Improved Heuristics for Multi-Agent Path Finding with Conflict-Based Search

Jiaoyang Li, Eli Boyarski, Ariel Felner, Hang Ma and Sven Koenig
jiaoyanl@usc.edu, eli.boyarski@gmail.com, felner@bgu.ac.il, {hangma,skoenig}@usc.edu

USC University of Southern California
Ben-Gurion University of the Negev

This paper will be published at IJCAI 2019. The research at the University of Southern California was supported by the National Science Foundation (NSF) under grant numbers 1409987, 1724392, 1817189 and 1837779 as well as a gift from Amazon. The research was also supported by the United States-Israel Binational Science Foundation (BSF) under grant number 2017692.

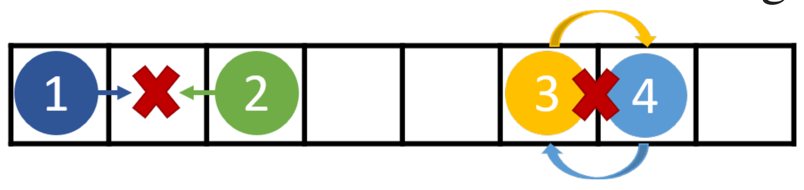
Abstract

Conflict-Based Search (CBS) and its enhancements are among the strongest algorithms for Multi-Agent Path Finding. Recent work introduced an admissible heuristic (called here CG) to guide the high-level search of CBS. In this work, we introduce two new admissible heuristics, DG and WDG, by reasoning about the pairwise dependency between agents. Empirically, CBS with both new heuristics significantly improves the success rate over CBS with the recent heuristic and reduces the number of expanded nodes and runtime by up to a factor of 50.

Multi-Agent Path Finding

Multi-Agent Path Finding is the problem of finding a set of paths for a team of agents on a given graph.

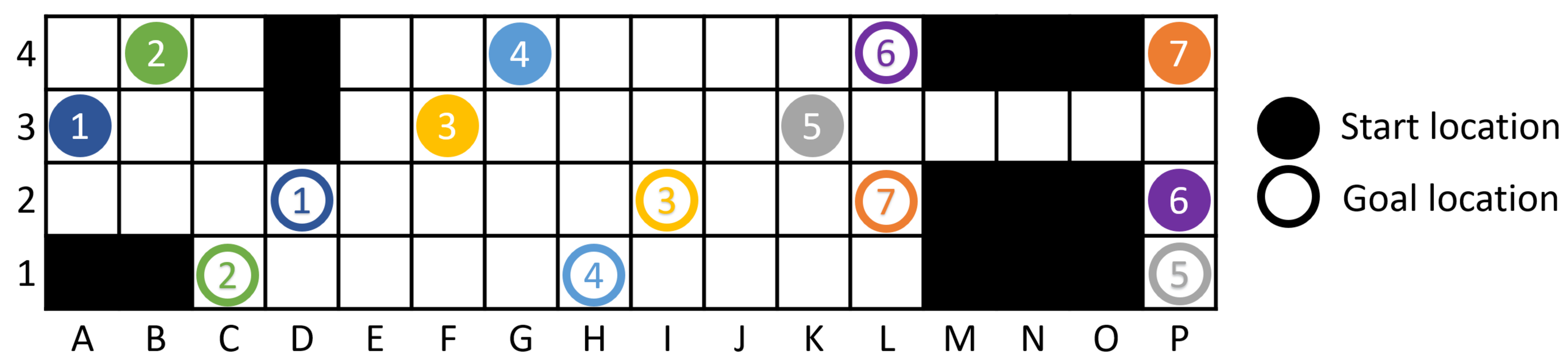
- Each agent is required to move from a given start location to a given goal location, while avoiding collisions with others.
- At each timestep, every agent can either wait at its current location or move to a neighbor location.
- The objective is to minimize the sum of the costs of the paths.
- A conflict is either a vertex collision or an edge collision:



Conflict-Based Search (CBS) [2]

1. Find a shortest path for every agent independently.
2. Check for conflicts among paths.
3. If there is a conflict where both agent A and agent B stay in location v at timestep t, try both options:
   - Option 1: prohibit A from staying in v at t.
   - Option 2: prohibit B from staying in v at t.
4. Repeat until finding a set of conflict-free paths.

Heuristics for Conflict-Based Search

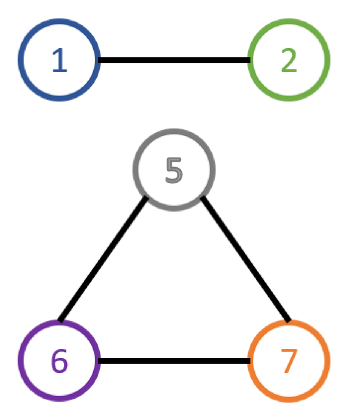


I. CG Heuristics [1]

1. Cardinal Conflicts

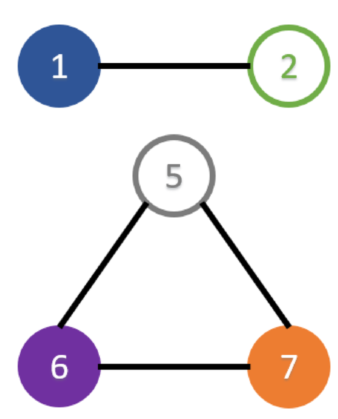
Two agents have a cardinal conflict iff all their shortest paths use the same vertex/edge at the same timestep.

2. Conflict Graph



Edges represent cardinal conflicts.

3. Minimum Vertex Cover (MVC)



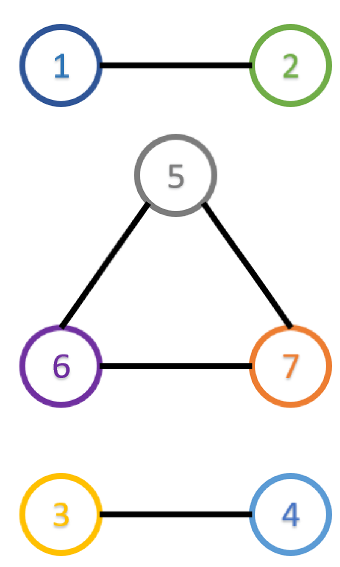
$h_{CG} = 3$

II. DG Heuristics

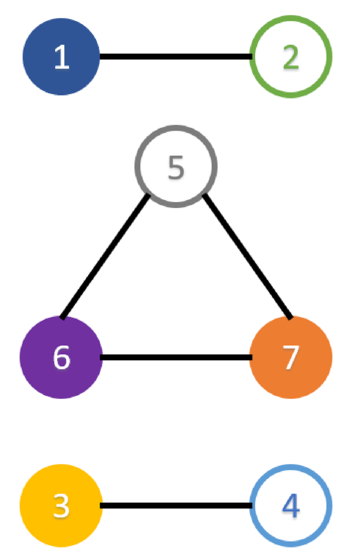
1. Pairwise Dependency

Two agents are dependent iff every pair of their shortest paths has at least one conflict.

2. Dependency Graph



3. MVC



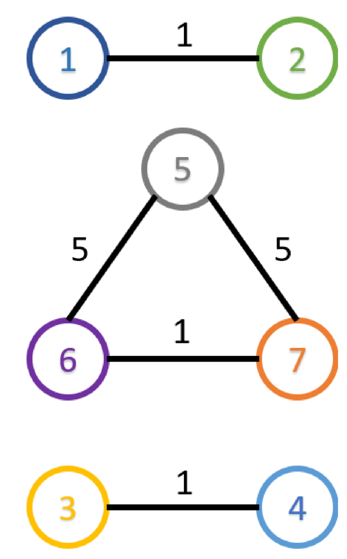
$h_{DG} = 4$

III. WDG Heuristics

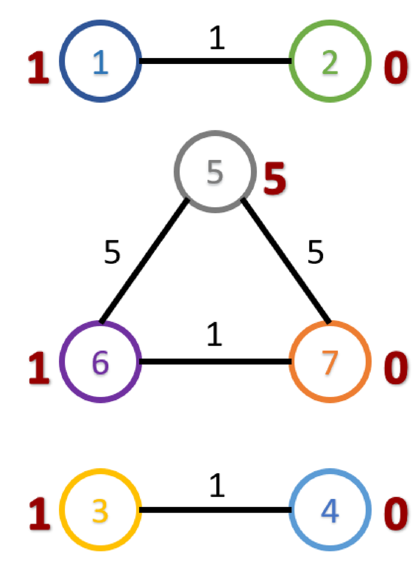
1. Dependency Weight

The weight for a pair of agents is the difference between the minimum sum of the costs of their conflict-free paths and the sum of the costs of their shortest paths.

2. Weighted Dependency Graph



3. Edge-Weighted MVC

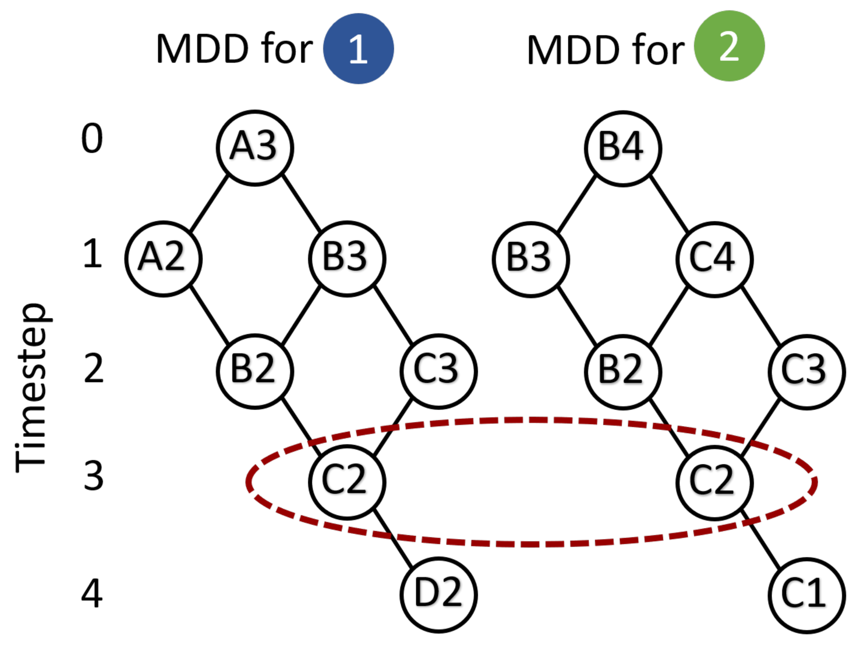


$h_{WDG} = 1 + 5 + 1 + 1 = 8$

Build Cardinal/Denpendency/Weighted Dependency Graph

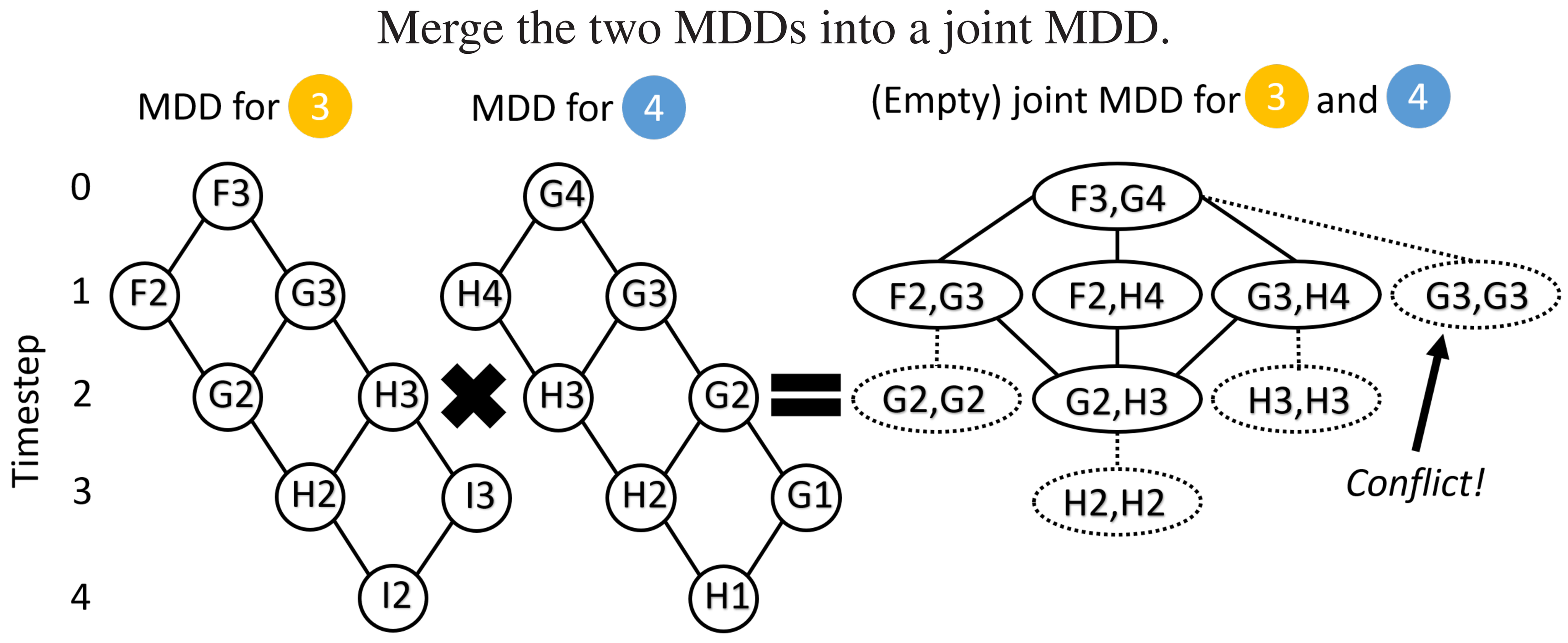
Identify Cardinal Conflicts for CG

Find the same singleton in both MDDs.



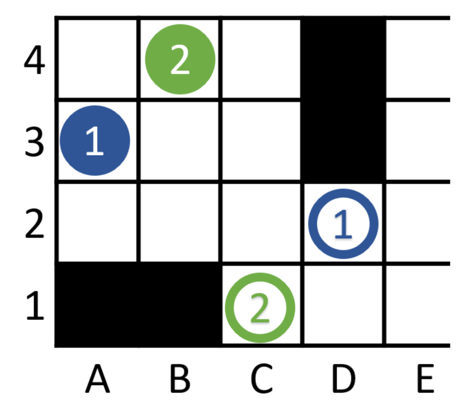
Identify Pairs of Dependent Agents for DG

Merge the two MDDs into a joint MDD.



Calculate Dependency Weights for WDG

Solve a 2-agent pathfinding problem (ignoring all other agents).



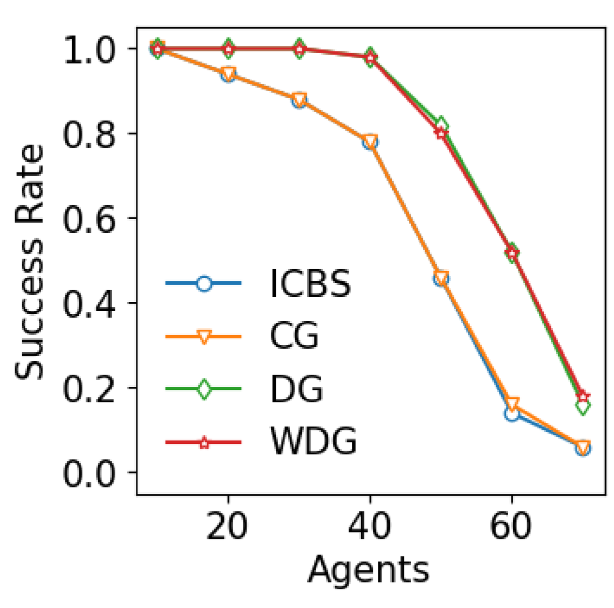
*An MDD for an agent is a directed acyclic graph that consists of all shortest paths for this agent.
**An MDD node is a singleton iff it is the only node at some level of the MDD.

Experimental Results

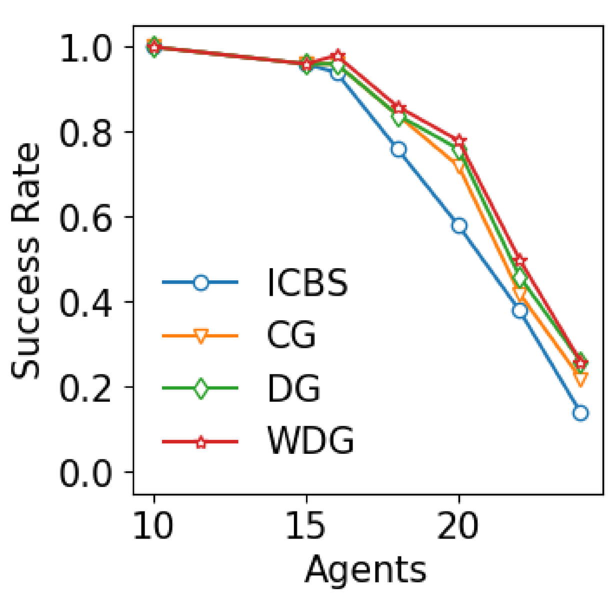
- Empty map: a 20 × 20 4-neighbor grid.
- Dense map: a 20 × 20 4-neighbor grid with 30% random blocked cells.
- Large map: a 192 × 192 4-neighbor benchmark game map from [3].

Empty map					Dense map					Large map				
k	CG	DG	WDG	h^*	k	CG	DG	WDG	h^*	k	CG	DG	WDG	h^*
30	0.2	1.0	1.2	1.7	16	3.9	3.9	11.5	18.6	60	3.6	4.0	6.7	7.6
40	0.5	1.6	2.0	3.3	20	4.7	4.7	14.0	23.2	80	5.7	6.5	10.9	12.2
50	0.5	2.2	2.6	4.7	24	6.5	6.5	18.9	28.5	100	8.6	9.2	15.6	18.0

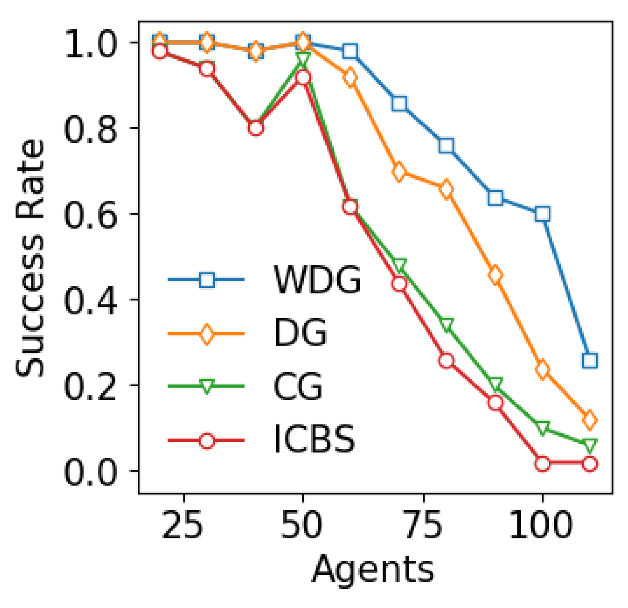
Table 1: h-values at the root node. k represents the number of agents.



(a) Empty map



(b) Dense map



(c) Large map

Figure 1: Success rates (= % solved instances) within 1 min. ICBS always uses zero as h-values.

[1] A. Felner, J. Li, E. Boyarski, H. Ma, L. Cohen, S. Kumar, and S. Koenig. Adding heuristics to conflict-based search for multi-agent path finding. In *ICAPS*, pages 83–87, 2018.
[2] G. Sharon, R. Stern, A. Felner, and N. R. Sturtevant. Conflict-based search for optimal multi-agent pathfinding. *Artificial Intelligence*, 219:40–66, 2015.
[3] N. Sturtevant. Benchmarks for grid-based pathfinding. *Transactions on Computational Intelligence and AI in Games*, 4(2):144–148, 2012.