Multi-Goal Multi-Agent Path Finding via Decoupled and Integrated Goal Vertex Ordering

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

Abstract

We introduce multi-goal multi agent path finding (MG-MAPF) which generalizes the standard discrete multi-agent path finding (MAPF) problem. While the task in MAPF is to navigate agents in an undirected graph from their starting vertices to one individual goal vertex per agent, MG-MAPF assigns each agent multiple goal vertices and the task is to visit each of them at least once. Solving MG-MAPF not only requires finding collision free paths for individual agents but also determining the order of visiting agent's goal vertices so that common objectives like the sum-of-costs are optimized.

Introduction

Mutli-agent path finding (MAPF) (Silver 2005; Ryan 2008; Surynek 2009; Luna and Bekris 2011; Wang and Botea 2011; Ma and Koenig 2017) is an abstraction for many real-life problems where agents need to be moved (see (Felner et al. 2017; Ma and Koenig 2017) for a survey). The environment in MAPF is modeled as an undirected graph where vertices represent positions and edges define the topology.

The standard variant of MAPF assumes that each agent starts in a specified starting vertex and its task is to reach a specified goal vertex. While such formalization encompass many real-life navigation tasks (Cáp et al. 2013; Ma et al. 2017a) there still exist problems especially in logistic domain where the standard MAPF lacks expressiveness.

Such problems that cannot be expressed as MAPF include situations where agents have multiple goals so that instead of reaching single goal location agents need to perform a round-trip to service a set of goals. Having multiple goals per agent adds a significant new challenge to the problem consisting of determining the order of visiting agent's goal vertices. Hence the ordering of goals as well as non-conflicting path finding are subject to decision making which in addition to this aims on the optimization of various objectives such as commonly adopted sum-of-costs (Standley 2010; Sharon et al. 2013)

We introduce a problem we call a *multi-goal multi-agent path finding* (MG-MAPF) and a novel solving algorithm for MG-MAPF: a search-based Hamiltonian CBS (HCBS), a derivative of the CBS algorithm (Sharon et al. 2015) [1].

[1]This is a short version of (Surynek 2020, in press in Proceedings of AAAI 2021) where in addition, a SAT-based algorithm for MG-MAPF has been introduced

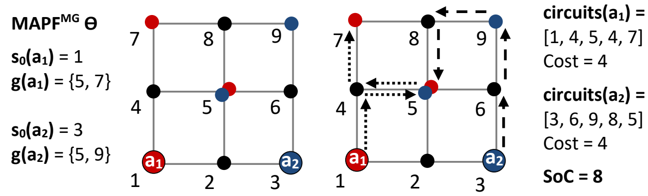

Figure 1: A MG-MAPF instance with and its sum-of-costs optimal solution.

Background and Definitions

Agents in MAPF are placed in vertices of an undirected graph so that there is at most one agent per vertex. Formally, $s : A \mapsto V$ is a *configuration* of agents in vertices of the graph. A configuration can be transformed instantaneously to the next one by *valid* movements of agents; the next configuration corresponds to the next *time step*. An agent can move into another vertex across an edge provided no collision occurs. The configuration at time step t is denoted s_t.

Definition 1 Mutli-goal multi-agent path finding *(MG-MAPF) is a 4-tuple* $\Theta = (G = (V, E), A, s_0, g)$ *where* $G = (V, E)$ *is an undirected graph,* $A = \{a_1, a_2, ..., a_k\}$ *with* $k \in \mathbb{N}$ *is a set of agents where* $k \leq |V|$, $s : A \mapsto V$ *represents agents' starting vertices (starting configuration), and* $g : A \mapsto 2^V$ *assigns a set of goal vertices to each agent.*

Each agent in MG-MAPF has the task to visit its goal vertices. Agent's goal vertices can be visited in an arbitrary order but each goal vertex must be visited by the agent at least once. Various objectives can be taken into account. We develop all concepts here for the *sum-of-costs* objective commonly adopted in MAPF but different cumulative objectives can be used as well (Surynek et al. 2016).

Formally, an MG-MAPF solution is a sequence of configurations that can be obtained via valid moves from the starting configuration s_0 following the MAPF movement rules such that each agent visits each of its goal vertices at least once:

Definition 2 *A solution to MG-MAPF* $\Theta = (G = (V, E), A, s_0, g)$ *is a sequence of configurations* $S =$

$[s_0, s_1, ..., s_{t_M}]$ *such that* s_t *results from* s_{t-1} $\forall t \in \{1, 2, ..., t_M\}$ *via valid MAPF movements and* $\forall a_i \in A$ *it holds that* $(\forall v_g \in g(a_i))$ $(\exists t \in \{1, 2, ..., t_M\})$ $(s_t(a_i) = v_g)$.

Given a solution $S = [s_0, s_1, ..., s_{t_M}]$, t_M is the *makespan*, denoted $Make(S)$. We define the *sum-of-costs* as the sum of costs for individual agents: $SoC(S) = \sum_{i=1}^{k} Cost(a_i)$, where the individual cost for agent a_i is defined as: $Cost(a_i) = \min \{t_c \mid (\forall v_g \in g(a_i))\ (\exists t \in 1, 2, ..., t_c)\ (s_t(a_i) = v_g)\}$.

Hamiltonian Conflict-based Search: HCBS

We suggest a novel algorithm called Hamiltonian Conflict-based Search (Hamiltonian CBS, HCBS) which shares the high level structure with the CBS algorithm (Sharon et al. 2015).

When trying to use CBS for MG-MAPF, the significant challenge is represented by the fact that at the low level there is no longer search for a minimum cost path with respect to the set of conflicts, a polynomial-time problem, but rather the search for a minimum cost Hamiltonian path.

Definition 3 *A* Hamiltonian path *(HP) in G starting at* $u \in V$ *covering a subset of vertices* $U \subseteq V$ *is a sequence of vertices denoted* $H_P(u, U) = [h_0, h_1, ..., h_{t_H}]$ *such that* $h_1 = u$, $\{h_t, h_{t+1}\} \in E$ *for* $t \in \{0, 1, ..., t_H - 1\}$, *and for each* $v \in U$ $\exists t \in \{0, 1, ..., t_H\}$ *such that* $h_t = v$. *The cost of Hamiltonian path corresponds to the number of its edges:* $Cost(H_P(u, U)) = t_H - 1$.

The key to adapt CBS for MG-MAPF is to decouple the goal vertex ordering from conflict avoidance. The search for a Hamiltonian path going through agent's goal vertices with respect to a set of conflicts is done in two level fashion. At the high-level (of this low level) we are trying to determine optimal ordering of agent's goal vertices. To this purpose we made use of the A* algorithm (Hart, Nilsson, and Raphael 1968) that searches the space of possible **permutations** of agent's goals. After determining the next goal vertex to visit, the algorithm searches for the shortest path observing the conflicts that interconnects the next goal vertex with the current one. The search for the shortest path is done by another instance of A*. Another important factor for the performance of the decoupled approach are the heuristics.

We define a variant of *spanning tree* with respect to a subset of vertices of undirected graph $G = (V, E)$.

Definition 4 *A* spanning tree *(ST) of an undirected graph* $G = (V, E)$ *with respect to a subset of vertices* $U \subseteq V$, *denoted* $T_S(U) = (V_U, E_U)$ *is a tree covering U, that is,* $U \subseteq V_U \subseteq V$ *and* $E_U \subseteq E$. *The* cost *of a spanning tree is defined as the number of edges included in the tree:* $Cost(T_S(U)) = |E_U|$. *A* minimum spanning tree *(MST) with respect to U is a spanning tree with minimum cost.*

Observe that $T_S(U)$ may contain other vertices in addition to U to keep it connected. We use the notation $T_S(u, U)$ for $u \in V$ denoting a spanning tree covering $\{u\} \cup U$.

The important property of MST is that it can be found in polynomial time with respect to G (Boruvka 1926; Nesetril, Milková, and Nesetrilová 2001) and can serve as the lower bound for the cost of shortest Hamiltonian path.

Algorithm 1: HCBS algorithm for MG-MAPF.

HCBS$_{conflicts}$ **(**$\Theta = (G = (V, E), A, s_0, g)$**)**
  $N.constraints \leftarrow \emptyset$
  $N.circuits \leftarrow \{circuit^*(a_i)$ a shortest Hamiltonian path from $s_0(a_i)$ covering $g(a_i) \mid i = 1, 2, ..., k\}$
  $N.SoC \leftarrow \sum_{i=1}^{k} Cost(N.circuits(a_i))$
  insert (SoC, N) into OPEN
  while OPEN $\neq \emptyset$ **do**
    $(key, N) \leftarrow$ min-Key(OPEN)
    remove-Min-Key(OPEN)
    $collisions \leftarrow$ validate($N.circuits$)
    if $collisions = \emptyset$ **then**
      return $N.circuits$
    let $(a_i, a_j, v, t) \in collisions$
    for *each* $a \in \{a_i, a_j\}$ **do**
      $N'.constraints \leftarrow N.constraints \cup \{(a, v, t)\}$
      $N'.circuits \leftarrow N.circuits$
      $N'.circuits(a_i) \leftarrow$ HCBS$_{ordering}$ $(\Theta, a, N'.constraints)$
      $SoC' \leftarrow \sum_{i=1}^{k} Cost(N'.circuits(a_i))$
      insert (SoC', N') into OPEN

This enables us to use the concept of MST as a basis for a *consistent* A* heuristic (proof omitted). The HCBS algorithm is described using pseudo-code as Algorithm 1. The algorithm consists of **thee levels of search** in three different spaces: (i) space of conflicts, (ii) goal ordering space, and (iii) path space.

Related Work

The most closely related problem to MG-MAPF is *multi-agent pickup and delivery* (MAPD) (Ma et al. 2017b; Liu et al. 2019), defining a set of tasks $T = \{t_1, t_2, ...t_m\}$ where each task $t_j = (p_j, d_j)$ is characterized by a pickup location $p_j \in V$ and a delivery location $d_j \in V$. Agents can freely select tasks to fulfill. The contemporary solving approaches for MAPD first assign tasks to agents and followed by determining the ordering of tasks per agent ignoring collisions. Then collision free paths are planned according to the task ordering. As there is no feedback between the phases the resulting plan is sub-optimal.

Conclusion

HCBS introduces three level search in which conflict resolution is done at the high level and goal vertex ordering and path planning are done at the low level. The key technique is decoupling the vertex ordering from collision-free path planning. CBS framework also provides great room for integrating heuristics that we made use of when adapting it for MG-MAPF.

Acknowledgement

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

References

Boruvka, O. 1926. O jistém problému minimáním. *Práce Moravské přírodovědecké společnosti* 3(3): 7–58.

Cáp, M.; Novák, P.; Vokrínek, J.; and Pechoucek, M. 2013. Multi-agent RRT: sampling-based cooperative pathfinding. In Gini, M. L.; Shehory, O.; Ito, T.; and Jonker, C. M., eds., *International conference on Autonomous Agents and Multi-Agent Systems, AAMAS 2013*, 1263–1264. IFAAMAS.

Felner, A.; Stern, R.; Shimony, S. E.; Boyarski, E.; Goldenberg, M.; Sharon, G.; Sturtevant, N. R.; Wagner, G.; and Surynek, P. 2017. Search-Based Optimal Solvers for the Multi-Agent Pathfinding Problem: Summary and Challenges. In *Proceedings of the Tenth International Symposium on Combinatorial Search, SOCS 2017*, 29–37. AAAI Press.

Hart, P. E.; Nilsson, N. J.; and Raphael, B. 1968. A Formal Basis for the Heuristic Determination of Minimum Cost Paths. *IEEE Trans. Syst. Sci. Cybern.* 4(2): 100–107.

Liu, M.; Ma, H.; Li, J.; and Koenig, S. 2019. Task and Path Planning for Multi-Agent Pickup and Delivery. In *Proceedings of the 18th International Conference on Autonomous Agents and MultiAgent Systems, AAMAS '19, Montreal, QC, Canada, May 13-17, 2019*, 1152–1160. International Foundation for Autonomous Agents and Multiagent Systems.

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.

Ma, H.; and Koenig, S. 2017. AI buzzwords explained: multi-agent path finding (MAPF). *AI Matters* 3(3): 15–19.

Ma, H.; Koenig, S.; Ayanian, N.; Cohen, L.; Hönig, W.; Kumar, T. K. S.; Uras, T.; Xu, H.; Tovey, C. A.; and Sharon, G. 2017a. Overview: Generalizations of Multi-Agent Path Finding to Real-World Scenarios. *CoRR* abs/1702.05515.

Ma, H.; Li, J.; Kumar, T. K. S.; and Koenig, S. 2017b. Lifelong Multi-Agent Path Finding for Online Pickup and Delivery Tasks. In *Proceedings of the 16th Conference on Autonomous Agents and MultiAgent Systems, AAMAS 2017, São Paulo, Brazil, May 8-12, 2017*, 837–845. ACM.

Nesetril, J.; Milková, E.; and Nesetrilová, H. 2001. Otakar Boruvka on minimum spanning tree problem Translation of both the 1926 papers, comments, history. *Discret. Math.* 233(1-3): 3–36.

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

Sharon, G.; Stern, R.; Felner, A.; and Sturtevant, N. R. 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 *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 Fox, M.; and Poole, D., eds., *Proceedings of the Twenty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2010*. AAAI Press.

Surynek, P. 2009. A novel approach to path planning for multiple robots in bi-connected graphs. In *2009 IEEE International Conference on Robotics and Automation, ICRA 2009, Kobe, Japan, May 12-17, 2009*, 3613–3619. IEEE.

Surynek, P. 2020, in press in Proceedings of AAAI 2021. Multi-Goal Multi-Agent Path Finding via Decoupled and Integrated Goal Vertex Ordering. *CoRR* abs/2009.05161.

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, Tarrytown, NY, USA, July 6-8, 2016*, 145–147. AAAI Press.

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