

You May Split but You Might Work It Out Later: First Steps toward Merging Nodes in MAPF (Extended Abstract)

Grigorios Mouratidis¹, Bernhard Nebel¹, Sven Koenig²

¹University of Freiburg

²University of California, Irvine

mouratig@cs.uni-freiburg.de, nebel@uni-freiburg.de, svenk@uci.edu

Abstract

CBS is a state-of-the-art MAPF algorithm whose performance has been enhanced over the years by the introduction of heuristics that focus the search and reasoning techniques that identify specific types of conflicts that can be resolved faster. To further improve the efficiency of CBS, we present a novel idea based on constraint-reasoning techniques that merges similar high-level nodes close to the root of the constraint tree while preserving CBS’ optimality. As a result, some of CBS’ duplicate work that occurs when expanding similar high-level nodes is avoided. Our first experimental results using a simple CBS variant (ICBS-h) show a significant reduction in the number of expanded high-level nodes on average.

Introduction and Background

Multi-agent path finding (MAPF) is a problem typically modeled as an undirected graph $G = (V, E)$ where a set of agents $a_i \in A$ have to navigate from their starting positions $s_i : A \rightarrow V$ to their goal positions $g_i : A \rightarrow V$ without colliding with each other.

Conflict-based search (CBS) (Sharon et al. 2012) is one of the state-of-the-art algorithms that solve MAPF problems optimally. It is a two-level search algorithm that searches for independent single-agent paths on the low level and then resolves the emerging collisions (which are formally called conflicts) between the agents on the high level. To resolve a conflict, the high level imposes constraints on the conflicting agents, and then the low level recalculates their paths. This procedure continues until all conflicts have been resolved.

CBS’ high-level performs a search in the conflict state-space by creating a binary tree, called the constraint tree (CT). Each node $N \in CT$ contains the following attributes:

- A set of constraints for all agents,
- a solution, which is a plan that respects the current set of constraints,
- a cost, which is the cost of the solution for the given objective.

CBS’ efficiency has been improved over the years by prioritizing the resolution of some specific types of conflicts (Boyarski et al. 2015), developing heuristics such as

Copyright © 2025, Association for the Advancement of Artificial Intelligence (www.aaai.org). All rights reserved.

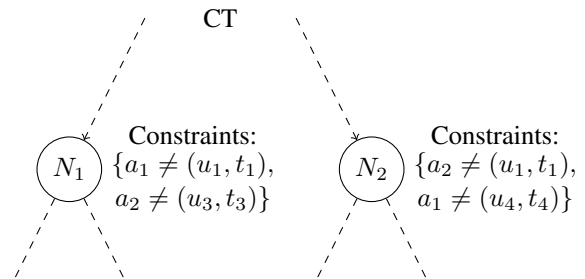


Figure 1: Constraint tree

CG (Felner et al. 2018), DG and WDG (Li et al. 2019a) to guide its high-level search and reasoning techniques (Li et al. 2019b) that identify specific conflicts that cause an exponential number of node expansions and resolve them in one step.

By comparison, constraint-reasoning techniques have received significantly less attention in enhancing CBS. Lazy CBS (Gange, Harabor, and Stuckey 2019) is one of the few approaches that leverage constraint-reasoning techniques to store useful information about subproblems already solved to speed up the search. In the following section, we present a novel idea based on constraint-reasoning techniques that enables the merging of similar high-level CBS nodes.

Node Merging and Constraint Reasoning

Sometimes, high-level nodes in the OPEN list of the constraint tree are very similar regarding the agents’ constraints (and consequently the agents’ paths). As a result, these nodes will often resolve very similar or identical conflicts while CBS’ high-level searches for a solution. Identifying such nodes, especially early in the constraint tree, and exploiting these similarities to avoid duplicate computations can potentially reduce significantly the number of high-level nodes that need to be expanded.

For example, in Figure 1, nodes N_1 and N_2 in the OPEN list of the constraint tree would need to be independently expanded by CBS’ high-level while searching for an optimal solution. Without any specific context, these two nodes seem to have nothing in common. However, this perception changes if we give some extra information about the con-

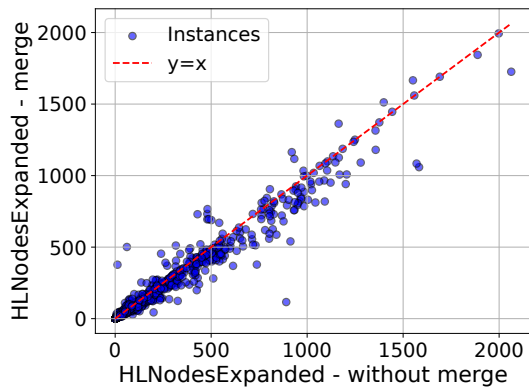


Figure 2: Comparison of high-level nodes expanded in ICBS-h with and without merging for random-32-32-20

straints and apply constraint reasoning.

Single constraints in MAPF (e.g. $a_i \neq (u, t)$) often imply a series of other constraints that are not explicitly stated. Such information can be exploited to prune nodes or even merge very similar nodes. Let’s assume, for example, that in Figure 1 the following two propositions hold:

- $a_1 \neq (u_1, t_1)$ implies $a_1 \neq (u_4, t_4)$
- $a_2 \neq (u_1, t_1)$ implies $a_2 \neq (u_3, t_3)$

This extra information shows that nodes N_1 and N_2 are very similar and could actually be merged into a more general high-level node that preserves CBS’ optimality. This can be done by relaxing some of the constraints of the nodes. Specifically, the constraint $a_1 \neq (u_1, t_1)$ of node N_1 can be relaxed to $a_1 \neq (u_4, t_4)$ and the constraint $a_2 \neq (u_1, t_1)$ of node N_2 can be relaxed to $a_2 \neq (u_3, t_3)$. After applying these relaxations, the two nodes have exactly the same constraints and can be safely merged.

By relaxing the constraints of a high-level node, the number of possible solutions of the node can only grow larger, therefore, it can be proven that CBS’ optimality is preserved. Furthermore, if the subtrees under the two initial nodes need to resolve many identical conflicts, merging will avoid this duplicate work and potentially reduce high-level node expansions.

Experimental Evaluation

To showcase the potential of merging similar nodes, we initially decided to run experiments with a relatively simple CBS variant (ICBS-h) that does not include all the state-of-the-art heuristics.

In our first experiments, we used the random-32-32-20 map from `movingai.com`, and the agents’ starting and goal positions were randomly selected except for the positions of two agents that were engineered so that a merging opportunity always arose at the beginning of the constraint tree. Finally, we varied the agents from 10 to 60 and created 50 instances per number of agents. The number of high-level node expansions with and without merging for each instance is shown in Figure 2 and seems promising.

Conclusion and Future Work

We presented a novel idea about merging high-level nodes in CBS based on constraint-reasoning techniques and showed empirically that such an approach can, in some cases, reduce the number of high-level node expansions.

In future work, we plan to benchmark the effect of node merging on more sophisticated CBS variants that use more informed heuristics like DG and WDG and symmetry reasoning techniques. Furthermore, we want to investigate theoretically and empirically the effect of merging similar high-level nodes at different depths of the constraint tree.

References

- Boyarski, E.; Felner, A.; Stern, R.; Sharon, G.; Tolpin, D.; Betzalel, O.; and Shimony, S. E. 2015. ICBS: Improved Conflict-Based Search Algorithm for Multi-Agent Pathfinding. In *Proceedings of the Twenty-Fourth International Joint Conference on Artificial Intelligence, IJCAI 2015, Buenos Aires, Argentina, July 25-31, 2015*, 740–746. AAAI Press.
- 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 Twenty-Eighth International Conference on Automated Planning and Scheduling, ICAPS 2018, Delft, The Netherlands, June 24-29, 2018*, 83–87. AAAI Press.
- Gange, G.; Harabor, D.; and Stuckey, P. J. 2019. Lazy CBS: Implicit Conflict-Based Search Using Lazy Clause Generation. In *Proceedings of the Twenty-Ninth International Conference on Automated Planning and Scheduling, ICAPS 2019, Berkeley, CA, USA, July 11-15, 2019*, 155–162. AAAI Press.
- 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 Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI 2019, Macao, China, August 10-16, 2019*, 442–449. ijcai.org.
- Li, J.; Harabor, D.; Stuckey, P. J.; Ma, H.; and Koenig, S. 2019b. Symmetry-Breaking Constraints for Grid-Based Multi-Agent Path Finding. In *The Thirty-Third AAAI Conference on Artificial Intelligence, AAAI 2019, The Thirty-First Innovative Applications of Artificial Intelligence Conference, IAAI 2019, The Ninth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2019, Honolulu, Hawaii, USA, January 27 - February 1, 2019*, 6087–6095. AAAI Press.
- Sharon, G.; Stern, R.; Felner, A.; and Sturtevant, N. R. 2012. Conflict-Based Search For Optimal Multi-Agent Path Finding. In *Proceedings of the Twenty-Sixth AAAI Conference on Artificial Intelligence, July 22-26, 2012, Toronto, Ontario, Canada*, 563–569. AAAI Press.