $A = A_1 \times \ldots \times A_n$: set of actions
  - $\Psi : A \rightarrow \{0, 1\}$: admissibility function
  - $c = (c_1, \ldots, c_n)$, where $c_i : A \rightarrow \mathbb{R}$ is the cost function of agent i
- Goal: find a plan $\pi = \langle a^1, \ldots, a^k \rangle$ of joint actions from I to G

Notation (cont.)

- Fluents can be partitioned as $F = F_1 \cup \ldots \cup F_n \cup F_{pub}$

Notation (cont.)

- Fluents can be partitioned as $F = F_1 \cup \ldots \cup F_n \cup F_{pub}$
- Each action $a_i = \langle pre(a_i), eff(a_i) \rangle \in A_i$ satisfies $pre(a_i) \subseteq F_i \cup F_{pub}$ and $eff(a_i) \subseteq F_i \cup F_{pub}$ (trivially extended to negative pre-conditions and effects)

Notation (cont.)

- Fluents can be partitioned as $F = F_1 \cup \ldots \cup F_n \cup F_{pub}$
- Each action $a_i = \langle pre(a_i), eff(a_i) \rangle \in A_i$ satisfies $pre(a_i) \subseteq F_i \cup F_{pub}$ and $eff(a_i) \subseteq F_i \cup F_{pub}$ (trivially extended to negative pre-conditions and effects)
- For each agent i, $G_i \subseteq F_i \cup F_{pub}$ (public goals are shared)

Notation (cont.)

- Fluents can be partitioned as $F = F_1 \cup \ldots \cup F_n \cup F_{pub}$
- Each action $a_i = \langle pre(a_i), eff(a_i) \rangle \in A_i$ satisfies $pre(a_i) \subseteq F_i \cup F_{pub}$ and $eff(a_i) \subseteq F_i \cup F_{pub}$ (trivially extended to negative pre-conditions and effects)
- For each agent i, $G_i \subseteq F_i \cup F_{pub}$ (public goals are shared)
- The cost of a plan π to agent i is $C_i(\pi) = \sum_{j=1}^{k} c_i(a^j)$

Admissibility function

- Represents concurrency constraints regarding individual actions

Admissibility function

- Represents concurrency constraints regarding individual actions
- A joint action $a \in A$ is part of the MAP ($\Psi(a) = 1$) or not ($\Psi(a) = 0$)

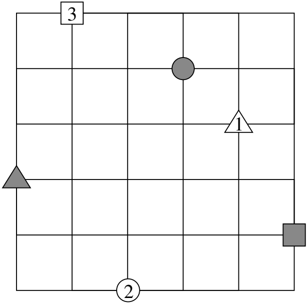
Admissibility function

- Represents concurrency constraints regarding individual actions
- A joint action $a \in A$ is part of the MAP ($\Psi(a) = 1$) or not ($\Psi(a) = 0$)
- Even though $|A|$ is exponential in n, Ψ can usually be represented compactly

Admissibility function

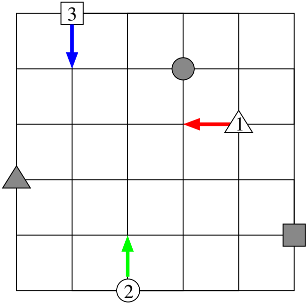
- Represents concurrency constraints regarding individual actions
- A joint action $a \in A$ is part of the MAP ($\Psi(a) = 1$) or not ($\Psi(a) = 0$)
- Even though $|A|$ is exponential in n, Ψ can usually be represented compactly
- Our approach requires quickly checking if a joint action is part of Π

Example



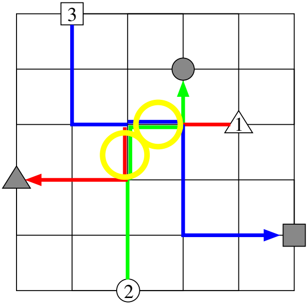
- Set of agents sending packages through a network
- F_i: current location of package i
- Action: send a package across a link of the network

Example (cont.)



- Joint action: each agent acts in parallel
- Cost to agent *i* of a joint action = number of agents simultaneously sending packages across the same link

Example (cont.)



- Figure shows example joint plan
- Cost is suboptimal in areas marked with yellow

Best-Response Planning

- Assume that there exists a joint plan $\pi = \langle a^1, \ldots, a^k \rangle$ of length $|\pi| = k$ for solving a MAP

Best-Response Planning

- Assume that there exists a joint plan $\pi = \langle a^1, \dots, a^k \rangle$ of length $|\pi| = k$ for solving a MAP
- Given an agent i, we define a best-response planning (BRP) problem as a tuple $\langle F', A', I', G', c' \rangle$, where
  - $F' = F_i \cup F_{pub} \cup \{time(0), \dots, time(k)\}$
  - $I' = (I \cap F') \cup \{time(0)\}$
  - $G' = G_i \cup \{time(k)\}$

Best-Response Planning (cont.)

- Each joint action of π is of the form $a^j = (a^j_i, a^j_{-i})$, where
  - a^j_i: the individual action of agent i
  - a^j_{-i}: the joint action of agents other than i

Best-Response Planning (cont.)

- Each joint action of π is of the form $a^j = (a_i^j, a_{-i}^j)$, where
  - a_i^j: the individual action of agent i
  - a_{-i}^j: the joint action of agents other than i
- For each $a_i \in A_i$, let $a = (a_i, a_{-i}^j)$ be the joint action that replaces a_i^j with a_i

Best-Response Planning (cont.)

- Each joint action of π is of the form $a^j = (a^j_i, a^j_{-i})$, where
  - a^j_i: the individual action of agent i
  - a^j_{-i}: the joint action of agents other than i
- For each $a_i \in A_i$, let $a = (a_i, a^j_{-i})$ be the joint action that replaces a^j_i with a_i
- If $\Psi(a) = 1$, add an action a' to A' such that
  - $pre(a') = (pre(a) \cap F') \cup \{time(j-1)\}$
  - $eff(a') = (eff(a) \cap F') \cup \{not(time(j-1)), time(j)\}$
  - $c'(a') = c_i(a)$

Best-Response Planning (cont.)

- Add noop actions $noop_i$, applicable when agents are done with other actions

Best-Response Planning (cont.)

- Add noop actions $noop_i$, applicable when agents are done with other actions
- For each $a_i \in A_i$, let $a = (a_i, noop_{-i})$ be the joint action composed of a_i and the noop action for each other agent

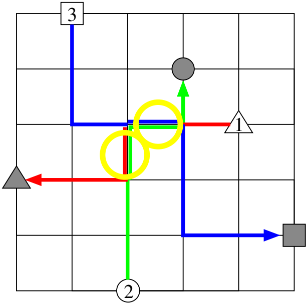
Best-Response Planning (cont.)

- Add noop actions $noop_i$, applicable when agents are done with other actions
- For each $a_i \in A_i$, let $a = (a_i, noop_{-i})$ be the joint action composed of a_i and the noop action for each other agent
- Add an action a' to A' such that
  - $pre(a') = (pre(a) \cap F') \cup \{time(k)\}$
  - $eff(a') = eff(a) \cap F'$
  - $c'(a') = c_i(a)$

Best-Response Planning (cont.)

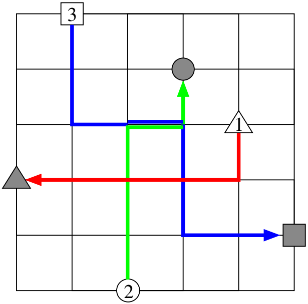
- To compute the best response of agent i to the actions of other agents, solve the BRP problem using an optimal planner
- Replace the actions for i with the actions of the new plan
- Iterate over each agent until no agent can improve its cost

Example (cont.)



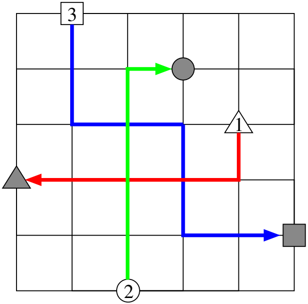
- Given the actions of agents 2 and 3, agent 1 performs best-response planning

Example (cont.)



- To agent 1, the new plan is cheaper and still solves the problem
- Repeat the process for agent 2

Example (cont.)



- Eventually, no agent can improve their cost by choosing a cheaper plan

Congestion Games

- In game theory, a congestion game is a tuple $\langle N, R, A, c \rangle$, where
  - $N = \{1, \ldots, n\}$: set of agents
  - $R = \{r_1, \ldots, r_m\}$: set of resources
  - $A = A_1 \times \ldots \times A_n$, where $A_i \subseteq 2^R - \emptyset$ is the action set of agent i,
  - $c = (c_{r_1}, \ldots, c_{r_m})$, where $c_r : \mathbb{N} \rightarrow \mathbb{R}$ is the cost function of resource r
- An action consists in selecting a non-empty subset of resources

Congestion Games

- In game theory, a congestion game is a tuple $\langle N, R, A, c \rangle$, where
  - $N = \{1, \ldots, n\}$: set of agents
  - $R = \{r_1, \ldots, r_m\}$: set of resources
  - $A = A_1 \times \ldots \times A_n$, where $A_i \subseteq 2^R - \emptyset$ is the action set of agent i,
  - $c = (c_{r_1}, \ldots, c_{r_m})$, where $c_r : \mathbb{N} \rightarrow \mathbb{R}$ is the cost function of resource r
- An action consists in selecting a non-empty subset of resources
- The utility function of agent i is $u_i(a) = -\sum_{r \in a_i} c_r(\#(r, a))$
- $\# : R \times A \rightarrow \mathbb{N}$ counts the number of agents selecting a resource

Congestion Games (cont.)

- Define a potential function $Q(a) = \sum_{r \in R} \sum_{j=1}^{\#(r,a)} c_r(j)$
- Given two joint actions (a_i, a_{-i}) and (a'_i, a_{-i}), it holds that $u_i(a_i, a_{-i}) - u_i(a'_i, a_{-i}) = Q(a_i, a_{-i}) - Q(a'_i, a_{-i})$.

Congestion Games (cont.)

- Define a potential function $Q(a) = \sum_{r \in R} \sum_{j=1}^{\#(r,a)} c_r(j)$
- Given two joint actions (a_i, a_{-i}) and (a'_i, a_{-i}), it holds that $u_i(a_i, a_{-i}) - u_i(a'_i, a_{-i}) = Q(a_i, a_{-i}) - Q(a'_i, a_{-i})$.
- Games that satisfy this property are known as potential games
- Iterative best-response is guaranteed to converge to Nash equilibrium

Extending Congestion Games

- Define a new utility function $u'_i(a) = u_i(a) - d_i(a_i)$ and a new potential function $Q'(a) = Q(a) - \sum_{j \in N} d_j(a_j)$

Extending Congestion Games

- Define a new utility function $u'_i(a) = u_i(a) - d_i(a_i)$ and a new potential function $Q'(a) = Q(a) - \sum_{j \in N} d_j(a_j)$
- It is easy to show that this is still a potential game:

$$
\begin{aligned}
Q'(a_i, a_{-i}) - Q'(a'_i, a_{-i}) &= Q(a_i, a_{-i}) - d_i(a_i) - \sum_{j \in N - \{i\}} d_j(a_j) - \\
&- Q(a'_i, a_{-i}) + d_i(a'_i) + \sum_{j \in N - \{i\}} d_j(a_j) = \\
&= Q(a_i, a_{-i}) - Q(a'_i, a_{-i}) - d_i(a_i) + d_i(a'_i) = \\
&= u_i(a_i, a_{-i}) - u_i(a'_i, a_{-i}) - d_i(a_i) + d_i(a'_i) = \\
&= u'_i(a_i, a_{-i}) - u'_i(a'_i, a_{-i})
\end{aligned}
$$

Congestion Planning

- Let $R = \{r_1, \ldots, r_m\}$ be a set of resources, each with a cost function $c'_r : \mathbb{N} \to \mathbb{R}$

Congestion Planning

- Let $R = \{r_1, \ldots, r_m\}$ be a set of resources, each with a cost function $c'_r : \mathbb{N} \rightarrow \mathbb{R}$
- A congestion planning problem (CPP) is a MAP augmented with R and $c' = (c'_{r_1}, \ldots, c'_{r_m})$ such that each action a_i is associated with a subset of resources $R(a_i) \subseteq R$ and

Congestion Planning

- Let $R = \{r_1, \ldots, r_m\}$ be a set of resources, each with a cost function $c'_r : \mathbb{N} \rightarrow \mathbb{R}$
- A congestion planning problem (CPP) is a MAP augmented with R and $c' = (c'_{r_1}, \ldots, c'_{r_m})$ such that each action a_i is associated with a subset of resources $R(a_i) \subseteq R$ and
  1. $F_{pub} = \emptyset$
  2. $\Psi(a) = 1$ for each joint action $a \in A$
  3. The cost function of agent i is $c_i(a) = \sum_{r \in R(a_i)} c'_r(\#(r, a)) + d_i(a_i)$
  4. A noop action $noop_i$ uses no resources and incurs no cost, i.e. $R(noop_i) = \emptyset$ and $d_i(noop_i) = 0$

Congestion Planning (cont.)

Theorem

For congestion planning problems, best-response planning is guaranteed to converge to a Nash equilibrium.

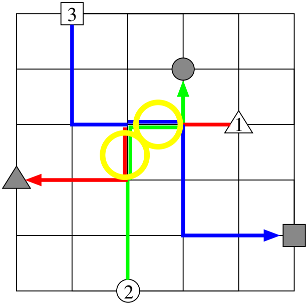
Proof.

For each joint plan $\pi = \langle a^1, \ldots, a^k \rangle$, define a potential function $Q(\pi) = \sum_{j=1}^{k} Q'(a^j)$. Consider two plans π and π' that only differ on the action choice of agent i. We have

$$\begin{aligned} Q(\pi) - Q(\pi') &= \sum_{j=1}^{k} (Q'(a^j) - Q'(a^{j'})) = \sum_{j=1}^{k} (u'_i(a^j) - u'_i(a^{j'})) = \\ &= \sum_{j=1}^{k} (c_i(a^{j'}) - c_i(a^j)) = C_i(\pi') - C_i(\pi). \end{aligned}$$

□

Example (cont.)



- Example MAP is a CPP!
- No public fluents nor goals
- Resources = links, cost of a link = number of agents using it

Experiments

- Two sets of experiments with BRP
- First set: network example, for different numbers of nodes and agents
- Second set: IPC domains with multi-agent flavor
- For each BRP problem, generate corresponding problem in PDDL
- Use HSP_f [Haslum 2008] to plan optimally

Network Example

- Example of a congestion planning problem
- Finding initial plan is easy (just assume no other agents are using resources)
- By the previous theorem, BRP is guaranteed to converge to a Nash equilibrium
- For 100 nodes and 100 agents, BRP converges in 10 minutes

IPC Domains

- Multi-agent problems from Logistics, Rovers, and Satellite
- Use DisCSP planner [Nissim et al. 2010] to find initial plans
- In Rovers, HSP_f fails to solve BRP problems, so we use LAMA [Richter & Westphal 2010] to generate suboptimal plans

IPC Domains (cont.)

Prob.	DisCSP T	DisCSP C	DisCSP M	BR-Optimal T	BR-Optimal I	BR-Optimal C	BR-Optimal M	BR-Satisficing T	BR-Satisficing I	BR-Satisficing C	BR-Satisficing M
Log_3_1	1.3	10	9	0.2	1	10	9	-	-	-	-
Log_4_2	307.0	14	12	0.6	3	14	6	-	-	-	-
Rov_3	53.0	33	13	-	-	-	-	179.6	2	34	13
Rov_4	408.4	44	14	-	-	-	-	414.8	2	45	14
Rov_5	784.2	55	15	-	-	-	-	2170.7	3	55	15
Rov_6	3958.7	66	16	-	-	-	-	2235.2	2	66	16
Sat_2	0.5	7	4	0.2	2	7	4	0.8	2	7	4
Sat_4	1.2	14	6	1.5	2	14	4	5.7	3	14	6
Sat_6	3.4	21	8	19.4	2	21	4	13.5	2	21	8
Sat_8	25.5	28	10	178.0	2	28	4	37.6	2	28	10

Conclusion

- A single-agent approach to multi-agent planning
- Each agent optimizes its own cost
- For congestion planning problems, guaranteed to converge
- In practice, converges in three IPC domains

Future Work

- Determine convergence guarantees for larger classes of MAPs
- Use single-agent approach to generate initial plans
- Best-response planning when public goals are not shared by agents
- Advances in single-agent planning will benefit BRP