Minimising Task Tardiness for Multi-Agent Pickup and Delivery

Extended Abstract

Saravanan Ramanathan
Singtel Cognitive and Artificial Intelligence Lab for Enterprises
Nanyang Technological University
Singapore
saravanan.r@ntu.edu.sg

Yihao Liu
Singtel Cognitive and Artificial Intelligence Lab for Enterprises
Nanyang Technological University
Singapore
yihao002@e.ntu.edu.sg

Xueyan Tang
Singtel Cognitive and Artificial Intelligence Lab for Enterprises
Nanyang Technological University
Singapore
asxytang@ntu.edu.sg

Wentong Cai
Singtel Cognitive and Artificial Intelligence Lab for Enterprises
Nanyang Technological University
Singapore
aswtcai@ntu.edu.sg

Jingning Li
NCS Pte Ltd
Singapore
jingning.li@ncs.com.sg

ABSTRACT

Multi-agent pickup and delivery, a variant of the multi-agent path finding problem, aims to find collision-free paths for a set of agents performing a continuous stream of pickup and delivery tasks. Owing to the service guarantee nature of applications, these agents often need to execute the tasks within their stipulated deadlines. When failure to meet task deadlines is unavoidable, there is a need to minimise the tardiness experienced by the tasks. To address this problem, we propose a cost-based integrated task assignment and path planning algorithm to assign tasks to the agents.

KEYWORDS

Multi-agent pickup and delivery; task assignment; tardiness

ACM Reference Format:
Saravanan Ramanathan, Yihao Liu, Xueyan Tang, Wentong Cai, and Jingning Li. 2023. Minimising Task Tardiness for Multi-Agent Pickup and Delivery: 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 systems are predominantly found in automated warehouse management, airport towing and etc [9, 12]. Several studies have been carried out to address the multi-agent pickup and delivery (MAPD) problem [1, 2, 4–7, 10, 11]. It involves task assignment to agents and collision-free path planning for agents using techniques such as A* search [3]. MAPD tasks are traditionally not associated with task deadlines in the literature. However, in reality, most of these tasks need to be completed by a specific deadline. That is, an agent has to pick up the task (*e.g.* an inventory item) from its pickup location and deliver it to the delivery location by its deadline for further processing. For instance, in a warehouse setting, this could be translated to a standard delivery request with a normal deadline or a priority request (*e.g.* VIP customer/urgent consignment) with an earlier deadline. This entails the need for a deadline-aware scheduling algorithm for the MAPD problem.

Motivated by scenarios such as emergency evacuation, a common deadline for all tasks is considered in [8]. A more practical model with individual task deadlines is studied in [11] with the objective of maximising the number of tasks completed by their deadlines. Although prior studies address the schedulability performance, the tardiness (*i.e.,* completion time minus the deadline) of MAPD tasks is unexplored. Hence, we study the problem of allocating MAPD tasks to agents such that the total tardiness experienced by the tasks is minimised while maintaining a high success rate.

2 PROBLEM DEFINITION

We model the MAPD environment as an undirected connected graph $\mathcal{G} = (V, E)$ where the nodes in V denote the locations and the edges in E denote the connections between locations through which an agent can move. Consider there are a set of M agents $\mathcal{A} = \{a_1, \ldots, a_M\}$ to perform a set of N delivery tasks $\mathcal{T} = \{\tau_1, \ldots, \tau_N\}$.

Each agent a_i has a unit carrying capacity and is allocated a dedicated parking location $\hat{s}_i \in V$ where it initially stays and return to upon completion of all the tasks. It is assumed that between any two consecutive timesteps, an agent can either stay at its current location or move to an adjacent location. When a task τ_j is assigned to a_i, the agent moves from $\hat{s}_i$ to the pickup location s_j (to pickup the inventory) and delivers it to the delivery location g_j without colliding with any other agents. Generally, collisions are avoided by imposing constraints during path planning which include: (i) no two agents can occupy the same node at the same timestep; (ii) no two agents can traverse the same edge (in opposite directions) between the same two consecutive timesteps. Finding the optimal subsets of tasks and collision-free paths is known to be an NP-Hard problem. Hence, we focus on developing heuristic solutions.

The tardiness of a task τ_j is given by $\delta_j = \max(0, c_j - d_j)$, where d_j and c_j denotes the deadline and completion time of the task τ_j. Our aim is to assign the tasks $\tau_j \in \mathcal{T}$ to agents such that either they can be completed before their deadlines d_j or the cumulative tardiness $\sum_j \delta_j$ is minimised.

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.

3 TASK ASSIGNMENT AND PATH PLANNING

We propose a cost-based joint task assignment and path planning algorithm to assign MAPD tasks with individual release times and deadlines to agents.

Let $\hat{t}_i$ denote the timesteps taken to complete the sequence of tasks assigned to an agent a_i according to its planned path. That is, the agent becomes available for executing the next task at timestep $\hat{t}_i$. Let $\epsilon_{i,j}$ denote the cost of executing an unassigned task τ_j by an agent a_i using an optimal path. An optimal path here refers to the shortest-time path to execute the task τ_j starting from timestep $\hat{t}_i$, which can be computed using A* search [3]. The execution cost is given by the completion time $c_{i,j}$ of the task (including the wait time of the agent for the task release) minus the completion time of the previously assigned task, *i.e.,*

$$\epsilon_{i,j} = c_{i,j} - \hat{t}_i. \tag{1}$$

The high-level idea of our approach is as follows. We assign tasks such that agents do not wait (idle) for future task releases while there are already released tasks pending to be executed in the system. All tasks are sorted in the earliest deadline first order to facilitate tardiness minimisation. Then, the completion time $c_{i,j}$ and the execution cost $\epsilon_{i,j}$ of the task τ_j by each agent are computed. We check if there exists an agent that can complete this task within its deadline. Prioritising agents that can meet task deadlines increases the success rate of the system. If at least one such agent exists, among these agents, the agent a_{i^*} with the least execution cost is chosen for assignment. If no such agent exists, we select the agent a_{i^*} with the least execution cost among all agents. Intuitively, selecting an agent with the least execution cost increases the utilisation of the agents. The agent a_{i^*} is given by

$$a_{i^*} = \begin{cases} \text{argmin}_{\{a_i | c_{i,j} \le d_j\}} \epsilon_{i,j} & \text{if } \exists (c_{i,j} \le d_j), \\ \text{argmin}_{a_i \in \mathcal{A}} \epsilon_{i,j} & \text{otherwise.} \end{cases} \tag{2}$$

Before assignment, we check if a_{i^*} is the least cost agent for any other task with a pickup time earlier than τ_j. If no such task exists, we assign task τ_j to a_{i^*}. Otherwise, we proceed with the next unassigned task. We use the multi-label A* algorithm [3] to plan the optimal path for an agent to execute the task. The node and edge access constraints imposed by the paths already planned for the previously assigned tasks are enforced while performing the A* search. We use the dummy path techniques in [5] to avoid deadlocks. Once all the task assignment process is completed, we plan a path for each agent to return to its respective parking location.

4 EXPERIMENTAL RESULTS

We evaluate the performance of our algorithm through simulation experiments and compare with several baseline approaches. We vary the number of agents $M \in \{10, 20, 30, 40, 50\}$ for a small warehouse environment [5]. The number of tasks is set to $N_a \times M$ where $N_a \in \{5, 10\}$ represents the number of tasks to be executed by an agent on average. We derive default task deadlines such that the deadlines are quite tight while feasible. That is, there exists an assignment of task sequences to agents such that all tasks can be completed by their default deadlines without considering conflicts among agents. We compare two task selection orders: (i) EDF - Earliest Deadline First and (ii) LFF - Least Flexibility First (flexibility is defined as the completion time minus the deadline) [11]. For the agent selection, we choose the agent with (i) the least execution cost (C in short) or (ii) the least response time (R in short) for the task (*i.e.,* the agent that can complete the task earliest). Same as our algorithm, the baselines also assign tasks to agents in a two-stage manner. Agents that can meet task deadlines are prioritised over agents that cannot; with both stages using particular metrics as the deciding parameters. They are named in the form of [EDF or LFF]-[C or R]-[C or R]. Thus, we end up with 8 different algorithms to compare including the proposed algorithm (*i.e.,* EDF-C-C).

Table 1: Average success rate

Small Warehouse					
Algorithm	Number of Agents				
	10	20	30	40	50
Proposed	0.838	0.781	0.723	0.693	0.670
EDF-C-R	0.524	0.372	0.257	0.198	0.179
EDF-R-R	0.469	0.330	0.226	0.194	0.173
EDF-R-C	0.745	0.709	0.652	0.636	0.629
LFF-C-C	0.722	0.608	0.522	0.494	0.484
LFF-C-R	0.159	0.05	0.021	0.010	0.008
LFF-R-R	0.500	0.311	0.233	0.188	0.159
LFF-R-C	0.615	0.554	0.492	0.461	0.451

Table 2: Average cumulative tardiness

Small Warehouse					
Algorithm	Number of Agents				
	10	20	30	40	50
Proposed	225.6	666.0	1776.6	3593.7	5945.1
EDF-C-R	291.3	1019.9	2447.7	4298.3	6725.7
EDF-R-R	352.3	1159.2	2726.9	4431.3	6965.7
EDF-R-C	411.8	1085.2	2633.8	5146.8	7659.1
LFF-C-C	418.4	1602.2	3838.1	6940	10683.8
LFF-C-R	937.6	3101.7	5885	9775.8	13829.6
LFF-R-R	293.7	1233.2	2456.7	4492.5	6817.6
LFF-R-C	725.3	2159.7	4274.9	7354	11891.4

Tables 1 and 2 show the average success rate and average cumulative tardiness of individual algorithms, respectively. The success rate of using the execution cost metric in both stages of agent selection dominates the remaining options for both EDF and LFF task selection orders. As seen, the EDF based approaches tend to perform better compared to their LFF based counterparts. As the number of agents increases, it gives rise to a potentially larger number of conflicts, resulting in a decrease in schedulability performance across all algorithms. The cumulative tardiness increases significantly with an increasing number of agents. As seen, the proposed algorithm dominates all the other options. Due to the good success rate, only a limited number of tasks are unable to meet their deadlines, resulting in a lower cumulative tardiness.

ACKNOWLEDGMENTS

This study is supported under the RIE2020 Industry Alignment Fund – Industry Collaboration Projects (IAF-ICP) Funding Initiative, as well as cash and in-kind contribution from Singapore Telecommunications Limited (Singtel), through Singtel Cognitive and Artificial Intelligence Lab for Enterprises (SCALE@NTU).

REFERENCES

[1] Zhe Chen, Javier Alonso-Mora, Xiaoshan Bai, Daniel D. Harabor, and Peter J. Stuckey. 2021. Integrated Task Assignment and Path Planning for Capacitated Multi-Agent Pickup and Delivery. *IEEE Robotics and Automation Letters* 6, 3 (2021), 5816–5823.

[2] Alessandro Farinelli, Antonello Contini, and Davide Zorzi. 2020. Decentralized Task Assignment for Multi-item Pickup and Delivery in Logistic Scenarios. In *Proceedings of the 19th International Conference on Autonomous Agents and MultiAgent Systems (AAMAS)*. 1843–1845.

[3] Florian Grenouilleau, Willem-Jan van Hoeve, and John N. Hooker. 2019. A Multi-Label A* Algorithm for Multi-Agent Pathfinding. In *Proceedings of the 29th International Conference on Automated Planning and Scheduling (ICAPS)*. 181–185.

[4] Jiaoyang Li, Andrew Tinka, Scott Kiesel, Joseph W. Durham, T. K. Satish Kumar, and Sven Koenig. 2020. Lifelong Multi-Agent Path Finding in Large-Scale Warehouses. In *Proceedings of the 19th International Conference on Autonomous Agents and MultiAgent Systems (AAMAS)*. 1898–1900.

[5] Minghua Liu, Hang Ma, Jiaoyang Li, and Sven Koenig. 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)*. 1152–1160.

[6] Hang Ma, Wolfgang Hönig, T. K. Satish Kumar, Nora Ayanian, and Sven Koenig. 2019. Lifelong Path Planning with Kinematic Constraints for Multi-Agent Pickup and Delivery. In *Proceedings of the 33rd AAAI Conference on Artificial Intelligence (AAAI)*. 7651–7658.

[7] Hang Ma, Jiaoyang Li, T. K. Satish Kumar, and Sven Koenig. 2017. Lifelong Multi-Agent Path Finding for Online Pickup and Delivery Tasks. In *Proceedings of the 16th International Conference on Autonomous Agents and MultiAgent Systems (AAMAS)*. 837–845.

[8] Hang Ma, Glenn Wagner, Ariel Felner, Jiaoyang Li, T. K. Satish Kumar, and Sven Koenig. 2018. Multi-Agent Path Finding with Deadlines. In *Proceedings of the 27th International Joint Conference on Artificial Intelligence (IJCAI)*. 417–423.

[9] R. Morris, C. S. Pasareanu, K. Luckow, W. Malik, H. Ma, T. K. S. Kumar, and S. Koenig. 2016. Planning, Scheduling and Monitoring for Airport Surface Operations. In *Proceedings of the AAAI Workshop on Planning for Hybrid Systems*.

[10] Oren Salzman and Roni Stern. 2020. Research Challenges and Opportunities in Multi-Agent Path Finding and Multi-Agent Pickup and Delivery Problems. In *Proceedings of the 19th International Conference on Autonomous Agents and Multi-Agent Systems* (Auckland, New Zealand) *(AAMAS '20)*. International Foundation for Autonomous Agents and Multiagent Systems, 1711–1715.

[11] Xiaohu Wu, Yihao Liu, Xueyan Tang, Wentong Cai, Funing Bai, Gilbert Khonstantine, and Guopeng Zhao. 2021. Multi-Agent Pickup and Delivery with Task Deadlines. In *Proceedings of the 20th IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology (WI-IAT)*.

[12] 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.