The Thirteenth International Symposium
on Combinatorial Search (SoCS 2020)

f-Cardinal Conflicts in Conflict-Based Search

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

1 Introduction and Overview

Multi-agent Path Finding (MAPF) is a coordination problem where the aim is to find a set of collision-free paths for a team of agents, each from its start location to its goal.

Conflict-based Search (CBS) (Sharon et al. 2015) is a two-level optimal MAPF solver which is popular and successful. The low level finds optimal paths for the individual agents. If the paths include collisions, the high level, via a split action, imposes constraints on the agents to avoid these collisions. The search space of CBS is therefore a binary *Conflict Tree* (CT) which the algorithm explores in best-first order. CBS is complete, optimal and often highly performant; e.g., recent variants can solve MAPF problems with > 100 agents. A detailed description of CBS and other algorithms appear in the survey by Felner et al. (2017).

Boyarski et al. (2015) showed that choosing the right conflict to resolve can greatly speed up the search by decreasing the size of the high-level search tree (CT). They recommended choosing *cardinal conflicts*, if any are found in the CT node, otherwise *semi-cardinal conflicts* should be chosen, and finally *non-cardinal conflicts*. When cardinal conflicts are resolved, the cost of the two child nodes that are generated is higher than the cost of their parent.

Later, Felner et al. (2018) and Li et al. (2019) added heuristics to CBS. Nodes of the CT are now prioritized based on the sum f of their cost g and their heuristic value h. Adding such heuristics adds more dimensions to the above prioritization. Often when a cardinal conflict is resolved, while the g-value (cost) of the resulting child nodes increases by 1 relative to their parent, their heuristic estimate h decreases by 1 at the same time. This diminishes the effectiveness of preferring cardinal conflicts in a CT node.

We propose an enhanced categorization function, based on the Δf and the Δg for the child nodes which are generated when the conflict is resolved. In particular, we add the notion of f-*cardinal* conflicts. Resolving f-cardinal conflicts generates child nodes with an increased f-value relative to their parent. Next, we propose two methods for identifying f-cardinal conflicts and provide a new prioritization scheme where f-cardinal conflicts should be resolved first in a CT node, if they are found, then any g-cardinal nodes, and then non-cardinal nodes. Finally, we demonstrate on standard benchmarks that this scheme significantly increases the effectiveness of modern CBS.

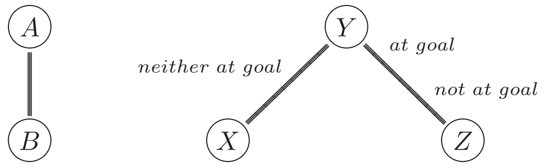

Figure 1: A conflict graph for five agents

2 Different Types of Conflicts

A CT node N is defined by a set of constraints $N.constraints$ on the agents of the problem. It determines for each agent separately a shortest path that does not violate any constraint in $N.constraints$. The node also records its g-value (the sum of the cost of the paths for each agent) as $N.g$. When using a heuristic function, N also stores its h- and f-values. In this paper we assume the h-value is given by the size of the minimum vertex cover (MVC) of the graph of the (g-)cardinal conflicts for the node (Felner et al. 2018).

We expand the conflict types from ICBS (Boyarski et al. 2015) and distinguish between five types of conflicts: given a conflict C in CT node N, and $N1$ and $N2$ the potential child CT nodes that would be generated if N were expanded according to C, we say C is f-**cardinal** if both $N1.f > N.f$ and $N2.f > N.f$, **semi-**f**-cardinal** if both $N1.f > N.f = N2.f$ or $N2.f > N.f = N1.f$, g-**cardinal** if both $N1.g > N.g$ and $N2.g > N.g$, **semi-**g**-cardinal** if either $N1.g > N.g = N2.g$ or $N2.g > N.g = N1.g$, and **non-cardinal** if $N1.g = N2.g = N.g$.

We note that, depending on Δh, a g-cardinal conflict may or may not be also an f-cardinal conflict. Similarly, for semi-g-cardinal conflicts and semi-f-cardinal conflicts.

Figure 1 shows a g-cardinal conflict graph of 5 agents, A, B, X, Y, and Z in a CT node N. Agent Z is planned to pass through agent Y's goal location after Y is planned to reach it. The conflict between Y and X, however, occurs *before* Y is planned to reach its goal. The size of the MVC of the g-cardinal conflict graph, and hence the h-value, is 2. We next examine different conflicts on this graph.

The conflict between agents A and B is g-cardinal. Resolving it will generate child nodes with $\Delta g = 1$ and the size of the MVC will decrease by 1 (unless new cardinal conflicts are caused by the new paths of A and B). Thus, based only on information from this conflict graph, the child nodes will have $\Delta h = -1$ and $\Delta f = 0$.

The conflict between agents X and Y is semi-f-cardinal. If it is resolved in CT node N, the child node that constrains agent X (to avoid the conflict between X and Y) will still have a cardinal conflict graph with a MVC of size (at least) 2, yielding $\Delta h = 0$ and $\Delta f = 1$. By contrast, the child node that constrains Y may have $\Delta h = -1$ (if the Y-Z conflict is resolved by the new path too) and $\Delta f = 0$.

The conflict between agents Y and Z is f-cardinal (recall that Y is at its goal). If it is resolved in CT node N, the child node that constrains Y will have $\Delta g \geq 2$, because Y is forced away from its goal. It will have $\Delta h \geq -1$ and $\Delta f \geq 1$. The child node that constrains Z will have $\Delta h \geq 0$ and $\Delta f = 1$ (similarly to the child that constrains X in the conflict between X and Y as shown earlier).

3 f-Cardinal Conflicts

The identification of f-cardinal conflicts depends on the high-level heuristic. Here, we assume the basic MVC heuristic is used. These methods can be adapted for other high-level heuristics, such as those proposed in (Li et al. 2019).

When CBS resolves an *at-goal conflict* in CT node N, in one of the child nodes a vertex constraint on an agent's goal location is added at time step t' which is *later* than t when the agent is planned to reach its goal. Such a constraint will incur a cost increase (Δg) of at least 2, because the earliest time step the agent's plan can end in is now $t' + 1 \geq t + 2$.

As a result, at-goal conflicts are either g-cardinal or semi-g-cardinal, because the g-value of at least one child node is guaranteed to increase. Moreover, under a basic MVC high-level heuristic, replanning the path of an agent can decrease a node's h-value by at most 1 (if its vertex was in the MVC), yielding $\Delta g \geq 2$, $\Delta h \geq -1$ and $\Delta f \geq 1$. So, at-goal conflicts are either f-cardinal or semi-f-cardinal. Identifying at-goal conflicts is trivial.

To identify f-cardinal conflicts in a CT node N, we first identify g-cardinal conflicts in the usual way (see (Boyarski et al. 2015)). Then, we construct the (g-)cardinal conflict graph for N and compute N's h-value to be the size of its MVC (see (Felner et al. 2018)). Finally, we iterate over all agents that have edges in the (g-)cardinal conflict graph: for each such agent A, we temporarily remove all of its edges and compute the size of the MVC of the remaining graph. This simulates the (g-)cardinal conflict graph and h-value of a child node of N that constrains A according to one of its g-cardinal conflicts, in the most optimistic case that agent A's new path causes no new g-cardinal conflicts and avoids *all* of A's current conflicts. If the size of the MVC remains unchanged, all g-cardinal conflicts that agent A participates in are semi-f-cardinal (like agent X and the X-Y conflict in the example), and all g-cardinal conflicts that A participates in where the other conflicting agent is at its goal are f-cardinal (like agent Z and the Y-Z conflict). Of course, if the size of the MVC decreases, the agent's g-cardinal conflicts remain g-cardinal, but they are not f-cardinal.

When an f-cardinal conflict exists it should be chosen, otherwise a semi-f-cardinal conflict should be chosen. When no such conflicts are found, the choice should be as proposed previously (see (Boyarski et al. 2015)): g-cardinal, then semi-g-cardinal and finally non-cardinal conflicts.

	All instances		Hard instances	
Group	CBS	f-cardinal	CBS	f-cardinal
City	12767	**13071**	270	**574**
Empty	9209	**9283**	163	**237**
Games	20745	**21721**	806	**1782**
Mazes	2584	**2728**	117	**261**
Random	11454	**11767**	333	**631**
Rooms	3986	**4231**	149	**394**
Warehouse	17357	**17627**	249	**519**

Table 1: Solved instances for CBS and CBS-f-cardinal. We report all instances and hard instances only (>30s).

4 Experimental Results

We experimented on the MAPF benchmarks (Stern et al. 2019) and ran a baseline CBS solver that only prioritizes g-cardinal conflicts against one that prioritizes f-cardinal conflicts, under a timeout of 1 minute. In total, 80581 problem instances were solved by at least one of the solvers. 2464 problem instances were only solved by our improved solver, while only 153 instances were only solved by the baseline.

Table 1 shows the number of instances that were solved successfully by each solver for different map groups. The table shows separate counts for all instances solved, and all hard instances that were solved. We define *hard instances* as instances any of the solvers either failed on or required more than half the allotted time to solve (30 seconds). Prioritizing f-cardinal conflicts is always beneficial, and especially on hard instances, where it more than doubles the number of instances that can be solved. The CBS version that prioritizes f-cardinal conflicts also consistently generates fewer nodes and runs faster.

References

Boyarski, E.; Felner, A.; Stern, R.; Sharon, G.; Tolpin, D.; Betzalel, O.; and Shimony, E. 2015. ICBS: Improved conflict-based search algorithm for multi-agent pathfinding. *IJCAI* 2015:740–746.

Felner, A.; Stern, R.; Shimony, S. E.; Boyarski, E.; Goldenberg, M.; Sharon, G.; Sturtevant, N.; Wagner, G.; and Surynek, P. 2017. Search-based optimal solvers for the multi-agent pathfinding problem: summary and challenges. In *SoCS 2017*, 29–37.

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 *ICAPS 2018*, 83–87.

Li, J.; Felner, A.; Boyarski, E.; Ma, H.; and Koenig, S. 2019. Improved Heuristics for Multi-Agent Path Finding with Conflict-Based Search. In *IJCAI 2019*, 442–449.

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 *SoCS 2019*, 151–159.