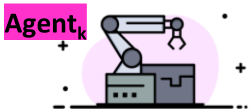
Multiagent Task Allocation and Planning with Multi-Objective Requirements



Tasks as Linear Temporal logic

$\varphi ::= a \mid \neg\varphi \mid \varphi_1 \wedge \varphi_2 \mid X\varphi \mid \varphi_1 U \varphi_2$

E.g. "Eventually visit location 3"

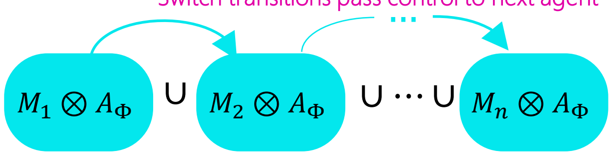
Agents as Markov Decision Processes

Represent co-safe task *j* as DFA automata A_j:

$$MDP_k = (S, s_0, A, P, L) \otimes \left(\bigotimes_{j=1}^{m} A_j = (Q, q_0, \Sigma, \delta, Q_{j,F}) \right)$$

Goal: **Allocate multiple tasks to multiple agents.**

Method: **Introduce a *team* MDP model to solve task allocation and planning as a reachability problem.**



This is powerful because taking the union of MDP product models exponentially reduces the state space.

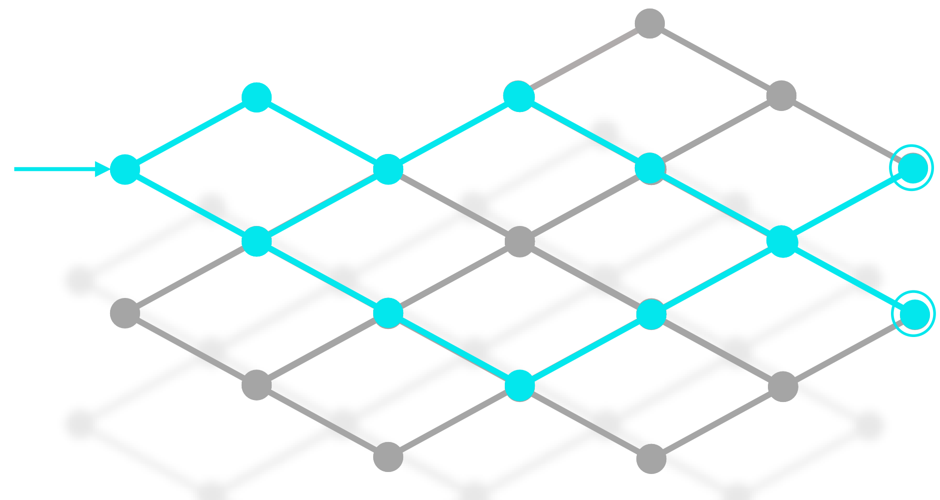
We then introduce an algorithm to exploit the team structure.

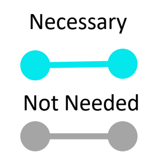
Result: **Comparison with conventional model.**

#Agents	Team MDP t	Team MDP $\|S\|$	Team MDP $\|P\|$	MAMDP t	MAMDP $\|S\|$	MAMDP $\|P\|$
3	9	4320	9504	9	26091	259853
4	12	7488	21488	77	260901	3.3×10^6
5	17	9360	34992	627	2.1×10^6	2.9×10^7
6	23	11232	50976	7585	2×10^7	3.6×10^8
7	30	13104	69120	-	-	-

*MAMDP – Multiagent Markov Decision Process

Manipulating task-agent graph structures significantly reduces computation time for multi-objective task allocation and planning in multiagent systems, scaling linearly in the number of agents.





Natural restriction on tasks:
(i) An agents can process one task at any given time.
(ii) There is only one action relevant to a task progress in any reachable state.
(iii) After a task is completed the MDP returns back to its initial state.

Thomas Robinson, Guoxin Su, Minjie Zhang