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

Unifying Search-Based and Compilation-Based Approaches to Multi-Agent Path Finding through Satisfiability Modulo Theories

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

Abstract

We unify search-based and compilation-based approaches to multi-agent path finding (MAPF) through *satisfiability modulo theories* (SMT). The task in MAPF is to navigate agents in an undirected graph to given goal vertices so that they do not collide. We rephrase Conflict-Based Search (CBS) in the terms of SMT. This idea combines MDD-SAT, a SAT-based optimal MAPF solver, at the low level with conflict elimination of CBS at the high level.

Introduction and Background

Multi-agent path finding in graphs (MAPF) (Silver 2005) is a task of navigating agents from their starting vertices to given goals while avoiding collisions. We address optimal solving of MAPF with respect to cumulative objectives.

We contribute by an optimal algorithm that **unifies** two major approaches to solving MAPF optimally: the **search-based** approach represented by Conflict-Based Search (CBS) (Sharon et al. 2015) and the **compilation-based** approach represented by reducing MAPF to propositional satisfiability (SAT) in the MDD-SAT algorithm (Surynek et al. 2016).

Our novel algorithm called SMT-CBS rephrases the ideas of CBS in the terms of *satisfiability modulo theories* (SMT) (Bofill et al. 2012) at the high level. While at the low level it uses the SAT encoding from from MDD-SAT.

Unlike the original CBS that resolves collisions between agents by branching the search, SMT-CBS refines the the propositional model with a disjunctive constraint.

The hypothesis behind the design of SMT-CBS is that in many cases we do not need to add all constraints while still be able to obtain a collision-free solution. Intuitively we expect that such cases will be represented by sparsely occupied instances with large environments.

MAPF consists of an undirected graph $G = (V, E)$ and a set of agents $A = \{a_1, a_2, ..., a_k\}$ such that $|A| < |V|$. Agents are assigned to vertices with at most one agent per vertex that we call a *configuration* and denote $\alpha : A \rightarrow V$. Starting configuration α_0 and goal configuration α_+ are specified.

At each step an agent can either *move* to an adjacent vertex or *wait*. The task is to find a sequence of move/wait actions for each agent a_i denoted $path(a_i)$ that moves the agent from $\alpha_0(a_i)$ to $\alpha_+(a_i)$ so that agents do not *collide*.

We usually search for solutions that minimize cumulative objective functions like the *makespan* (Surynek 2017) or the *sum-of-costs* (Sharon et al. 2013) denoted $\xi = \sum_{i=1}^{k} \xi(path(a_i))$, where $\xi(path(a_i))$ is the number of actions on $path(a_i)$.

Unifying Search and Compilation

CBS (Sharon et al. 2015), a representative of **search-based approach** for optimal MAPF solving, uses the idea of resolving conflicts lazily; that is, a solution of MAPF instance is not searched against the complete set of movement constraints that forbids collisions between agents but with respect to initially empty set of collision forbidding constraints that gradually grows as new collisions appear. The advantage of CBS is that it can find a valid solution before all collision elimination constraints are added.

After finding paths for individual agents, CBS validates these paths for collisions. If they are collision-free then we have a solution. Otherwise, if a collision is detected, say (a_i, a_j, v, t) between agents a_i and a_j in vertex v at time step t , CBS adds constrains to resolve the collision and branches the search. A new constraint preventing a_i from entering v at t is added in one branch and similarly a constraint forbidding a_j in v at t in the other branch.

The major alternative approach to CBS is represented by the **compilation** of MAPF to propositional satisfiability (SAT) (Surynek 2017). The idea is to construct a propositional formula $\mathcal{F}(\xi)$ such that it is satisfiable if and only if a solution of sum-of-costs ξ exists. We say $\mathcal{F}(\xi)$ to be a *complete propositional model* of MAPF if: $\mathcal{F}(\xi)$ is satisfiable $\Leftrightarrow$ Σ has a solution of sum-of-costs ξ.

Being able to construct such formula $\mathcal{F}$ one can obtain optimal MAPF solution by checking satisfiability of $\mathcal{F}(\xi_0)$, $\mathcal{F}(\xi_0 + 1)$, $\mathcal{F}(\xi_0 + 2)$,..., where ξ_0 is a lower bound estimation of the sum-of-costs calculated as the sum of lengths of shortest paths, until the first satisfiable $\mathcal{F}(\xi)$ is met. This is possible due to monotonicity of MAPF solvability with respect to increasing values of common cumulative objectives.

*This work has been supported by the Czech Science Foundation - GAČR (grant registration number 19-17966S).

The advantage of the SAT-based approach is that state-of-the-art SAT solvers can be used for determining satisfiability of $\mathcal{F}(\xi)$ (Audemard and Simon 2009).

A close look at CBS reveals that it operates similarly as problem solving in *satisfiability modulo theories* (SMT) (Bofill et al. 2012). SMT divides the satisfiability problem in some complex theory T into: **(i)** an abstract propositional part that keeps the Boolean structure of the decision problem but omits axioms of T and **(ii)** a simplified decision procedure $DECIDE_T$ that decides a fragment of T restricted on *conjunctive formulae*. The decision problem is solved iteratively: a solution of propositional part obtained from the SAT solver is validated against axioms of T by $DECIDE_T$. If an inconsistency is found the propositional part is refined by a constraint to eliminate the inconsistency.

Using the above observation we rephrased CBS in terms of SMT in an algorithm called SMT-CBS. A paths validation procedure will act as $DECIDE_T$ and will report back a set of collisions found in the current solution. Axioms of T will be represented by the movement rules of MAPF.

The propositional part for SMT-CBS has been taken from the MDD-SAT encoding (Surynek et al. 2016) where we omitted collision avoidance constraints. This results in an *incomplete propositional model* $\mathcal{H}$ where: $\mathcal{H}(\xi)$ is satisfiable $\Leftarrow \Sigma$ has a solution of sum-of-costs ξ. Bulding of $\mathcal{H}(\xi)$ in the SMT-inspired style is shown in Algorithm 1.

Algorithm 1: Part of the SMT-CBS algorithm

```
1  SMT-CBS-Fixed(Σ = (G, A, α0, α+), ξ)
2      H(ξ) ← encode-Basic(ξ, Σ)
3      while TRUE do
4          assignment ← consult-SAT-Solver(H(ξ))
5          if assignment ≠ UNSAT then
6              paths ← extract-Solution(assignment)
7              collisions ← validate(paths)
8              if collisions = ∅ then
9                  return paths
10             for each (a_i, a_j, v, t) ∈ collisions do
11                 H(ξ) ← H(ξ) ∪ {¬X_v^t(a_i) ∨ ¬X_v^t(a_j)}
12         return UNSAT
```

At the high-level SMT-CBS is almost identical to CBS: a solution of incomplete model $\mathcal{H}(\xi)$ is validated against MAPF rules (line 7) and if a collision is detected (line 10), then the model is refined. But in contrast to CBS that branches the high-level search, SMT-CBS adds a disjunctive constraint to eliminate the collision instead (line 11).

Experimental Evaluation

We compared SMT-CBS, CBS, and MDD-SAT implemented in C++ on standard MAPF benchmarks (Sturtevant 2012). We varied the number of agents in MAPF instances to obtain instances of various difficulties. For each number of agents in the MAPF instance we generated 10 instances with random initial and goal configuration of agents. Experiments were run on a Ryzen 7 CPU 3.0 Ghz with 16GB RAM and the timeout of 1000 seconds.

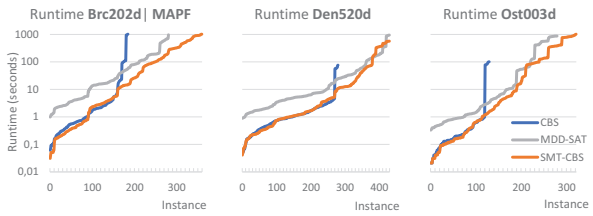

Figure 1: Comparison of CBS, MDD-SAT, and SMT-CBS.

Sorted runtimes obtained in large maps are reported in Figure 1. There is no significant difference between CBS and SMT-CBS in easier cases but MDD-SAT lags behind. The situation changes after going into medium difficulty region where runtimes of CBS go quickly up while SMT-CBS maintains significant advantage (factor 2 to 5) over MDD-SAT. Eventually however the performance of SMT-CBS and MDD-SAT meets in the hard region.

Conclusion

Experiments confirmed our hypothesis that SMT-CBS can produce a solution well before all constraints are added to the encoding which altogether leads to faster solving.

For the future work we plan to further generalize the suggested framework for geometric agents and reasoning in continuous space and time.

References

Audemard, G., and Simon, L. 2009. Predicting learnt clauses quality in modern SAT solvers. In *IJCAI*, 399–404.

Bofill, M.; Palahí, M.; Suy, J.; and Villaret, M. 2012. Solving constraint satisfaction problems with SAT modulo theories. *Constraints* 17(3):273–303.

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

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

Silver, D. 2005. Cooperative pathfinding. In *AIIDE*, 117–122.

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

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*, 810–818.

Surynek, P. 2017. Time-expanded graph-based propositional encodings for makespan-optimal solving of cooperative path finding problems. *Ann. Math. Artif. Intell.* 81(3-4):329–375.