Team Coordination on Graphs with State-Dependent Edge Costs

Manshi Limbu, Zechen Hu, Sara Oughourli,
Xuan Wang, Xuesu Xiao, and Daigo Shishika
George Mason University

INTRODUCTION

Human-robot teams traversing an environment with risks can provide support for each other from specific nodes.

We want to know:

- When such support/coordination is beneficial?
- How to best coordinate the actions as a team to minimize the overall cost?

PROBLEM FORMULATION

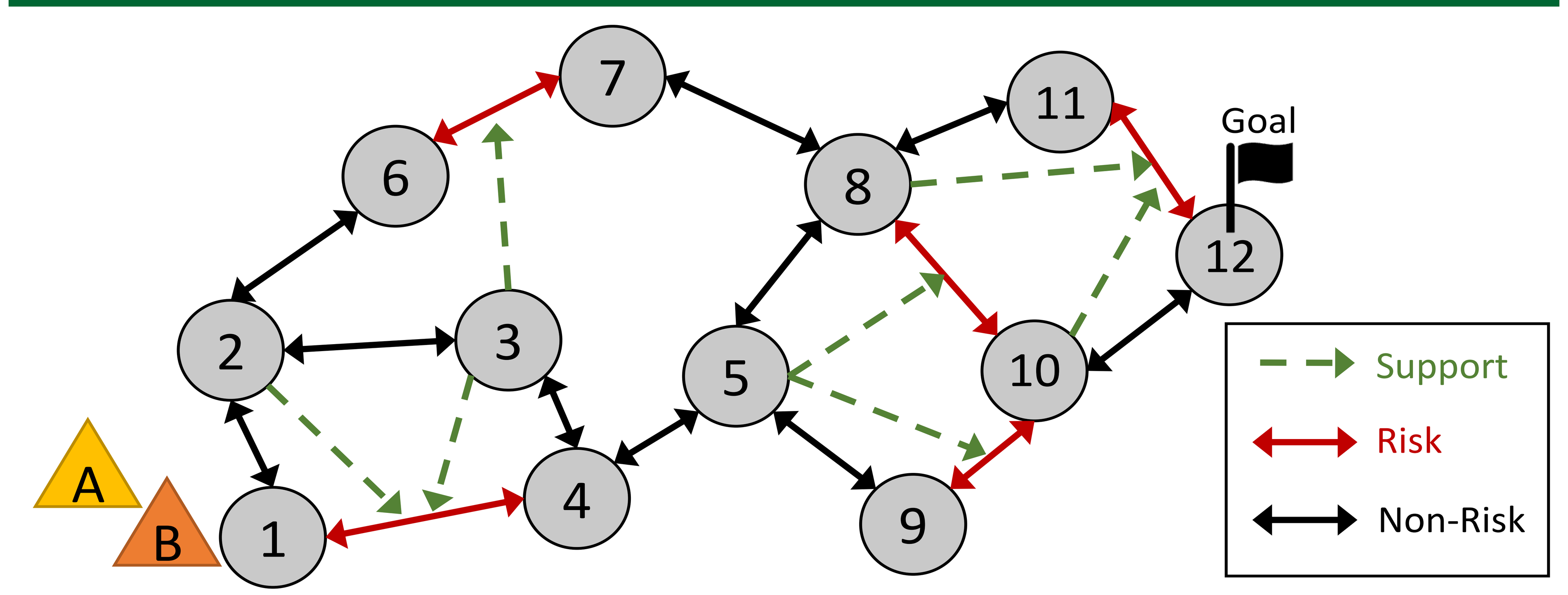
Formulate it as a minimum-cost graph traversal problem:

- Base graph $\mathbb{G} = (\mathcal{V}, \mathcal{E})$.
- Environment graph incorporates a notion of risk and support.
- Each edge $e_{i,j} \in \mathcal{E}$ is associated with a set of support nodes $\mathcal{Z}_{i,j} \subseteq \mathcal{V}$.
- Action set for agent n at node i is $\mathcal{A}_i^n = \left\{ \{a_{i,j}\}_{j \in \mathcal{N}_i}, a_s \right\}$.
- The different costs for agent A is:

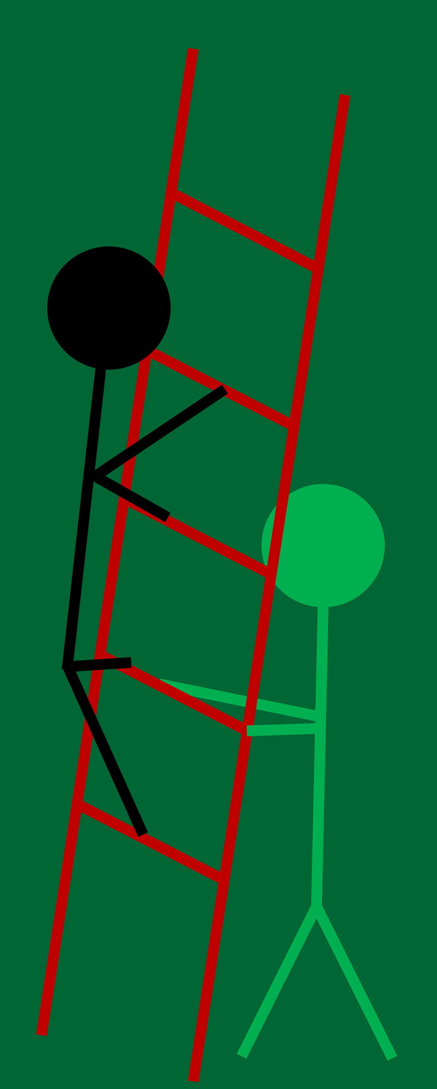
$$c_A^t(p^t, a^t) = \begin{cases} c_{i,j}, & \text{if } a_A = a_{i,j} \text{ and } p_B \notin \mathcal{Z}_{i,j} \text{ or } a_B \neq a_s, \\ \tilde{c}_{i,j}, & \text{if } a_A = a_{i,j}, p_B \in \mathcal{Z}_{i,j}, \text{and } a_B = a_s, \\ \tilde{c}, & \text{if } a_A = a_s, \\ 0, & \text{if } a_A \neq a_s \text{ and if } a_A \neq a_{i,j}. \end{cases}$$

- Compute costs of each action in a sequence to obtain overall cost.
- Goal is to find a pair of sequences (one for each agent) that minimizes overall costs.

We provide a problem formulation and two methods for solving multi-agent cooperation on a graph with a notion of ***risk*** and ***support***.



Environment graph with *risk* edges and *support* nodes



One agent provides *support* by holding up the *ladder* while the other agent climbs.

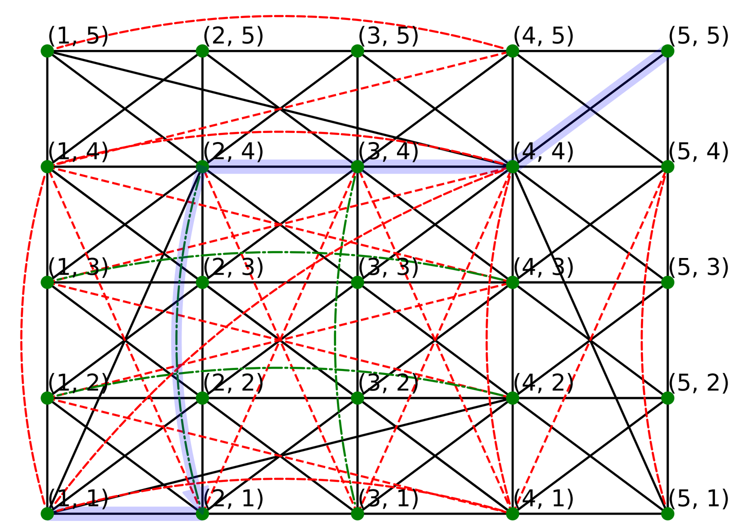
Experimentally, we find that CJSG is more efficient overall than JSG in generating optimal path planning solutions.

METHODS

Joint State Graph (JSG):

- Nodes represent the joint states.
- Edges represent possible transitions between those joint states.
- Cost of each edge is the sum of costs for each agent's actions.

The point is that JSG ***subsumes*** the action selection of the original problem, converting it into a single-agent path planning problem on JSG that can be solved with any standard shortest-path algorithm. However, it can be computationally expensive with greater graph sizes.

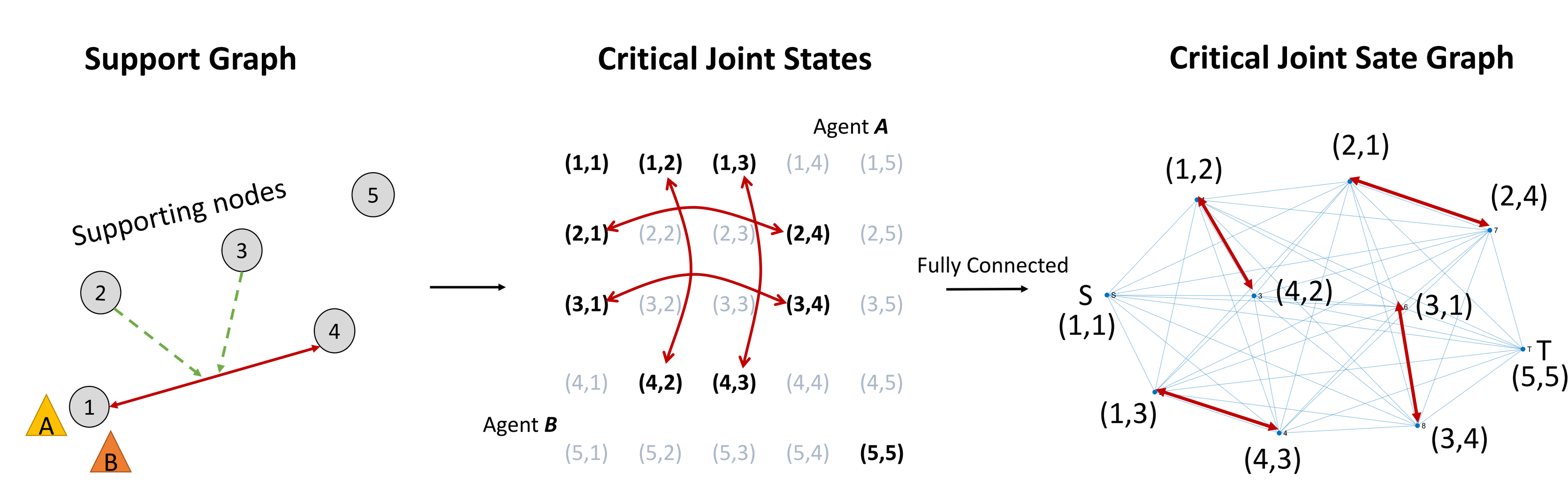


Joint State Graph for a 5-node environment graph. Red (green) edges represent traversing risk edge without (with) support.

Critical Joint State Graph (CJSG):

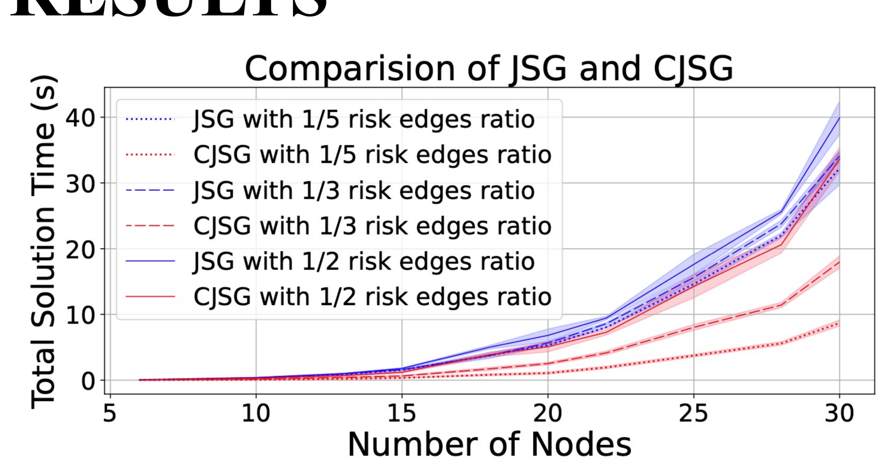
To address JSG's computational inefficiency, we propose to classify the agents' movements into coupled and decoupled modes:

- Coupled movements are planned in JSG, where supporting behavior is possible.
- Decoupled movements are independently planned by each agent on base graph.



Fully connected graph where nodes represent joint states where agents initiate/complete support and the start/goal nodes.

RESULTS



Nodes	Risky Edge	JSG		CJSG	
		GC	SP	GC	SP
10	1/5	0.21±0.04	0.14±0.02	0.01±0.00	0.02±0.02
	1/3	0.18±0.05	0.15±0.00	0.09±0.02	0.13±0.02
	1/2	0.19±0.03	0.16±0.01	0.18±0.01	0.15±0.05
20	1/5	3.17±0.04	2.10±0.04	0.65±0.09	0.43±0.07
	1/3	3.40±0.06	2.20±0.07	1.77±0.05	0.80±0.01
	1/2	3.86±0.09	2.32±0.03	3.64±0.59	1.25±0.12
30	1/5	20.94±0.83	11.64±0.13	6.11±0.55	2.80±0.18
	1/3	22.49±1.71	12.33±0.50	13.82±0.73	4.60±0.22
	1/2	25.98±0.43	13.44±0.14	26.82±1.49	6.91±0.27