Sum of Costs Optimal Multi-Agent Path Finding with Continuous Time via Satisfiability Modulo Theories

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

Abstract

Multi-agent path finding with continuous movements and time (denoted MAPF$^{\mathcal{R}}$) is addressed. The task is to navigate agents that move smoothly between predefined positions to their individual goals so that they do not collide. Recently a novel solving approach for obtaining makespan optimal solutions called SMT-CCBS based on *satisfiability modulo theories* (SMT) has been introduced. We extend the approach further towards the sum-of-costs objective which is a more challenging case in the yes/no SMT environment due to more complex calculation of the objective.

Introduction

In *multi-agent path finding* (MAPF) (Kornhauser, Miller, and Spirakis 1984; Silver 2005; Ryan 2008; Surynek 2009; Wang and Botea 2011; Sharon et al. 2013, 2015; Botea and Surynek 2015) the task is to navigate agents from given starting positions to given individual goals. The problem takes place in undirected graph $G = (V, E)$ where agents from set $A = \{a_1, a_2, ..., a_k\}$ are placed in vertices with at most one agent per vertex. The navigation task can be then expressed formally as transforming an initial configuration of agents $\alpha_0 : A \rightarrow V$ to a goal configuration $\alpha_+ : A \rightarrow V$ using instantaneous movements across edges assuming no collision occurs.

In this work, we are dealing with an extension of MAPF introduced recently (Andreychuk et al. 2019; Surynek 2020) that considers continuous movements and time (MAPF$^{\mathcal{R}}$). Agents move smoothly along predefined curves interconnecting predefined positions placed arbitrarily in some continuous space. In contrast to MAPF, where the collision is defined as the simultaneous occupation of a vertex or an edge by two agents, collisions are defined as any spatial overlap of agents' bodies in MAPF$^{\mathcal{R}}$.

We use the definition of MAPF with continuous movements and time denoted MAPF$^{\mathcal{R}}$ from (Andreychuk et al. 2019). MAPF$^{\mathcal{R}}$ shares components with the standard MAPF: undirected graph $G = (V, E)$, set of agents $A = \{a_1, a_2, ..., a_k\}$, and the initial and goal configuration of agents: $\alpha_0 : A \rightarrow V$ and $\alpha_+ : A \rightarrow V$. A simple 2D variant of MAPF$^{\mathcal{R}}$ is as follows:

Definition 1 *(**MAPF$^{\mathcal{R}}$**) Multi-agent path finding with continuous time and space is a 5-tuple $\Sigma^{\mathcal{R}} = (G = (V, E), A, \alpha_0, \alpha_+, \rho)$ where G, A, α_0, α_+ are from the standard MAPF and ρ determines continuous extensions:*

- *$\rho.x(v), \rho.y(v)$ for $v \in V$ represent the position of v*
- *$\rho.speed(a)$ for $a \in A$ determines constant speed of a*
- *$\rho.radius(a)$ for $a \in A$ determines the radius of a; we assume that agents are omni-directional discs*

For simplicity we assume circular agents with constant speed and instant acceleration. The major difference from the standard MAPF where agents move instantly between vertices (disappears in the source and appears in the target instantly) is that smooth continuous movement between a pair of vertices (positions) along the straight line interconnecting them takes place in MAPF$^{\mathcal{R}}$.

An example of MAPF$^{\mathcal{R}}$ and makespan/sum-of-costs optimal solution is shown in Figure 1.

A Satisfiability Modulo Theory Approach

A recent algorithm called SMT-CBS$^{\mathcal{R}}$ (Surynek 2020) rephrases CCBS as problem solving in *satisfiability modulo theories* (SMT) (Bofill et al. 2012; Tinelli 2010). The basic use of SMT divides the satisfiability problem in some complex theory T into a propositional part that keeps the Boolean structure of the problem and a simplified procedure $DECIDE_T$ that decides fragment of T restricted on *conjunctive formulae*. T in our case is represented by MAPF$^{\mathcal{R}}$ movement rules.

The key question in the propositional logic-based approach is what will be the decision variables. In the standard MAPF, time expansion of G for every time step can be done resulting in a multi-value decision diagram (MDD) (Surynek et al. 2016) representing possible positions of agents at any time step. Since MAPF$^{\mathcal{R}}$ is no longer discrete we cannot afford to use a decision variable for every time moment.

Analogously to MDD, we introduce *real decision diagram* (RDD). RDD$_i$ defines for agent a_i its space-time positions and possible movements. Formally, RDD_i is a directed graph (X^i, E^i) where X_i consists of pairs (u, t) with $u \in V$ and $t \in \mathbb{R}_0^+$ is time and E_i consists of directed edges of the form $((u, t_u); (v, t_v))$. Edge $((u, t_u); (v, t_v))$ correspond to agent's movement from u to v started at t_u and finished at t_v. Waiting in u is possible by introducing edge

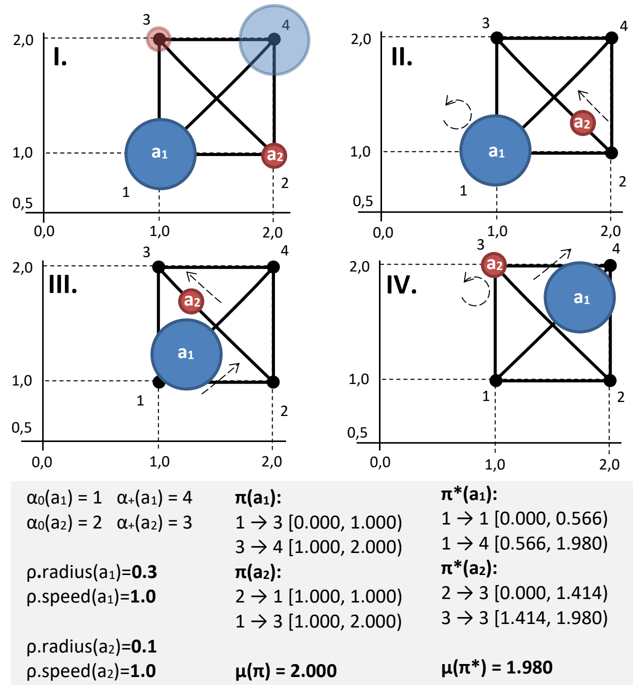

Figure 1: An example of MAPF$^{\mathcal{R}}$ instance with two agents. A feasible makespan sub-optimal solution π (makespan $\mu(\pi) = 2.0$) and makespan optimal solution $\pi*$ (makespan $\mu(\pi*) = 1.980$) are shown.

$((u, t_u); (v, t'_u))$. Pair $(\alpha_0(a_i), 0) \in X_i$ indicates start and $(\alpha_+(a_i), t)$ for some t corresponds to reaching the goal position.

RDDs for individual agents are constructed with respect to collision avoidance constraints. If there is no collision avoidance constraint then RDD$_i$ simply corresponds to a shortest temporal plan for agent a_i. But if a collision avoidance constraint is present, say $(a_i, (u, v), [\tau_0, \tau_+))$, and we are considering movement starting in u at t that interferes with the constraint, then we need to generate a node into RDD$_i$ that allows agent to wait until the unsafe interval passes by, that is node (u, τ^+) and edge $((u, \tau^+); (u, \tau^+))$ are added.

We introduce a decision variable for each node and edge $[RDD_1, ..., RDD_k]$; RDD$_i$ $= (X^i, E^i)$: we have variable $\mathcal{X}_u^t(a_i)$ for each $(u, t) \in X^i$ and $\mathcal{E}_{u,v}^{t_u, t_v}(a_i)$ for each directed edge $((u, t_u); (v, t_v)) \in E^i$. The meaning of variables is that $\mathcal{X}_u^t(a_i)$ is $TRUE$ if and only if agent a_i appears in u at time t and similarly for edges: $\mathcal{E}_{u,v}^{t_u, t_v}(a_i)$ is $TRUE$ if and only if a_i moves from u to v starting at time t_u and finishing at t_v.

From the perspective of SMT, the propositional level does not understand geometric properties of agents so cannot know what simultaneous variable assignments are invalid. This information is only available at the level of theory T = MAPF$^{\mathcal{R}}$ through $DECIDE_{MAPF^{\mathcal{R}}}$. We also leave the bounding of the sum-of-costs at the level of $DECIDE_{MAPF^{\mathcal{R}}}$.

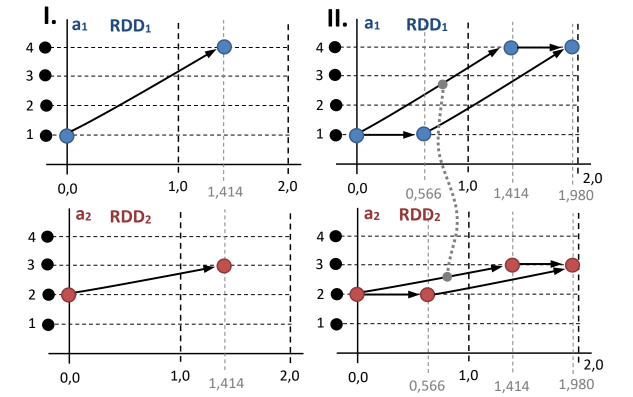

Figure 2: Real decision diagrams (RDDs) for agents a_1 and a_2 from MAPF$^{\mathcal{R}}$ from Figure 1. Decisions corresponding to shortest paths for agents a_1 and a_2 moving diagonally towards their goals are shown: $a_1 : 1 \rightarrow 4$, $a_2 : 2 \rightarrow 3$ (left). This leads to a collision whose resolution is either waiting for agent a_1 in vertex 1 from 0.000 until 0.566 or waiting for agent a_2 in vertex 2 from 0.000 until 0.566 (right).

Lazy Encoding of Sum-of-Costs Bounds

The SMT-based algorithm similarly as CCBS resolves collisions between agents (Surynek 2020). After resolving all collisions we check whether the sum-of-costs bound is satisfied by the resulting plan. This can be done easily by checking if $\mathcal{X}_u^{t_u}(a_i)$ variables across all agents together yield higher cost than the cost bound ξ or not. If cost bound ξ is exceeded then corresponding nogood is recorded and added to the encoding and the algorithm continues by searching for a new truth-value satisfying assignment. The nogood says that $\mathcal{X}_u^{t_u}(a_i)$ variables that jointly exceed ξ cannot be simultaneously set to $TRUE$.

Formally, the nogood constraint can be represented as a set of variables $\{\mathcal{X}_{u_1}^{t_1}(a_1), \mathcal{X}_{u_2}^{t_2}(a_2), ... \mathcal{X}_{u_k}^{t_k}(a_k)\}$. We say the nogood to be *dominated* by another nogood $\{\mathcal{X}_{u_1}^{t'_1}(a_1), \mathcal{X}_{u_2}^{t'_2}(a_2), ... \mathcal{X}_{u_k}^{t'_k}(a_k)\}$ if and only if $t'_i \leq t_i$ for $i = 1, 2, ...k$ and $\exists i \in \{1, 2, ..., k\}$ such that $t'_i < t_i$. To make the nogood reasoning more efficient we do not need to store nogoods that are dominated by some previously discovered nogood. In such case however, the single nogood does not forbid one particular assignment but all assignments that could lead to dominated nogoods.

Conclusion

We extended the approach based on *satisfiability modulo theories* (SMT) for solving MAPF$^{\mathcal{R}}$ from the makespan objective towards the sum-of-costs objective. Bounding the sum-of-costs is done in a lazy way through introducing nogoods incrementally. The work on MAPF$^{\mathcal{R}}$ could be further developed into multi-robot motion planning in continuous configuration spaces (LaValle 2006).

Acknowledgement. The author has been supported by GAČR - the Czech Science Foundation, grant registration number 19-17966S.

References

Andreychuk, A.; Yakovlev, K. S.; Atzmon, D.; and Stern, R. 2019. Multi-Agent Pathfinding with Continuous Time. In *Proceedings of IJCAI 2019*, 39–45.

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

Botea, A.; and Surynek, P. 2015. Multi-Agent Path Finding on Strongly Biconnected Digraphs. In *AAAI*, 2024–2030.

Kornhauser, D.; Miller, G. L.; and Spirakis, P. G. 1984. Coordinating Pebble Motion on Graphs, the Diameter of Permutation Groups, and Applications. In *FOCS, 1984*, 241–250.

LaValle, S. M. 2006. *Planning algorithms*. Cambridge University Press. ISBN 978-0-521-86205-9.

Ryan, M. R. K. 2008. Exploiting Subgraph Structure in Multi-Robot Path Planning. *J. Artif. Intell. Res. (JAIR)* 31: 497–542.

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

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.

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

Surynek, P. 2009. A novel approach to path planning for multiple robots in bi-connected graphs. In *ICRA 2009*, 3613–3619.

Surynek, P. 2020. On Satisfisfiability Modulo Theories in Continuous Multi-Agent Path Finding: Compilation-based and Search-based Approaches Compared. In Rocha, A. P.; Steels, L.; and van den Herik, H. J., eds., *Proceedings of the 12th International Conference on Agents and Artificial Intelligence, ICAART 2020, Volume 2, 2020*, 182–193. SCITEPRESS.

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.

Tinelli, C. 2010. Foundations of Satisfiability Modulo Theories. In *Proceedings is WoLLIC 2010*, 58.

Wang, K.; and Botea, A. 2011. MAPP: a Scalable Multi-Agent Path Planning Algorithm with Tractability and Completeness Guarantees. *JAIR* 42: 55–90.