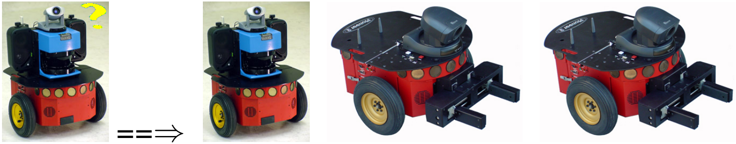
A Formal Analysis of Required Cooperation in Multi-agent Planning

Yu ("Tony") Zhang   Subbarao Kambhampati

YOCHAN Research Group
Computer Science and Engineering Department
Arizona State University

DMAP
International Conference on Automated Planning and Scheduling, 2014

Multi-agent Planning



Through planning for cooperation, multi-agent systems can achieve tasks that are unachievable by single-agent systems

Previous Research

Previous research on multi-agent planning:

- Complexity analysis [Brafman and Domshlak, 2008]
- DisCSP-based search [Nissim et al., 2010]
- A^*-based search [Nissim and Brafman, 2012]
- POP-based search [Torreno et al., 2012]
- SAP-based iterative improvement [Jonsson and Rovatsos, 2011]

However, what characterizes multi-agent planning is undefined:

When cooperation is required

When cooperation is required

When multiple agents are used, are they

strictly required to solve the problem?	VS.	used to increase the efficiency (e.g., via parallelism) of the solution?

We attempt to characterize this in terms of "Required Cooperation"

This analysis can help:

- Reveal the structure of multi-agent planning problems
- Provide useful information on the "right" number of agents to solve a problem

Contributions

In this work, we:

- Introduce the notion of required cooperation (RC) and the problems to determine RC and the minimum number of agents required (k-agent solvable)
- Provide formal characterizations of situations where cooperation is required
- Provide upper bounds on the minimum number of agents required for RC problems

SAS$^+$

A SAS$^+$ problem is given by a tuple $P = \langle V, A, I, G \rangle$, where:

- $V = \{v_1, ..., v_n\}$ is a set of state variables. Each variable $v_i \in V$ is associated with its domain $D(v_i)$
- $A = \{a_1, ..., a_m\}$ is a finite set of actions. Each action a_j is a tuple $\langle pre(a_j), post(a_j), prv(a_j) \rangle$
- I and G denote the initial and goal state

A plan π is a sequence of actions $\pi = \langle a_1, ..., a_l \rangle$

$$re(s, \langle \pi; o \rangle) = \begin{cases} re(s, \langle \pi \rangle) \oplus post(o) & \text{if } pre(o) \sqcup prv(o) \sqsubseteq re(s, \langle \pi \rangle) \\ s & \text{otherwise} \end{cases}$$

Extending SAS$^+$ to MAP

A SAS$^+$ MAP problem is given by a tuple $\Pi = \langle V, \Phi, I, G \rangle$, where:

- $\Phi = \{\phi_g\}$ is the set of agents
- Each agent ϕ_g is associated with a set of actions $A(\phi_g)$

A plan π_{MAP} in MAP is a sequence of agent-action pairs:

- $\pi_{MAP} = \langle (a_1, \phi(a_1)), ..., (a_L, \phi(a_L)) \rangle$

Extension to temporal domain to consider concurrency and/or synchronization is to be studied in future work

Required Cooperation

Definition (k-agent Solvable)

Given a MAP problem $P = \langle V, \Phi, I, G\rangle$ ($|\Phi| \geq k$), the problem is k-agent solvable if $\exists \Phi_k \subseteq \Phi$ ($|\Phi_k| = k$), such that $\langle V, \Phi_k, I, G\rangle$ is solvable.

Definition (Required Cooperation (RC))

Given a solvable MAP problem $P = \langle V, \Phi, I, G\rangle$, there is required cooperation if it is not 1-agent solvable.

Definition (Minimally k-agent Solvable)

Given a solvable MAP problem $P = \langle V, \Phi, I, G\rangle$ ($|\Phi| \geq k$), it is minimally k-agent solvable if it is k-agent solvable, and not $(k-1)$-agent solvable.

Complexity Results

Lemma

Given a solvable MAP problem $P = \langle V, \Phi, I, G \rangle$, determining whether it satisfies RC is PSPACE-complete.

Corollary

Given a solvable MAP problem $P = \langle V, \Phi, I, G \rangle$, determining the minimally solvable k ($k \leq |\Phi|$) is PSPACE-complete.

Although directly querying for RC is intractable, we aim to identify all the conditions that can cause RC

Agent Capability

To specify state and capability that are independent of agents:

Definition (Action Signature (AS))

An action signature is an action with the reference of the executing agent replaced by a global *EX-AG* symbol.

Action signatures are grounded (instantiated) except for the agent field; $AS(\phi)$ for the action signatures of agent ϕ

E.g., *Drive*(*EX-AG*, *pgh-po*, *pgh-airport*)

Agent State

Definition (Agent Variable (Agent Fluent))

A variable (fluent) is an agent variable (fluent) if it is associated with the reference of an agent.

Agent variables are used to specify agent state, V_ϕ for agent ϕ

E.g., *location*(*truck-pgh*)

Definition (Variable (Fluent) Signature (VS))

Given an agent variable (fluent), its variable (fluent) signature is the variable (fluent) with the reference of agent replaced by *EX-AG*.

$VS(\phi)$ for the variable signatures of V_ϕ

E.g., *location*(*EX-AG*)

Assumptions

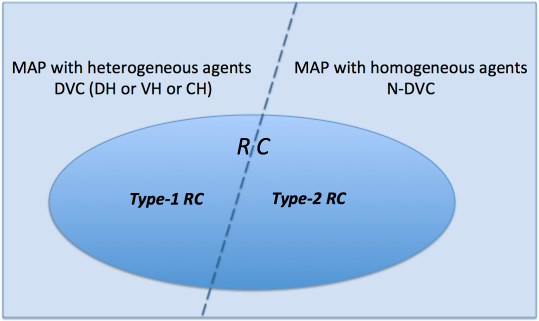
- Agents can only interact with each other through non-agent variables, V_o
- Agent variables for different agents are positively and non-exclusively defined

E.g., positively and non-exclusively defined: $equipped\_for\_imaging(rover)$, $equipped\_for\_rock\_analysis(rover)$

E.g., exclusively defined: $use\_gas(trunk)$, $use\_kerosene(plane)$

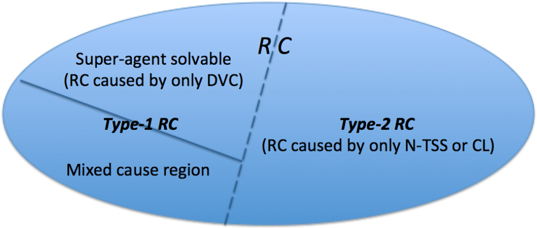
Negative or exclusive definitions can be compiled away
E.g., combining multiple variables into a single variable, $use(agent) = \{gas, kerosene\}$

Division of RC Problems



Heterogeneity (i.e., DVC) of agents is defined using AS and VS

Division of RC Causes



- RC in Type-1 RC problems (with heterogeneous agents) may not be caused by heterogeneity
- RC in Type-2 RC problems (with homogeneous agents) is only caused by N-TSS or CL
- RC in RC problems is only introduced by DVC or N-TSS or CL

Heterogeneity

Given a MAP problem $P = \langle V, \Phi, I, G \rangle$, the heterogeneity of agents can come from:

- *Domain Heterogeneity (DH)*: $\exists v \in V_\phi$ and $D(V') \setminus D(v) \neq \emptyset$, in which $V' = \{v' | v' \in V_{\phi'} (\phi' \neq \phi)$ and $VS(v) = VS(v')\}$
  E.g., $use(truck) = gas$, $use(plane) = kerosene$
- *Variable Heterogeneity (VH)*: $VS(\Phi \setminus \phi) \setminus VS(\phi) \neq \emptyset$
  E.g., $equipped\_for\_imaging(rover)$
- *Capability Heterogeneity (CH)*: $AS(\Phi \setminus \phi) \setminus AS(\phi) \neq \emptyset$
  E.g., $Fly(plane, A, B)$ and $Drive(truck, A, B)$

Type-1 RC

Definition (Type-1 RC)

An RC problem belongs to type-1 RC if at least one of DH, VH and CH is satisfied for an agent.

Most of the RC problems in the IPC domains belong to type-1 RC

- Presence of DVC (i.e., heterogeneity) in a solvable MAP problem does not always cause RC

  E.g., Define an agent that is not needed
- Presence of DVC in a type-1 RC problem is not always the cause of RC

Super-agent solvable

Definition (Combined Agent (Super Agent))

A super agent is an agent ϕ^* that satisfies:

- $\forall v \in V_\Phi, \exists v^* \in V_{\phi^*}, D(v^*) = D(V)$, in which $V = \{v | v \in V_\Phi$ and $VS(v^*) = VS(v)\}^a$
- $VS(\phi^*) = VS(\Phi)$
- $AS(\phi^*) = AS(\Phi)$

[a] Actions to switch between the static value options must be added, e.g., *Switch_to_use*(*agent*, {*gas*, *kerosene*}), for the *combined agent*

Generally, super agent construction does not combine (or choose) the (initial) states of agents, e.g., *location*(*EX*-*AG*)

Exceptions (when state is expressed as the access of a variable), e.g., *equipped_for_imaging*(*EX*-*AG*)

Most of the RC problems in the IPC domains are super agent solvable

Type-2 RC

Definition (Type-2 RC)

An RC problem belongs to type-2 RC if it satisfies N-DVC (for all agents).

Factors for Type-2 RC problems? Agent or world state.

E.g., one-way road, $1 \rightarrow 2$, $1 \rightarrow 3$, $Drive(agent, 1, 3)$

E.g., non-replenishable resources, $has\_bullet(agent) = 10$

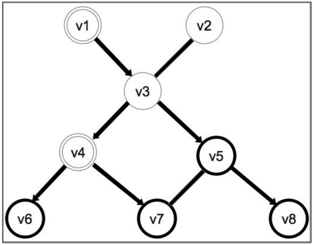
Causal Graph

Definition (Causal Graph)

Given a MAP problem $P = \langle V, \Phi, I, G \rangle$, the causal graph G is a graph with directed and undirected edges over the nodes V. For two nodes v and v' ($v \neq v'$), a directed edge $v \rightarrow v'$ is introduced if there exists an action that updates v' while having a prevail condition associated with v. An undirected edge $v - v'$ is introduced if there exists an action that updates both.

- agents are homogeneous $\rightarrow$ the causal graphs are the same
- use agent VSs $\rightarrow$ *individual causal graph signature* (ICGS)

OC and IC



Definition (Inner and Outer Closures (IC and OC))

An inner closure (IC) in an ICGS is any set of variables for which no other variables are connected to them with undirected edges; an outer closure (OC) of an IC is the set of nodes that have directed edges going into nodes in the IC.

E.g., $\{v_2, v_3\}$ is an IC, and its OC is $\{v_1\}$

Define local state space connectivity

TSS

Definition ((Locally) Traversable State Space (TSS))

An IC has a traversable state space if and only if: given any two states of this IC, denoted by s and s', there exists a plan that connects them, assuming that the state of the OC of this IC can be changed freely within its state space.

E.g., one-way road, $1 \rightarrow 2$, $1 \rightarrow 3$, $Drive(agent, 1, 3)$, $location(agent)$

E.g., non-replenishable resources, $Shoot(agent, X)$, $has\_bullet(agent) = 10$

An ICGS is (locally) traversable if all ICs satisfy TSSs

Causal Loop

Definition (Causal Loop (CL))

A causal loop in the ICGS is a directed loop that contains at least one directed edge.

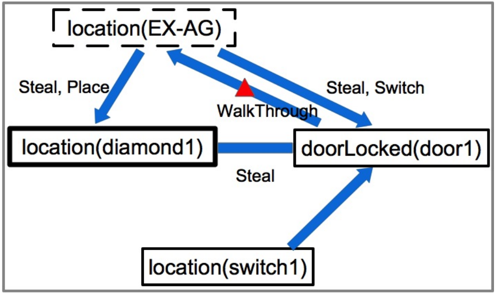


Figure : Diamond example that illustrates causal loop, which can cause RC in type-2 RC problems

Gap between MAP and Single Agent Planning

Theorem

Given a solvable MAP problem that satisfies N-DVC for all agents, and for which the ICGS is traversable and contains no causal loops, any single agent can also achieve the goal.

Upper Bounds for the Minimal K

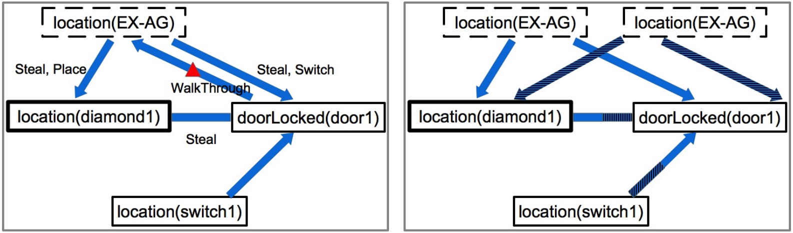


Figure : Illustration of the process for breaking causal loops in the diamond example.

Relaxing Causal Loops

Lemma

Given a solvable MAP problem that satisfies N-DVC for all agents and for which the ICGS is traversable, if no CLs contain agent VSs and all the edges going in and out of agent VSs are directed, the minimum number of agents required is upper bounded by $\times_{v \in CR(\Phi)} |D(v)|$*, when assuming that the agents can choose their initial states, in which* $CR(\Phi)$ *is constructed as follows:*

1. *add the set of agent VSs that are in the CLs into* $CR(\Phi)$*;*
2. *add in an agent VS into* $CR(\Phi)$ *if there exists a directed edge that goes into it from any variable in* $CR(\Phi)$*;*
3. *iterate* 2 *until no agent VSs can be added.*

Relaxing TSS

Lemma

Given a solvable MAP problem that satisfies N-DVC for all agents, if all the edges going in and out of agent VSs are directed, the minimum number of agents required is upper bounded by $\times_{v \in VS(\Phi)} |D(v)|$*, when assuming that the agents can choose their initial states.*

- Diamond example, bounds returned are 2 for both
- For discrete domains

Contributions

In this work, we:

- Introduce the notion of required cooperation (RC) and the problems to determine RC and the minimum number of agents required (k-agent solvable)
- Provide formal characterizations of situations where cooperation is required
- Provide upper bounds on the minimum number of agents required for RC problems

For future work, we plan to:

- Extend to temporal domain
- Provide tighter bounds for the minimal k, and extend to continuous domain

References

Brafman, R. I. and Domshlak, C. (2008).

From One to Many: Planning for Loosely Coupled Multi-Agent Systems.

In *Proceedings of the 18th International Conference on Automated Planning and Scheduling*, pages 28–35. AAAI Press.

Jonsson, A. and Rovatsos, M. (2011).

Scaling Up Multiagent Planning: A Best-Response Approach.

In *Proceedings of the 21th International Conference on Automated Planning and Scheduling*, pages 114–121. AAAI Press.

Nissim, R. and Brafman, R. I. (2012).

Multi-agent a* for parallel and distributed systems.

In *Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems*, volume 3, pages 1265–1266.

Nissim, R., Brafman, R. I., and Domshlak, C. (2010).
A general, fully distributed multi-agent planning algorithm.
In *Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems*, pages 1323–1330.

Torreno, A., Onaindia, E., and Sapena, O. (2012).
An approach to multi-agent planning with incomplete information.
In *European Conference on Artificial Intelligence*, volume 242, pages 762–767.