Proceedings of the Fifteenth International Symposium on Combinatorial Search (SoCS 2022)

Sparse Decision Diagrams for SAT-based Compilation of Multi-Agent Path Finding (Extended Abstract)

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

Abstract

Multi-agent path finding (MAPF) represents a task of finding non-colliding paths for agents via which they can navigate from their initial positions to specified goal positions. Contemporary optimal solving algorithms include dedicated search-based methods, that solve the problem directly, and compilation-based algorithms that reduce MAPF to a different formalism for which an efficient solver exists. In this paper, we enhance the existing Boolean satisfiability-based (SAT) algorithm for MAPF via using sparse decision diagrams representing the set of candidate paths for each agent, from which the target Boolean encoding is derived, considering more promising paths before the less promising ones are taken into account. Suggested sparse diagrams lead to a smaller target Boolean formulae that can be constructed and solved faster while optimality guarantees of the approach are kept. Specifically, considering the candidate paths sparsely instead of considering them all makes the SAT-based approach more competitive for MAPF on large maps.

Introduction and Motivation

Multi-agent path planning in graphs (MAPF) represents a fundamental problem in combinatorial motion planning in robotics (Silver 2005; Ryan 2007; Standley 2010; Luna and Bekris 2011; Yu and LaValle 2013). The task is to navigate each agent from the set of agents $A = \{a_1, a_2, ..., a_k\}$ from its initial position to a specified goal position. The environment is modeled as an undirected graph $G = (V, E)$ where vertices represent positions and agents move across edges between vertices. Two requirements make the problem challenging: **(1)** the agents must not collide with each other, that is they can never share a vertex nor can traverse an edge in opposite directions, and **(2)** some objective such as the total number of actions must be optimized (minimized).

We address the MAPF problem from the perspective of compilation-based techniques. *Compilation* is one of the most important techniques used across many computing fields ranging from theory to practice. In the context of problem solving, the compilation approach reduces an input problem instance from its source formalism to a different, usually well established, target formalism for which an efficient solver exists. After obtaining a solution by the solver, it is interpreted back to the input formalism, which altogether constitutes a **reduction-solving-interpretation** loop.

The target formalisms are often combinatorial optimization frameworks like *constraint programming / optimization* (CP) (Dechter 2003), *mixed integer linear programming* (MILP) (Jünger et al. 2010; Rader 2010), Boolean satisfiability (SAT) (Biere et al. 2009), satisfiability modulo theories (SMT) (Barrett and Tinelli 2018), or answer set programming (ASP) (Lifschitz 2019). Employing the advancements in solvers for the target formalisms, often accumulated for decades, in solving the input problem represents the key benefit of problem solving via compilation. However, the way how the input instance is reduced to the target formalism and presented to the solver has a great impact on the efficiency of the reduction-solving-interpretation loop.

Currently, **compilation-based** optimal solvers for MAPF represent a major alternative to **search-based** solvers, that model and solve the problem directly, and often provide more modular and versatile architecture than the search-based solvers while keeping competitive performance. Contemporary compilation-based solvers for MAPF include those based on CP (Gange, Harabor, and Stuckey 2019), MILP (Lam et al. 2019), SAT/SMT (Surynek et al. 2016; Surynek 2019), as well as ASP-based solvers (Erdem et al. 2013).

We focus in this paper on SAT-based solvers for MAPF. Our contribution consists in a new technique for encoding the MAPF instance as Boolean formulae via **sparsification** of the set of candidate paths for each agent. The existing encoding introduces a Boolean decision variable for each vertex and edge from a candidate path for a given agent. The interpretation of a decision variable is that it is set to $TRUE$ if and only if the agent traverses the corresponding vertex or edge at the corresponding time step.

The novel encoding is integrated into a modified SMT-CBS (Surynek 2019), an optimal MAPF solving algorithm that uses *lazy compilation* scheme. Since Boolean formulae, to which the input MAPF instance is reduced in SMT-CBS, are derived from the set of candidate paths, the effect of sparsification of the set is twofold: **(1)** it leads to smaller target Boolean formulae that can be constructed faster and **(2)** the satisfiability of formulae can be decided by the SAT solver faster, altogether improving the reduction-solving-interpretation loop in SMT-CBS. At the same time, optimality guarantees in the modified SMT-CBS are kept via maintaining easily verifiable property of the sparse set of candidate paths.

Sparse Multi-valued Decision Diagrams

SMT-CBS expands the underlying graph G over time so that it can represent all possible paths of certain length. These path are represented using multi-valued decision diagrams (MDD) (Andersen et al. 2007) (Figure 1). However, for instances that takes place on large graphs, the size of MDDs that is directly reflected in the size of the target Boolean formula could be prohibitive.

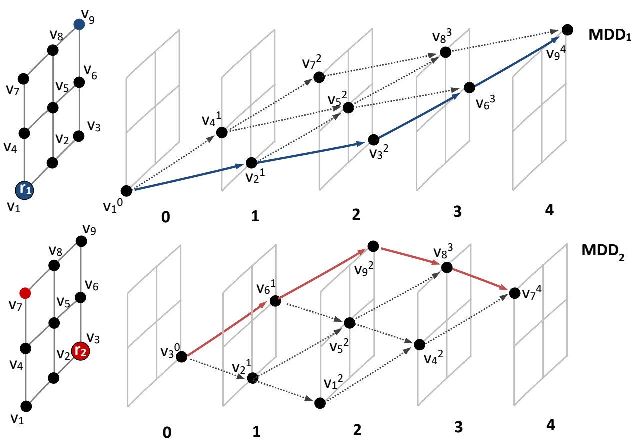

Figure 1: An example of MDDs representing all candidate paths of cost 4 for agents a_1 and a_2.

Therefore we suggest to simplify MDDs and the formula by reducing the number of paths that are represented. To do this, we suggest to use sparse sets of candidate paths for each agent that satisfy given sum-of-costs and makespan bounds. That is, instead of considering all such paths as done in the standard time expansion using MDDs, we consider only a relevant subset of them. Similar concept called the *pool of paths* has been used in the context of MILP-based compilation for MAPF (Gange, Harabor, and Stuckey 2019) which however does not explicitly focus on sparsification.

We integrated the sparse paths set reasoning into the SMT-CBS framework, designing a new algorithm we called **Sparse-SMT-CBS**. Sparse-SMT-CBS uses the identical sum-of-costs and makespan bounds increasing scheme to find the optimum as SMT-CBS at the high-level. Each iteration at the high-level resolves a question whether there exists a solution to the input MAPF such that it fits in the current sum-of-costs SoC and makespan μ bounds. This question is compiled as a series of Boolean formulae and consulted with the SAT solver. Conflict elimination is encoded lazily into the formulae, starting with no conflict elimination constraint followed by refinements as conflicts are being discovered (Clarke 2003).

The low-level in Sparse-SMT-CBS is different from SMT-CBS. In both algorithms, we try to find a non-conflicting set of paths satisfying SoC and μ, but the set of candidate paths from which SMT-CBS selects is fixed in advance in MDD, while Spare-SMT-CBS starts with a minimal set of candidate paths and each time a new conflict is discovered the set of candidate paths is extended to reflect the new conflict. To keep soundness and optimality of the algorithm, the sparse set of candidate paths must be selected according to the following conditions.

Definition 1 (path feasibility). *Let C be a set of conflicts of the form (a_i, v, t) forbidding an agent $a_i \in A$ to reside in $v \in V$ at time step t. We say a path $path(a_i) = [p_0, p_1, ..., p_m]$ for agent a_i to be* feasible *with respect to C if and only if $(a_i, p_t, t) \notin C$ for $\forall t \in \{0, 1, ..., m\}$.*

In other words, a feasible path avoids all conflicts from C. The sparse set of paths for agent a_i denoted $\Pi(a_i)$ with respect to a set of conflicts C and makespan μ must satisfy the following property:

- **(P1)** $\Pi(a_i)$ contains at least one feasible path with respect to each subset of conflicts $C' \subseteq C$ if such path exists.

(P1) ensures that after discovering a new conflict at least one new path avoiding the new conflict for each possible combination of previous conflicts is added to Π provided that such path exists for given makespan bound μ.

We extend the usage of MDDs on sparse sets of candidate path and denote them as *sparse MDDs* (SMDDs) (Figure 2).

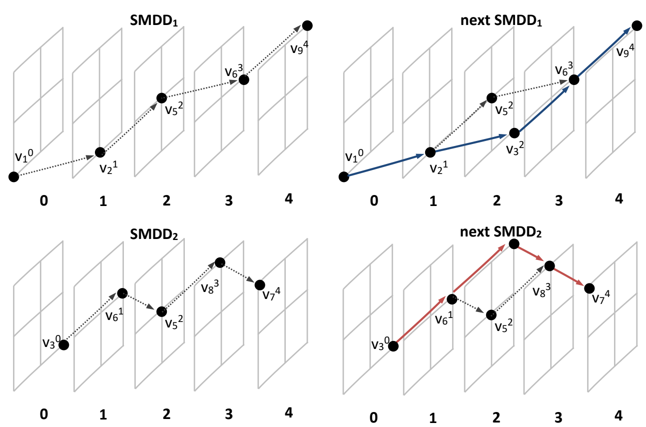

Figure 2: An example of sparse MDDs (SMDDs) for agents a_1 and a_2. The first iteration yields a conflict in v_5 at timestep 2 between the agents which can be avoided via newly represented paths in the next iteration.

Conclusion

According to our experiments on a number of benchmarks, the new algorithm called Sparse-SMT-CBS performs significantly better than SMT-CBS especially on MAPFs with large graphs and outperforms the basic search-based CBS.

Another important advantage of sparsification is that it provides a room for integrating domain specific heuristics via giving a preference to some paths being selected into the set of candidate paths for an agent.

Acknowledgments

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

References

Andersen, H. R.; Hadzic, T.; Hooker, J. N.; and Tiedemann, P. 2007. A Constraint Store Based on Multivalued Decision Diagrams. In *Principles and Practice of Constraint Programming - CP 2007, Proceedings*, volume 4741 of *Lecture Notes in Computer Science*, 118–132. Springer.

Barrett, C. W.; and Tinelli, C. 2018. Satisfiability Modulo Theories. In Clarke, E. M.; Henzinger, T. A.; Veith, H.; and Bloem, R., eds., *Handbook of Model Checking*, 305–343. Springer.

Biere, A.; Heule, M.; van Maaren, H.; and Walsh, T., eds. 2009. *Handbook of Satisfiability*, volume 185 of *Frontiers in Artificial Intelligence and Applications*. IOS Press. ISBN 978-1-58603-929-5.

Clarke, E. M. 2003. SAT-Based Counterexample Guided Abstraction Refinement in Model Checking. In *Automated Deduction - CADE-19, 19th International Conference on Automated Deduction Miami Beach, 2003, Proceedings*, volume 2741 of *Lecture Notes in Computer Science*, 1. Springer.

Dechter, R. 2003. *Constraint processing*. Elsevier Morgan Kaufmann.

Erdem, E.; Kisa, D. G.; Öztok, U.; and Schüller, P. 2013. A General Formal Framework for Pathfinding Problems with Multiple Agents. In *Proceedings of the Twenty-Seventh AAAI Conference on Artificial Intelligence, 2013*. 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 2018*, 155–162. AAAI Press.

Jünger, M.; Liebling, T. M.; Naddef, D.; Nemhauser, G. L.; Pulleyblank, W. R.; Reinelt, G.; Rinaldi, G.; and Wolsey, L. A., eds. 2010. *50 Years of Integer Programming 1958-2008 - From the Early Years to the State-of-the-Art*. Springer.

Lam, E.; Bodic, P. L.; Harabor, D. D.; and Stuckey, P. J. 2019. Branch-and-Cut-and-Price for Multi-Agent Pathfinding. In Kraus, S., ed., *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI 2019, Macao, China, August 10-16, 2019*, 1289–1296. ijcai.org.

Lifschitz, V. 2019. *Answer Set Programming*. Springer.

Luna, R.; and Bekris, K. E. 2011. Push and Swap: Fast Cooperative Path-Finding with Completeness Guarantees. In Walsh, T., ed., *IJCAI 2011, Proceedings of the 22nd International Joint Conference on Artificial Intelligence, Barcelona, Catalonia, Spain, July 16-22, 2011*, 294–300. IJCAI/AAAI.

Rader, D. 2010. *Deterministic Operations Research: Models and Methods in Linear Optimization*. Wiley. ISBN 9780470484517.

Ryan, M. 2007. Graph Decomposition for Efficient Multi-Robot Path Planning. In Veloso, M. M., ed., *IJCAI 2007, Proceedings of the 20th International Joint Conference on Artificial Intelligence, Hyderabad, India, January 6-12, 2007*, 2003–2008.

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

Standley, T. S. 2010. Finding Optimal Solutions to Cooperative Pathfinding Problems. In *Proceedings of the Twenty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2010*. 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 Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI 2019*, 1177–1183. ijcai.org.

Surynek, P.; Felner, A.; Stern, R.; and Boyarski, E. 2016. An Empirical Comparison of the Hardness of Multi-Agent Path Finding under the Makespan and the Sum of Costs Objectives. In *Proceedings of the Ninth Annual Symposium on Combinatorial Search, SOCS 2016*, 145–147. AAAI Press.

Yu, J.; and LaValle, S. M. 2013. Planning optimal paths for multiple robots on graphs. In *2013 IEEE International Conference on Robotics and Automation, 2013*, 3612–3617. IEEE.