Further Improved Heuristics for Conflict-Based Search

Eli Boyarski[1], Ariel Felner[1], Pierre Le Bodic[2], Daniel Harabor[2], Peter J. Stuckey[2], Sven Koenig[3]
[1]Ben-Gurion University of the Negev
[2]Monash University
[3]University of Southern California
boyarske@post.bgu.ac.il, felner@bgu.ac.il, {pierre.lebodic, peter.stuckey, daniel.harabor}@monash.edu, skoenig@usc.edu

1 Introduction and Overview

In Multi-Agent Path Finding (MAPF), the aim is to find a set of collision-free paths for a team of agents, each from its start location to its target, minimizing the sum of path costs.

Conflict-Based Search (CBS) (Sharon et al. 2015) is a popular two-level optimal MAPF solver. The low level finds optimal paths for individual agents. If the paths of two agents collide, the high level, via a *split* action, imposes constraints on the agents to avoid the collision. The search space of CBS is therefore a binary *Constraint Tree* (CT), which the algorithm explores in best-first order. Originally, CBS prioritized CT nodes according to the sum of the costs of the paths in them, which can be interpreted as the g-values of the CT nodes. Felner et al. (2018) and Li et al. (2019a) added an admissible heuristic to CBS that estimated the remaining costs, which are the h-values. CT nodes are now prioritized by $f = g + h$. CBS is complete, optimal, and often highly performant; e.g., recent variants (Li et al. 2019a,b,c) can solve MAPF instances with more than 100 agents.

We enhance all known heuristics for CBS by using information about the costs of resolving certain conflicts, with only a small additional computational overhead. Our experiments indicate CBS is more efficient with our heuristics. An early version of this work was published this year (Boyarski et al. 2021). In that version, our heuristics resulted only in a marginal improvement in the number of solved MAPF instances. This work expands on the ideas from that paper and shows more significant results.

1.1 Heuristics for CBS

The Conflict Graph (CG) heuristic (Felner et al. 2018) is the first non-trivial admissible heuristic for CBS. Its h-value is calculated as the size of the minimum vertex cover (MVC) of the *g-cardinal conflict graph* of the current CT node. A *g-cardinal conflict* (Boyarski et al. 2015b, 2021) in a CT node N is one where where the cost (g-value) of *both* child CT nodes that are generated when resolving the conflict is larger than the cost of N. The *g-cardinal conflict graph* contains a vertex for each agent in N, and an edge exists between two vertices iff the paths of the corresponding two agents in N have a g-cardinal conflict.

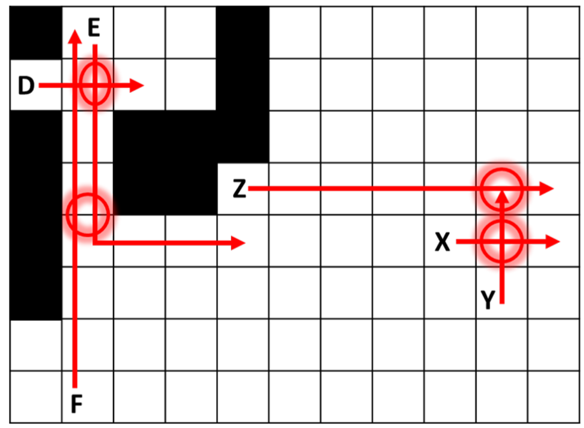

Figure 1: Agents, their paths, and their conflicts

The h-values of the Dependency Graph (DG) (Li et al. 2019a) are calculated as the sizes of the MVC of the *pairwise dependency graphs*, which generalize g-cardinal conflict graphs. A pairwise dependency graph edge exists between two vertices iff the cost of the optimal conflict-free solution for the corresponding two agents is larger than the sum of the costs of the two individual solutions.

The h-values of the (Edge-)Weighted Dependency Graph (WDG) heuristic (Li et al. 2019a) are calculated as follows: *The Weighted Dependency Graph* is constructed by setting the weight of each edge between a pair of agents to the difference between the cost of the optimal conflict-free solution for the corresponding two agents (computed by a subsolver) and the sum of the costs of their paths at the current CT node. Then, the heuristic value $h = \sum_i x_i$ is calculated as a minimal edge-weighted vertex cover of the resulting graph, that is, an assignment of non-negative integers x_i to each vertex i so that $x_i + x_j \geq w_{ij}$ for each edge (v_i, v_j) with cost w_{ij}.

1.2 Adding Near-Vertex Weights to Edges

Boyarski et al. (2021) improved upon each of the three heuristics (CG, DG, and WDG) by creating heuristics that are more informed than their baselines because they use information about the expected g-value increases resulting from resolving a conflict in each potential child CT node. They added *near-vertex weights* to the two ends of each edge

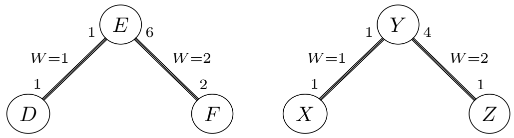

Figure 2: The near-vertex-weighted weighted dependency graph for Figure 1, if corridor and target reasoning are used

in all three graphs that are are used by the heuristics. These weights represent the minimum additional cost needed to resolve the *current* conflict between the two agents by re-planning the path of the agent whose corresponding vertex in the graph is at that end of the edge. They described how these weights are calculated for g-cardinal conflicts and *target conflicts* (Li et al. 2020), which occur when an agent conflicts with an agent that has finished its plan. In the calculation of the improved heuristics, it is also required of every edge that at least one of its vertices be assigned a value no less than the weight at its end of the edge.

The near-vertex weights for an edge (v_i, v_j) representing a basic g-cardinal conflict are $\langle 1, 1\rangle$. The minimal additional cost of the paths to resolve a target conflict is also simple to compute. Assume that agent a_i reaches its target at time step c_i and later has a conflict at time step $t > c_i$ with agent a_j. If the conflict is resolved by forcing agent a_i away from its target, the minimal additional cost of its new path is $(t + 1) - c_i$. Hence, the near-vertex weights for the edge (v_i, v_j) are $\langle (t + 1) - c_i, 1\rangle$.

2 Further Improved Heuristics

A *corridor conflict* (Li et al. 2020) occurs when two agents attempt to traverse a narrow corridor in opposite directions at intersecting time intervals. Resolving a corridor conflict requires one agent to either wait for the other to fully traverse the corridor, or take a possibly-longer alternative path that does not go through the corridor.

Near-vertex weights for corridor conflicts require computing the cost of the shortest path for each agent under the additional range constraint it would receive to resolve the corridor conflict. Let those costs be c'_i and c'_j. We compute c'_i and c'_j using state-time A*, similar to Li et al. (2020). For NVW-CG, we treat all corridor conflicts that, when resolved using corridor reasoning, cause both child CT nodes to have larger costs than their parent, as g-cardinal and add their edges to the conflict graph. This is done even if resolving those conflicts with a regular constraint would not increase the costs of both child CT nodes.

Figure 1 shows an example with 6 agents and their paths on a 4-neighbor grid in a CT node N. All agents have no alternative paths of the same costs. Figure 2 shows the near-vertex-weighted edge-weighted dependency graph for N.

We now examine the expected g-value increases from resolving the conflicts in the example. The weights for (X, Y), (Y, Z), and (D, E) are the same as in (Boyarski et al. 2021). If the (E, F) conflict is resolved in CT node N with corridor reasoning, then the child CT node that constrains agent E will have $\Delta g = 6$, and the child node that constrains agent F will have $\Delta g = 2$. The asymmetry here is due to the fact that the agents are planned to arrive at an entrance to the corridor at different time steps - agent E is closer to the corridor, so it finishes traversing it earlier than agent F.

The NVW-WDG heuristic would give a value of 6 in our example: E is assigned a value of zero, D a value of 1, F a value of 2, and X, Y and Z are assigned values that sum to 3 and satisfy the constraints. The WDG heuristic is only 4 by assigning E and Y values of 2 and D, F, X, and Z values of zero.

CG	N.-CG	h^*	DG	N.-DG	h^*	WDG	N.-WDG	h^*
1.50	1.64	4.64	1.93	1.93	5.22	3.62	**3.78**	4.86

Table 1: Average h-values of the root CT node with each heuristic and with h^* on co-solved instances from scen. 1

#Inst.	CG	N.-CG	DG	N.-DG	WDG	N.-WDG
3,904	3,049	3,042	3,671	**3,712**	3,349	3,361

Table 2: Number of solved MAPF instances with the CG, NVW-CG, DG, NVW-DG, WDG, and NVW-WDG heuristics for instances from scenario 1

3 Experimental Results

We experiment on the MAPF benchmarks (Stern et al. 2019), under a time limit of 60 seconds. We use CBS with bypassing conflicts (Boyarski et al. 2015a) and target and corridor reasoning (Li et al. 2020) coupled with the following heuristics: (1) CG, (2) DG, (3) WDG, (4) NVW-CG, (5) NVW-DG, and (6) NVW-WDG. We use scenario 1 out of the 25 scenarios of the benchmark. The MAPF solver used by WDG for the 2-agent subproblems is CBS with the same configuration, except that it uses the CG heuristic and rectangle reasoning (Li et al. 2019c), a technique that further speeds up CBS. Previous evaluations of heuristics for CBS have been performed on small numbers of instances. Here we examine almost 4,000 MAPF instances, of which 3,838 are successfully solved with at least one of the six heuristics.

Table 1 shows the average h-values of the root CT node on instances solved by CBS with each of the 6 heuristics. Each average is computed over *co-solved* instances of the corresponding pair of solvers: instances solved both by CBS with CG and CBS with NVW-CG, DG and NVW-DG, and WDG and NVW-WDG. To the right of each pair of columns, the average optimal h-value (h^*) over the same set of instances is given. The table shows that the average h-value of the root CT node is improved with NVW-CG and NVW-WDG.

Table 2 shows the number of instances that were solved successfully by each solver. While Table 1 shows NVW-CG and NVW-WDG improve on their baselines in terms of the average h-values, this does not translate into better success rate in the case of NVW-CG. Interestingly, CBS with NVW-DG, and not the stronger NVW-WDG, has the best success rate. With the short time limit of 60 seconds, the overhead of calculating stronger heuristics does not sufficiently pay off.

References

Boyarski, E.; Felner, A.; Le Bodic, P.; Harabor, D.; Stuckey, P. J.; and Koenig, S. 2021. f-Aware Conflict Prioritization & Improved Heuristics For Conflict-Based Search. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI-2021)*.

Boyarski, E.; Felner, A.; Sharon, G.; and Stern, R. 2015a. Don't Split, Try To Work It Out: Bypassing Conflicts in Multi-Agent Pathfinding. In *Proceedings of the International Conference on Automated Planning and Scheduling (ICAPS-2015)*, 47–51. ISBN 9781577357315. ISSN 23340843.

Boyarski, E.; Felner, A.; Stern, R.; Sharon, G.; Tolpin, D.; Betzalel, O.; and Shimony, E. S. 2015b. ICBS: Improved Conflict-Based Search Algorithm for Multi-Agent Pathfinding. In *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI-2015)*, 740–746. ISBN 9781577357384. ISSN 10450823.

Felner, A.; Li, J.; Boyarski, E.; Ma, H.; Cohen, L.; Kumar, T. K. S.; and Koenig, S. 2018. Adding Heuristics to Conflict-Based Search for Multi-Agent Path Finding. In *Proceedings of the International Conference on Automated Planning and Scheduling (ICAPS-2018)*, 83–87.

Li, J.; Felner, A.; Boyarski, E.; Ma, H.; and Koenig, S. 2019a. Improved Heuristics for Multi-Agent Path Finding with Conflict-Based Search. In *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI-2019)*, 442–449. doi:10.24963/ijcai.2019/63.

Li, J.; Gange, G.; Harabor, D.; Stuckey, P. J.; Ma, H.; and Koenig, S. 2020. New Techniques for Pairwise Symmetry Breaking in Multi-Agent Path Finding. In *Proceedings International Conference on Automated Planning and Scheduling (ICAPS-2020)*, 193–201. ISSN 23340843.

Li, J.; Harabor, D.; Stuckey, P. J.; Ma, H.; and Koenig, S. 2019b. Disjoint Splitting for Multi-Agent Path Finding with Conflict-Based Search. In *Proceedings of the International Conference on Automated Planning and Scheduling (ICAPS-2019)*, 279–283.

Li, J.; Harabor, D.; Stuckey, P. J.; Ma, H.; and Koenig, S. 2019c. Symmetry-Breaking Constraints for Grid-Based Multi-Agent Path Finding. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI-2019)*, 6087–6095. ISSN 2159-5399. doi:10.1609/aaai.v33i01.33016087.

Sharon, G.; Stern, R.; Felner, A.; and Sturtevant, N. R. 2015. Conflict-Based Search for Optimal Multi-Agent Pathfinding. *Artificial Intelligence* 219: 40–66.

Stern, R.; Sturtevant, N. R.; Felner, A.; Koenig, S.; Ma, H.; Walker, T. T.; Li, J.; Atzmon, D.; Cohen, L.; Kumar, T. K. S.; Barták, R.; and Boyarski, E. 2019. Multi-Agent Pathfinding: Definitions, Variants, and Benchmarks. In *Proceedings of the Annual Symposium on Combinatorial Search (SoCS-2019)*, 151–159.