Proceedings of the Twelfth International
Symposium on Combinatorial Search (SoCS 2019)

Improved Heuristics for Multi-Agent Path Finding with Conflict-Based Search: Preliminary Results*

Jiaoyang Li,[1] Eli Boyarski,[2] Ariel Felner,[2] Hang Ma,[1] Sven Koenig[1]
[1]University of Southern California
[2]Ben-Gurion University
jiaoyanl@usc.edu, eli.boyarski@gmail.com, felner@bgu.ac.il {hangma,skoenig}@usc.edu

Abstract

Conflict-Based Search (CBS) and its enhancements are among the strongest algorithms for Multi-Agent Path Finding. Recent work introduced an admissible heuristic to guide the high-level search of CBS. In this work, we introduce two new admissible heuristics 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.

Introduction

The *Multi-Agent Path Finding* (MAPF) problem is specified by an undirected graph $G = (V, E)$ and a set of k agents $\{a_1 \dots a_k\}$, where agent a_i has start vertex $s_i \in V$ and goal vertex $g_i \in V$. At each discretized timestep, an agent can either *move* to an adjacent vertex or *wait* at its current vertex. Both move and wait actions have unit cost unless it terminally waits at its goal vertex, which has zero cost. A *conflict* happens when two agents occupy the same vertex or traverse the same edge in opposite directions at the same timestep. The objective is to find a set of conflict-free paths which move all agents from their start vertices to their goal vertices while minimizing the sum of the costs of these paths.

CBS is a popular optimal MAPF algorithm which resolves conflicts by adding constraints at a high level and computing paths consistent with these constraints at a low level. CBSH introduces an admissible heuristic (called here CG) for the high-level search of CBS by reasoning about a special type of conflicts in the current solution (i.e, the paths in the current high-level node). In this paper, we introduce two new admissible heuristics, DG and WDG, by considering potential conflicts in the future solutions (i.e, the paths in the descendant high-level nodes) and reasoning about the pairwise dependency between agents. Empirically, the runtime overhead of the new heuristics is reasonable, and WDG improves the success rate and runtime of CBS significantly compared to CBS with CG.

Conflict-Based Search(CBS)

Conflict-Based Search (CBS) (Sharon et al. 2015) has two levels. The high level of CBS searches the binary *constraint tree* (CT) in a best-first manner according to the costs of the CT nodes. Each CT node N contains: (1) a set of constraints $N.constraints$, where each constraint prohibits an agent from occupying a vertex or an edge at a timestep; (2) a solution $N.solution$, which consists of a set of k cost-minimal paths, one for each agent, that satisfy $N.constraints$, and (3) a cost $N.cost$, that is equal to the sum of the costs of the paths in $N.solution$. The root CT node contains an empty set of constraints. When CBS chooses a CT node N for expansion, it checks for conflicts in $N.solution$. If there are none, CBS terminates and returns $N.solution$. Otherwise, CBS chooses one of the conflicts and resolves it by *splitting* N into two child CT nodes. In each child CT node, one agent from the conflict is prohibited from using the contested vertex or edge by way of an additional constraint. The path of this agent no longer satisfies the constraints of the child CT node and must be replanned by a low-level search. All other paths remain unchanged. With two child CT nodes per conflict, CBS guarantees optimality by exploring both ways of resolving each conflict.

The CG Heuristic

CBSH (Felner et al. 2018) speeds up the high-level search of CBS through the addition of an admissible heuristic. The idea is simple: If $N.solution$ contains one *cardinal conflict* (i.e., a conflict where all cost-minimal paths of two conflicting agents traverse the conflicting vertex or edge at the conflicting timestep), then an h-value of 1 is admissible for N because the cost of any of its descendant CT nodes with a conflict-free solution is at least $N.cost + 1$. If $N.solution$ contains multiple cardinal conflicts, CBSH builds a *conflict graph*, whose vertices represent agents and edges represent cardinal conflicts in $N.solution$. The cost of the path of at least one agent from each cardinal conflict has to increase by at least 1. Thus, the size of a *minimum vertex cover* (MVC) of the conflict graph (i.e., a set of vertices such that each edge is incident on at least one vertex in the set) is an ad-

*This paper is a short version of (Li et al. 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.

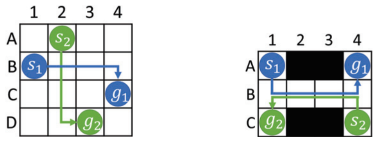


Figure 1: Examples of MAPF instances on 4-neighbor grids.

missible h-value for N. We refer to this heuristic as the CG heuristic.

The DG Heuristic

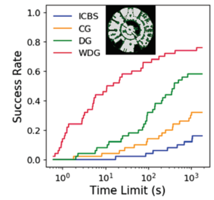
The CG heuristic only considers cardinal conflicts in $N.solution$. We improve it by also considering conflicts in future solutions, i.e., solutions of N's descendant CT nodes. For example, in Figure 1(left), if CBS resolves the non-cardinal conflict at B2 at timestep 1 by adding a constraint to one of the agents, a new conflict will occur no matter what new cost-minimal path the agent picks. In fact, any two cost-minimal paths of the two agents conflict in one of the 4 cells in the middle (B2,B3,C2,C3). Hence, an h-value of 1 is admissible here. This is not captured by CG because the conflicts are initially non-cardinal. Inspired by this example, we generalize the conflict graph described above to a pairwise dependency graph whose edges reflect that all cost-minimal paths of the corresponding two agents have conflicts.

Formally, we define a *pairwise dependency graph* $G_D = (V_D, E_D)$ for each CT node N. Each agent a_i induces a vertex $v_i \in V_D$. An edge (v_i, v_j) is in E_D iff a_i and a_j are *dependent*, i.e., all their cost-minimal paths that satisfy $N.constraints$ have conflicts. Similar to the analysis for the conflict graph, for each edge $(v_i, v_j) \in E_D$, the cost of the path of at least one agent, a_i or a_j, has to increase by at least 1 in order to obtain a conflict-free solution. Hence, the size of the MVC of G_D is an admissible h-value for N. We refer to this heuristic as the DG heuristic. We use the merging-MDD method described in (Sharon et al. 2013) to analyze the dependency between pairs of agents and use the algorithm in (Felner et al. 2018) to determine an optimal MVC.

The WDG Heuristic

Although G_D captures the information about whether the sum of the costs of the paths has to increase for any pair of agents, it does not capture the information about how much the sum of the costs has to increase. In some cases, the sum of the costs for two agents has to increase by more than 1. For instance, in Figure 1(right), the sum of the costs has to increase by 4 because one of the agents must wait 4 timesteps at its start vertex to avoid conflicts with the other agent. Therefore, we introduce the WDG heuristic, which captures not only the pairwise dependency between agents but also the cost that each pair of dependent agents can contribute to the total cost.

We generalize the pairwise dependency graph to a *weighted pairwise dependency graph* G_{WD}. It uses the same vertices and edges as G_D. Every edge $(v_i, v_j) \in E_D$ has a positive integer weight $w_{ij} \geq 1$, which is set to the minimal sum of the costs of the conflict-free paths for agents a_i and a_j minus the sum of the costs of the cost-minimal paths for them at N. We also generalize the MVC to an *edge-weighted minimum vertex cover* (EWMVC), which is an assignment of non-negative integer values $x_1, \ldots, x_k$, one for each vertex, such that $x_i + x_j \geq w_{ij}$ for all $(v_i, v_j) \in E_D$ while minimizing the sum of x_i. x_i can be interpreted as the increase in the cost of the path of a_i. The sum of x_i of the EWMVC of G_{WD} is an admissible h-value for N since, for each edge $(v_i, v_j) \in E_D$, the sum of the costs of the paths of agents a_i and a_j has to increase by at least w_{ij}. We refer to this heuristic as the WDG heuristic. We use a 2-agent MAPF solver (i.e., CBSH with the CG heuristic) to compute the weight on each edge and use a branch and bound algorithm to determine an optimal EWMVC.



	CG	DG	WDG
All 50 instances			
h-value of the CT root node	9.4	10.1	**17.0**
Runtime per CT node (ms)	**16.1**	21.7	21.9
Success rate	0.32	0.58	**0.76**
16 instances solved by all CBSH solvers			
Expanded CT nodes ($\times 1000$)	19.9	6.9	**0.4**
Runtime (s)	318.5	140.6	**6.1**

Figure 2: Results on 50 random instances of 100 agents. The right table uses a time limit of 30 minutes.

Experimental Results

We experiment with ICBS (i.e., CBSH without heuristics) and CBSH with the CG, DG and WDG heuristics. We use a benchmark game map *lak503d* from (Sturtevant 2012). Figure 2(left) shows the success rates (= % instances solved). As the time limit increases, the benefit of using WDG and DG over CG increases as well. In general, it is worth spending a "constant" extra time per CT node to obtain a better heuristic, since a larger heuristic value usually leads to an exponential reduction in the number of CT nodes. Figure 2(right) shows the results with a time limit of 30 minutes. WDG significantly outperforms DG, which in turn significantly outperforms CG in terms of both success rate and runtime. Compared with CG, WDG improves the success rate by a factor of 2 and runs faster by a factor of 50.

References

Felner, A.; Li, J.; Boyarski, E.; Ma, H.; Cohen, L.; Kumar, S.; and Koenig, S. 2018. Adding heuristics to conflict-based search for multi-agent path finding. In *ICAPS*, 83–87.

Li, J.; Boyarski, E.; Felner, A.; Ma., H.; and Koenig, S. 2019. Improved heuristics for multi-agent path finding with conflict-based search. In *IJCAI*.

Sharon, G.; Stern, R.; Goldenberg, M.; and Felner, A. 2013. The increasing cost tree search for optimal multi-agent pathfinding. *Artificial Intelligence* 195:470–495.

Sharon, G.; Stern, R.; Felner, A.; and Sturtevant, N. 2015. Conflict-based search for optimal multi-agent pathfinding. *Artificial Intelligence* 219:40–66.

Sturtevant, N. 2012. Benchmarks for grid-based pathfinding. *Transactions on Computational Intelligence and AI in Games* 4(2):144 – 148.