Multi-Agent Path Finding with Time Windows: Preliminary Results

Extended Abstract

Jianqi Gao
Harbin Institute of Technology (Shenzhen)
Shenzhen, China
gaojianqi205a@stu.hit.edu.cn

Qi Liu
Harbin Institute of Technology (Shenzhen)
Shenzhen, China
19B953036@stu.hit.edu.cn

Shiyu Chen
Harbin Institute of Technology (Shenzhen)
Shenzhen, China
chenshiyu@stu.hit.edu.cn

Kejian Yan
Harbin Institute of Technology (Shenzhen)
Shenzhen, China
22s053071@stu.hit.edu.cn

Xinyi Li
Harbin Institute of Technology (Shenzhen)
Shenzhen, China
22S153132@stu.hit.edu.cn

Yanjie Li*
Harbin Institute of Technology (Shenzhen)
Shenzhen, China
autolyj@hit.edu.cn

ABSTRACT

We formalize the problem of multi-agent path finding with time windows (MAPF-TW). The optimization objective is to maximize the average customer satisfaction for all agents when they reach their respective goal vertices without path conflicts. We first prove that solving MAPF-TW optimally is NP-hard. We then reduce the MAPF-TW problem into a multi-commodity flow problem and propose an integer linear programming (ILP) model. Next, we propose the conflict-based search with time windows (CBS-TW) for the MAPF-TW problem, which is also optimal. Finally, we conduct simulation experiments on two different maps with random obstacles.

KEYWORDS

Multi-robot system; Path planning for multiple mobile robots or agents; Planning, scheduling and coordination

ACM Reference Format:
Jianqi Gao, Qi Liu, Shiyu Chen, Kejian Yan, Xinyi Li, and Yanjie Li*. 2023. Multi-Agent Path Finding with Time Windows: Preliminary Results: Extended Abstract. In *Proc. of the 22nd International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2023), London, United Kingdom, May 29 – June 2, 2023*, IFAAMAS, 3 pages.

1 INTRODUCTION

Multi-agent path finding (MAPF) studies how to find paths without collision for multiple agents [2]. MAPF has attracted significant attention because of its widely practical applications in automated systems, such as automated warehouses [13], and automatic aircraft trailers [4]. Solving MAPF optimally is NP-hard[16]. Many critical results are reported on this topic, such as optimal [6–8], bounded sub-optimal [1, 10], and unbounded sub-optimal [3, 9, 14, 15] algorithms.

Sometimes agents need to complete some tasks with time constraints. Several studies have explored the effect of task time constraints on MAPF. MAPF-DL [5] maximizes the number of agents that can reach their given goal vertices within the deadline. MAPF-DT [12] proposes some due time-related objectives, which reflect the degree of deadline violations. In many practical scenarios, such as takeaway and delivery, the customer is satisfied only when the agent completes a task within the time window. However, the previous studies do not focus on how to meet the time window. We thus formalize the multi-agent path finding with time windows (MAPF-TW) to maximize the average customer satisfaction without path conflicts. We first prove solving MAPF-TW optimally is NP-hard. Then the MAPF-TW problem is reduced into a multi-commodity flow problem, and an integer linear programming (ILP) model is proposed. Next, we propose the conflict-based search with time windows (CBS-TW), whose cost is the customer satisfaction.

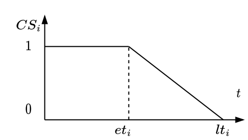

Figure 1: An illustration of time window.

2 PROBLEM DEFINITION

The input of MAPF-TW consists of a undirected graph $G(V, E)$, agent set R and time window set T ($T \subset \mathbb{R}^+$). V and E in graph G are the vertex and edge set respectively. Agent set $\{a_1, ..., a_n\}$ includes n agents moving on graph G. Every agent a_i has a start vertex, $s_i \in V$ and a goal vertex, $g_i \in V$. Time is assumed to be discreted. Agent a_i is located in vertix v at time t and move to an adjacent unblocked vertex v' that meet $(v, v') \in E$ or stays in current vertex v in the next time $t + 1$. The path π_i of agent a_i can be represented by a sequence of vertices $\pi_i = \left(v_i^1, v_i^2, ..., v_i^t\right)$, where $v_i^t \in V$. Vertex conflict $\left(a_i, a_j, v, t\right)$ and edge conflict $\left(a_i, a_j, v, v', t\right)$ represent the path collisions. The solution $\pi = (\pi_1, \pi_2, ..., \pi_n)$ is a set of feasible conflict-free paths. As shown in Fig. 1, the time window TW_i of agent a_i is composed of two elements, the early time et_i and the last service time lt_i. Customer satisfaction (CS) is

Proc. of the 22nd International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2023), A. Ricci, W. Yeoh, N. Agmon, B. An (eds.), May 29 – June 2, 2023, London, United Kingdom.

equal to 1 when agent a_i reaches the goal vertex before et_i. When agent a_i reaches after lt_i, customer satisfaction is equal to 0. When agent a_i reaches between et_i and lt_i, customer satisfaction decreases linearly from 1 to 0. The objective of MAPF-TW is to maximize the average customer satisfaction $\overline{CS}$ or minimize its negative value.

It is NP-hard to solve MAPF-TW optimally with the maximum average customer satisfaction $\overline{CS}$. The NP-complete **3-SAT** problem [11] can be reduced to the MAPF-TW problem, which implies solving MAPF-TW optimally is also NP-hard.

3 ILP-BASED MAPF-TW MODEL

We first translate the MAPF-TW problem into the minimum cost maximum multi-flow problem on a time-expanded network $\mathcal{G}\{\mathcal{V}, \mathcal{E}\}$. Then we propose an ILP model for the MAPF-TW problem that is adapted from the model of MAPF-DT [12]. T is the time horizon of the time-expanded network. If agent a_i goes through $e_j \in \mathcal{E}$, then the decision variable $x_{i,j} = 1$; otherwise $x_{i,j} = 0$. If agent a_i reaches the goal vertex at time t, the decision variable $y_i^t = 1$; otherwise $y_i^t = 0$. If agent a_i goes through e_j to reach the goal vertex at time t, $x_{i,j}$ can be defined as x_i^t.

$$\sum_{i=1}^{n} x_{i,j} \leq 1, \forall e_j \in \mathcal{E} \tag{1}$$

$$\sum_{e_j \in \delta^+(v_t)} x_{i,j} - \sum_{e_j \in \delta^-(v_t)} x_{i,j} = 0, \forall 1 \leq i \leq n, v_t \in \mathcal{V} \setminus \{\mathcal{S}^+, \mathcal{S}^-\} \tag{2}$$

$$\sum_{e_j \in \delta^-(s_i)} x_{i,j} = \sum_{e_j \in \delta^+(g_i)} x_{i,j} = 1, \forall 1 \leq i \leq n, s_i \in \mathcal{S}^+, g_i \in \mathcal{S}^- \tag{3}$$

$$\sum_{i=1}^{n} \sum_{e_j \in \delta^-(v_t)} x_{i,j} \leq 1, \forall v_t \in \mathcal{V} \tag{4}$$

$$\sum_{i=1}^{n} x_{i,(u_t, v_{t+1})} + \sum_{i=1}^{n} x_{i,(v_t, u_{t+1})} \leq 1, \forall (u_t, v_{t+1}), (v_t, u_{t+1}) \in \mathcal{E} \tag{5}$$

$$y_i^t \leq y_i^{t+1}, y_i^t \leq x_i^t, y_i^{t+1} - y_i^t + x_i^t \leq 1, \forall 1 \leq i \leq n, 1 \leq t \leq T-1 \tag{6}$$

$$y_i^T = x_i^T, \forall 1 \leq i \leq n \tag{7}$$

We can use constraints (1)–(5) to generate feasible MAPF solution. Constraint (6) and constraint (7) represent the time constraint and the relationship between $x_i^T (x_{i,j})$ and y_i^T respectively. We assume that agents stay at their goal vertices after arrival. The arrival time for agent a_i can be expressed as $T - \sum_{t=1}^{T} y_i^t$. The customer satisfaction CS_i of agent a_i is:

$$CS_i = \begin{cases} 1, & T - \sum_{t=1}^{T} y_i^t \leqslant et_i \\ \frac{lt_i - (T - \sum_{t=1}^{T} y_i^t)}{lt_i - et_i}, & et_i < T - \sum_{t=1}^{T} y_i^t < lt_i \\ 0, & lt_i \leqslant T - \sum_{t=1}^{T} y_i^t \end{cases} \tag{8}$$

We minimize the negative average customer satisfaction, which can be expressed as:

$$\min \quad \overline{CS} = -\frac{\sum_1^n CS_i}{n}, \tag{9}$$

where n is the number of agents.

4 CBS-TW

In this section, we present an optimal MAPF-TW algorithm, CBS-TW, which is based on CBS [7]. Algorithm 2 shows the high-level of CBS-TW. At first, CBS-TW will check the root node to see whether there are path conflicts in a path plan. If there are no path conflicts, we get the optimal solution. Otherwise, based on the first path conflict, we expand two new constraint tree (CT) nodes from the root node and put them into the OPEN set. The root node is thrown into the CLOSED set. Next time, we do a best-first search and choose the CT node with the minimum cost from the OPEN set. If there are path conflicts in the selected CT node, we continue to expand until we find a CT node without path conflicts. On the low level, CBS-TW performs A^* to find the optimal path for a single agent from its start vertex to its goal vertex with the constraints. Then it returns the agent's path cost $-CS_i$, which is calculated by the function (8) of the ILP model. Additionally, we prune all agents with time step $> lt_i$.

Algorithm 1 High level of CBS-TW

Require: MAPF-TW instance
Ensure: Best path solution π found so far

```
Root.constraints ← ∅
Root.plan ← find path for every agent by the low-level search
Root.cost ← 0
OPEN set ← {Root}
while true do
    N ← argmin_{N∈OPEN set} N.cost
    OPEN set ← OPEN set \ {N}
    Check whether there is a path conflict in N.plan
    if N.plan has no path conflict then
        Return N.plan
    end if
    C ← first path conflict (a_i, a_j, ...) in N.plan
    for a_i in C do
        N' ← new CT node
        N'.plan ← N.plan
        N'.constraints ← N.constraints ⋃ {a_i, a_j, ...}
        Update N'.plan by the low-level search for a_i
        N'.cost ← average customer satisfaction in N'.plan
        OPEN ← OPEN set ⋃ {N'}
    end for
end while
```

5 SIMULATION EXPERIMENTS

In this section, we count the success rate (SR) of CBS-TW when the number of agents changes from 10 to 70 on two different maps through simulation experiments. We conduct all experiments on a 3.00GHz Intel Core i7-9700 desktop computer with 32 GB RAM. Experiments are done using python 3.7. We use the 4-neighbor 2D random map of 32×32 and 128×128. The density of the obstacle is 20%. Every test needs 50 instances. We randomly generate the start vertices, goal vertices, and time windows with the different numbers of agents for one instance. The limited running time is set to be 60 seconds. The following table shows the SR of CBS-TW. Since the agent density is high and the path conflicts are more in the small map, CBS-TW takes more time to expand more CT nodes to solve the path conflicts. While the agent density is low and the number of path conflicts is smaller in the large map, CBS-TW has a higher SR.

Table 1: The SR of CBS-TW in two different maps.

Metrics	Map	Number of agents						
		10	20	30	40	50	60	70
SR	32	100%	92%	42%	24%	2%	4%	0%
	128	100%	100%	100%	98%	76%	72%	48%

ACKNOWLEDGMENTS

This work was supported by the Shenzhen Basic Research Program under Grant JCYJ20180507183837726, the National Natural Science Foundation of China under Grant 61977019 and the Shenzhen Science and Technology Program under Grant JSGG20201103093802006.

REFERENCES

[1] Max Barer, Guni Sharon, Ron Zvi Stern, and Ariel Felner. 2014. Suboptimal variants of the conflict-based search algorithm for the multi-agent pathfinding problem. In *7th Annual Symposium on Combinatorial Search, SoCS.* 19–27.

[2] M. Erdmann and T. Lozano-Perez. 1986. On multiple moving objects. In *IEEE International Conference on Robotics and Automation, ICRA.* 1419–1424.

[3] Jiaoyang Li, Zhe Chen, Daniel Harabor, Peter J Stuckey, and Sven Koenig. 2021. Anytime multi-agent path finding via large neighborhood search: Extended abstract. In *Proceedings of the International Joint Conference on Autonomous Agents and Multi-Agent Systems, AAMAS.* 1581–1583.

[4] Jiaoyang Li, Han Zhang, Mimi Gong, Zi Liang, Weizi Liu, Zhongyi Tong, Liangchen Yi, Robert Morris, Corina Pasareanu, and Sven Koenig. 2019. Scheduling and Airport Taxiway Path Planning under Uncertainty. *AIAA Aviation Forum* (2019).

[5] Hang Ma, Glenn Wagner, Ariel Felner, Jiaoyang Li, TK Satish Kumar, and Sven Koenig. 2018. Multi-agent path finding with deadlines. In *International Joint Conference on Artificial Intelligence, IJCAI.* 417–423.

[6] Malcolm Ross Kinsella Ryan. 2008. Exploiting subgraph structure in multi-robot path planning. *Journal of Artificial Intelligence Research* 31 (2008), 497–542.

[7] Guni Sharon, Roni Stern, Ariel Felner, and Nathan Sturtevant. 2012. Conflict-Based Search For Optimal Multi-Agent Path Finding. In *Proceedings of the AAAI Conference on Artificial Intelligence.* 563–569.

[8] Guni Sharon, Roni Stern, Meir Goldenberg, and Ariel Felner. 2013. The increasing cost tree search for optimal multi-agent pathfinding. *Artificial intelligence* 195 (2013), 470–495.

[9] David Silver. 2005. Cooperative pathfinding. In *Proceedings of the aaai conference on artificial intelligence and interactive digital entertainment, AIIDE.* 117–122.

[10] Pavel Surynek, Ariel Felner, Roni Stern, and Eli Boyarski. 2017. Modifying optimal SAT-based approach to multi-agent path-finding problem to suboptimal variants. In *Proceedings of the International Symposium on Combinatorial Search, SoCS.* 169–170.

[11] Craig A Tovey. 1984. A simplified NP-complete satisfiability problem. *Discrete applied mathematics* 8, 1 (1984), 85–89.

[12] Hanfu Wang and Weidong Chen. 2022. Multi-Robot Path Planning With Due Times. *IEEE Robot. Automat. Lett* 7, 2 (2022), 4829–4836.

[13] Peter R Wurman, Raffaello D'Andrea, and Mick Mountz. 2008. Coordinating hundreds of cooperative, autonomous vehicles in warehouses. *AI magazine* 29, 1 (2008), 9–9.

[14] Y. Xu, Y. Li, Q. Liu, J. Gao, Y. Liu, and M. Chen. 2021. Multi-agent Pathfinding with Local and Global Guidance. *IEEE International Conference on Networking, Sensing and Control*, 1–7.

[15] Zhaohui Ye, Yanjie Li, Ronghao Guo, Jianqi Gao, and Wen Fu. 2022. Multi-agent Pathfinding with Communication Reinforcement Learning and Deadlock Detection. In *Intelligent Robotics and Applications: 15th International Conference, ICIRA.* 493–504.

[16] Jingjin Yu and Steven M LaValle. 2016. Optimal multirobot path planning on graphs: Complete algorithms and effective heuristics. *IEEE Transactions on Robotics* 32, 5 (2016), 1163–1177.