# Non-Refined Abstractions in Counterexample Guided Abstraction Refinement for Multi-Agent Path Finding (Extended Abstract)

**Pavel Surynek***

Faculty of Information Technology, Czech Technical University in Prague
Thákurova 9, 160 00 Praha 6, Czechia
pavel.surynek@fit.cvut.cz

**Abstract**

Counterexample guided abstraction refinement (CEGAR) represents a powerful symbolic technique for various tasks such as model checking and reachability analysis. Recently, CEGAR combined with Boolean satisfiability (SAT) has been applied for multi-agent path finding (MAPF), a problem where the task is to navigate agents from their start positions to given individual goal positions so that agents do not collide with each other. The recent CEGAR approach used the initial abstraction of the MAPF problem where collisions between agents were omitted and were eliminated in subsequent abstraction refinements. We propose in this work a novel CEGAR-style solver for MAPF based on SAT in which some abstractions are deliberately left non-refined. This adds the necessity to post-process the answers obtained from the underlying SAT solver as these answers slightly differ from the correct MAPF solutions. Non-refining however yields order-of-magnitude smaller SAT encodings than those of the previous approach and speeds up the overall solving process.

## Introduction

Multi-agent path finding (MAPF) (Silver 2005) is a task of finding non-conflicting paths for $k \in \mathbb{N}$ agents $A = \{a_1, a_2, ..., a_k\}$ that move in an undirected graph $G = (V, E)$ across its edges such that each agent reaches its goal vertex from the given start vertex via its path. Starting configuration of agents is defined by a simple assignment $s : A \rightarrow V$ and the goal configuration is defined by a simple assignment $g : A \rightarrow V$. A conflict between agents is usually defined as simultaneous occupancy of the same vertex by two or more agents or as a traversal of an edge by agents in opposite directions.

We address MAPF from the perspective of compilation techniques that represent a major alternative to search-based solvers (Sharon et al. 2015) for MAPF. Compilation-based solvers reduce the input MAPF instance to an instance in a different well established formalism for which an efficient solver exists. Such formalisms are for example *constraint programming* (CSP), *Boolean satisfiability* (SAT), or *mixed integer linear programming* (MILP).

*This work has been supported by GAČR - the Czech Science Foundation, grant registration number 22-31346S.


The basic compilation scheme (see Figure 1) for sum-of-costs optimal MAPF solving has been introduced by the MDD-SAT (Surynek et al. 2016) solver that uses so called *complete models* to compile MAPF instances into SAT. The target Boolean formula of the complete model is satisfiable if and only if the input MAPF has a solution of a specified sum-of-costs. The complete model as introduced in MDD-SAT consists of three group of constraints:

- **Agent propagation** constraints - these constraints ensure that if an agent appears in vertex $v$ at time step $t$ then the agent appears in some neighbor of $v$ (including $v$) at time step $t + 1$. The side effect of these constraints is that the agent never disappears. Cost calculation and bounding is done together with agent propagation.
- **Path consistency** constraints - these constrains ensure that agents move along proper paths, that is, agents do not clone themselves and do not appear spontaneously.
- **Conflict elimination** constraints - to ensure that agents do not conflict with each other according to the MAPF rules (vertex and edge conflict).

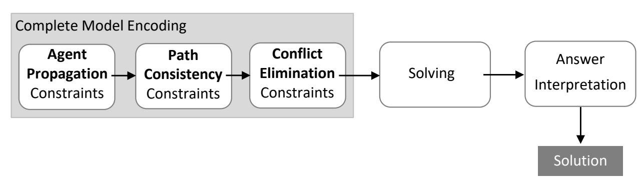


Figure 1: Schematic diagram of the basic MAPF compilation with complete model.

A significant improvement over complete models in problem compilation for MAPF is the introduction of laziness via *incomplete models* where the conflict elimination constraints are omitted for which equivalent solvability no longer holds, but only the implication: if the MAPF instance is solvable then the instance in the target formalism is solvable too (Surynek 2019).

The discrepancy between the original formulation of MAPF and its compiled variant in the target formalism is eliminated by abstraction refinements similarly as it is done in the *counterexample guided abstraction refinement* (CEGAR) (Clarke et al. 2000) approach for model checking (see Figure 2).

## CEGAR for MAPF

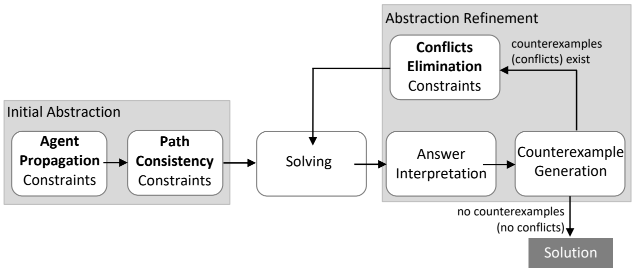


Figure 2: Schematic diagram of counterexample guided abstraction refinement (CEGAR) for MAPF. Conflicts between agents are treated as counterexamples and eliminated in the abstraction refinement loop.

The general CEGAR approach for compilation-based problem solving starts with a so called *initial abstraction* of the problem instance being solved in some target formalism such as SAT or CSP. The initial abstraction do not model the input instance in the full details. However still the initial abstraction is passed to the solver for the target formalism despite the solver is not provided all the details needed to solve it. Then the solver will come with some answer and since it could not take some details into account during the solving phase, the answer must be checked, usually against full details of how the problem instance is defined.

Two cases need to be distinguished at this stage. If the provided answer matches the instance definition then it is returned and the solving process finishes. Otherwise the CEGAR solving process generates *counterexample* that is determined by the mismatch between the provided answer and the requirements the expected answer should satisfy. This mismatch is usually represented by the violation of some constraints that were not expressed in the abstraction. Then the solving process continues with a so called *abstraction refinement* in which the abstraction is augmented to eliminate the counterexample and the solving process continues with the next iteration of (now refined) abstraction solving (see Figure 2).

## Non-Refined Abstractions in MAPF

We are going further in the CEGAR architecture of the MAPF solver. In addition to *conflict elimination* constrains we also omit *path consistency* constraints in the initial abstraction. Moreover we never make any refinement with respect to the omitted *path consistency* constraints - the corresponding abstraction remains **non-refined**.

Omitting the path consistency constraint however leads to solving a different though equisatisfiable problem. Instead of attempting to connect the initial positions of agents with their goals positions via paths, we are now attempting to make the connection via directed acyclic graphs (DAGs). This requires addition of a new polynomial-time post-processing step in the solving process that extracts valid paths from DAGs (see Figures 3 and 4).

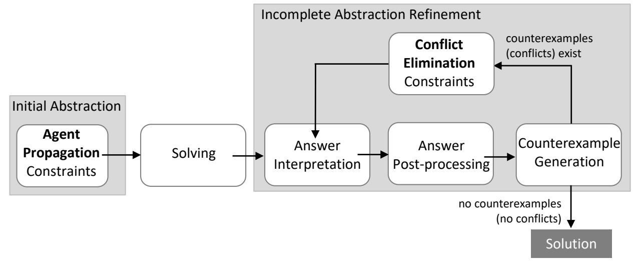


Figure 3: Schematic diagram of CEGAR problem solver for MAPF with non-refined abstractions.

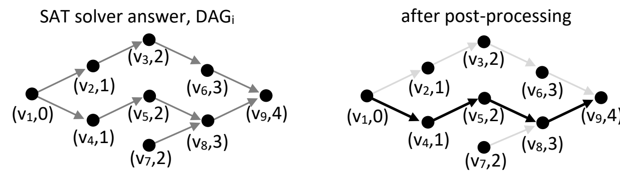


Figure 4: Post-processing step in which path is extracted from DAG answered by the SAT solver.

## Conclusion

Experiments show promising results as non-refining yields order of magnitude smaller formulae and faster solving runtimes than previous CEGAR-inspired SAT-based MAPF solver. We believe that non-refined abstractions in CEGAR-inspired MAPF solving opens new ways in compilation-based approaches for MAPF. Non-refining path consistency constrains is just one example and we believe that other abstractions can be discovered.

## References

Clarke, E. M.; Grumberg, O.; Jha, S.; Lu, Y.; and Veith, H. 2000. Counterexample-Guided Abstraction Refinement. In *Computer Aided Verification, 12th International Conference, CAV 2000, Proceedings*, volume 1855 of *LNCS*, 154–169. Springer.

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

Silver, D. 2005. Cooperative Pathfinding. In *Proceedings of the 1st Artificial Intelligence and Interactive Digital Entertainment Conference*, 117–122. AAAI Press.

Surynek, P. 2019. Unifying Search-based and Compilation-based Approaches to Multi-agent Path Finding through Satisfiability Modulo Theories. In Kraus, S., ed., *Proceedings of the 28th International Joint Conference on Artificial Intelligence, IJCAI 2019*, 1177–1183. ijcai.org.

Surynek, P.; Felner, A.; Stern, R.; and Boyarski, E. 2016. Efficient SAT Approach to Multi-Agent Path Finding Under the Sum of Costs Objective. In *ECAI 2016 - 22nd European Conference on Artificial Intelligence*, volume 285, 810–818. IOS Press.