Human-On-The-Loop Multi-Robot Demand-Aware Task Scheduling: Benchmarking and Analysis with a Mixed Reality User Study

Ajay Kumar Sandula[1], Rajatsurya M[1], Arushi Khokhar[2], Debasish Ghose[1] and Pradipta Biswas[1]

***Abstract*— Scheduling tasks for multiple robots with diverse capabilities can be challenging. This challenge arises as these robots collaborate under precedence constraints, especially when environmental demands require certain robots to handle heavier workloads. This complexity is compounded when human intervention is needed to allocate limited resources. To address this, we introduce the Demand-Aware multi-Robot Task Scheduling (DARTS) approach, employing a multi-agent multi-armed bandit framework to estimate environmental demand. We conducted a mixed reality user study to investigate how human perception influences task scheduling and personalize the resource allocation process. To assess our approach's performance, we conducted a comprehensive benchmarking analysis focusing on multi-robot planning and scheduling. Experimental results demonstrate that the proposed DARTS approach outperforms rate-monotonic scheduling and is comparable to state-of-the-art scheduling methods in terms of robot waiting time, task completion time, and demand-aware performance metrics. Our findings of user study indicate that human task schedulers demonstrated improved performance in scenarios involving autonomous agent assistance compared to situations with lesser autonomy of the robotic systems.**

I. INTRODUCTION

Multi-robot systems are effective in enhancing task efficiency and scalability by distributing the workload among multiple robots, enabling parallel execution of tasks, and facilitating collaboration and coordination among the robots. In recent years such systems gained popularity due to their potential in various applications such as emergency rescue and care, surveying, manufacturing and logistics as reported by Rajan et al. [1]. Nonetheless, these robotic systems may not possess the robustness to autonomously handle all tasks, necessitating either human involvement within the loop or human supervision, as elucidated by Nahavandi et al. [2]. Achieving efficient task planning, scheduling, motion planning, and control while preserving mission objectives can be intricate, especially in the context of multi-robot systems that involve human intervention, which brings its own set of distinctive challenges. Goodrich et al. [3] explored the spectrum of human-robot interactions and the intricacies within human-supervised and human-in-the loop robotic systems.

In a mixed reality user study involving eight participants (six male and two female), we investigated the influence of human perception on human-supervised resource allocation. This study encompassed various scenarios, including independent scheduling, autonomous agent assistance, and simulated scheduling environments. The user study enables us to achieve a deeper understanding of human preferences, facilitating further customization of the resource allocation process. Another significant contribution is the introduction of the Demand-Aware multi-Robot Task Scheduling (DARTS) approach, which addresses multi-robot task scheduling by taking into account environmental demands and precedence constraints, promoting effective collaboration and coordination among robots. Environmental demand estimation is achieved through a multi-agent multi-armed bandit approach, as elucidated by Sandula et al. [4]. In this context, the DARTS approach utilizes the bandit solver to frame the task scheduling problem as a multi-armed bandit problem, considering the precedence constraints among robot coalitions.

Human-robot interaction can be classified into two primary categories. The first category is human-in-the-loop (HIL) [5]–[9], where a human operator actively collaborates with or controls the robot during task execution. The second category is human-on-the-loop (HOL) [10]–[12] interaction, where a human operator supervises the task performed by the robotic system. Our research centers on HOL robotic systems, with a particular emphasis on human task scheduling for multi-robot systems. The literature on robotics has investigated task scheduling in various scenarios for both single [13]–[15] and multi robot [7], [16]–[21] systems. Previous studies have investigated task scheduling for a single and multiple robot task scheduling in applications industrial automation, assembling, and palletizing tasks etc. Dhanaraj et al. [7] have proposed a human-in-the-loop task scheduling with focus on fault recovery. Zhang et al. [21] proposed heuristic based task scheduling approaches for a generalized scenario where robots form a coalition to accomplish a task. However, a human-on-the-loop demand aware scheduling strategy considering the precedence constraints is not investigated in the current literature. Therefore, to address this gap, we introduce a novel approach: DARTS - a Demand-Aware multi-Robot Task Scheduling algorithm. This algorithm leverages a multi-agent multi-armed bandit [22] framework, as elucidated by Sandula et al. [4], to estimate environmental demand.

In summary, our work presents two significant contributions: firstly, the introduction of a novel approach called DARTS (Demand-Aware multi-Robot Task Scheduling) for human-on-the-loop multi-robot task scheduling, which incorporates environmental demand and includes a thorough benchmarking analysis; secondly, a mixed reality user study

*This work was not supported by any organization

[1] [ajaykumars,dghose,pradipta]@iisc.ac.in, [rajatsurya.m,arushikhokhar]@gmail.com Associated with Department of Cyber-Physical Systems Indian Institute of Science Bangalore, Karnataka, 560012, India

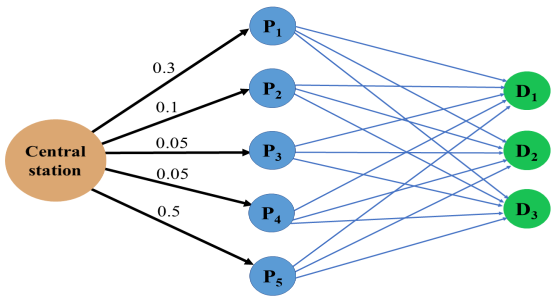

Fig. 1: Probabilities at which tasks have been generated. A central station receives the packages and then transfers them to the pickup points based on the type of packages.

aimed at exploring the impact of human perception on human-supervised task scheduling.

II. Problem statement

We consider a set of robots $R = [r_1, r_2, \ldots, r_i, \ldots, r_n]$, each of which acts as an independent agent. The robots are needed to execute a task or a sub-task. A set of tasks is given by $T = [T_1, T_2, \ldots, T_j, \ldots]$. We assume that all the tasks and the respective sub-tasks within the tasks are non-pre-emptive. Non-pre-emptive tasks are those tasks which cannot be partially accomplished. Complete attention of the robot is needed until it is executed.

A. Multi-robot task scheduler with demand awareness

Let us define an ordered coalition as an ordered set containing elements from the set R. We further define $P(R, k)$ as the collection of all possible ordered sets containing precisely k distinct elements from the set R. Thus, for any set κ, we have $\kappa = P(R, 1) \cup P(R, 2) \cup \ldots \cup P(R, |R|)$, signifying the encompassing set of all possible ordered sets with distinct elements from R. For any element $C \subset \kappa$, represented as $C = (r_a, r_b, \ldots, r_*)$, it can be regarded as an ordered coalition of robots arranged in a specific order. In the context of a task T_j, if it can be accomplished by an ordered coalition C, then the robots $r_a, r_b, \ldots, r_*$ can execute their corresponding sub-tasks in the specified sequence. Each task T_j can be achieved either by a single ordered coalition or by multiple such ordered coalitions. It is, however, crucial to acknowledge the precedence of sub-tasks among the robots in completing the overall task. The ensemble of ordered sets $C_j \subseteq \kappa$ delineates the full range of ordered coalitions capable of accomplishing task T_j.

In pickup-dispatch tasks, a coalition consists of both a mobile robot and a fixed-base robot. In this scenario, the mobile robot is responsible for transporting a load to the fixed-base robot, which subsequently conducts the pick-and-place operation. For instance, consider a pickup-dispatch task denoted as 'T_j,' which can only be executed by a unique coalition comprising a fixed-base robot $[r_a]$ and mobile robots $[r_b, r_c, r_d]$. The set encompassing all conceivable ordered coalitions capable of accomplishing task 'T_j' can be represented as $C_j = [(r_b, r_a), (r_c, r_a), (r_d, r_a)]$. In this particular context, either mobile robot r_b, r_c, or r_d must undertake the sub-task of transporting the load to a dispatch location, where a fixed-base robot can then execute the pick-and-place operation to relocate the load within its designated workspace. The equitable allocation of task T_j among all the potential coalitions in C_j can be achieved through an auction-based approach, such as the one outlined in [23].

After the robots were allotted tasks, it is imperative to establish a well-defined sequence for task execution, which is contingent upon task priorities in order to attain the desired objective. In the case of the exemplified scenario concerning the dispatch of pickup tasks, a key consideration arises when a fixed-base robot is tasked with the relocation of cargo items originating from various mobile robots. This necessitates the allocation of priorities among the sub-tasks associated with the base robot's operations. In the context of a pickup-dispatch scenario, involving pickup points $(P_1, P_2, P_3, P_4, P_5)$ and drop points (D_1, D_2, D_3) for the task, we define a set of possible tasks as T. Each task is denoted as $T = [T_1, T_2, \ldots, T_{15}]$, with T_1 representing the pickup-dispatch task originating from pickup point P_1 and destined for drop point D_1. Likewise, we define tasks such as $T_1 =< P_1, D_1 >$, $T_2 =< P_1, D_2 >$, and so forth, encompassing all pickup-dispatch combinations. Now, we further consider that, all these tasks are generated based on a probability distribution. Figure 1 describes the probabilities according to which the tasks are being generated from the 'central station' based on the demand from the environment. We assumed that from each pickup point only a specific type of load can be picked up, which can be executed by only a unique set of mobile robots. For example, a load from pickup point P_1 can be picked up by only a specific set of mobile robots based on the limitation of containers and load-carrying capacity. However, at the drop point, the fixed-base resource can pick and place loads from all the mobile robots that are approaching a drop point. The mobile robots which collaborate on the pickup-dispatch task with fixed-base robots and carry loads from P_i are denoted by Ψ_i. Hence Ψ_i denotes a set of homogeneous robots which are capable of executing the sub-task of moving the package from pickup point P_i and collaborate at a drop point for any given task $< P_i, D_j >$. The scheduling is only needed at the drop point when multiple mobile robots among Ψ_{1-5} approach the drop point. The goal is to schedule the tasks in a way which reduces the waiting time for the mobile robots executing the tasks that are generated with higher probability.

B. Scheduling by Human Users in Mixed Reality

We conducted a user study in a mixed reality environment to observe how human perception of the environment affects the task scheduling. The result of this user study can be leveraged to further customize the resource allocation process. In the context of human-on-the loop, multi-robot task scheduling we envision a scenario where human users are tasked with performing actions in a multi-armed bandit [22] environment. This setup presents intriguing challenges that can be expressed mathematically as follows.

Manuscript 3680 submitted to 2024 IEEE International Conference on Robotics and Automation (ICRA). Received September 15, 2023.

A user conducts a series of trials, denoted by $n(> 0)$, involving $k(> 0)$ reward stations. Each station generates rewards based on a probability distribution. During each trial t, the user selects an arm a_t from the set 1, 2, ..., k and receives a reward $r_{a_t,t}$ sampled from the corresponding arm's probability distribution. The reward from arm i during trial t is represented as $r_{i,t}$. The true mean reward for arm i is μ_i for $i \in$ 1, 2, ..., k, and the standard deviation of arm i's reward distribution is σ_i. These reward distributions are independent and identically distributed (i.i.d.) across trials. The cumulative reward obtained after t trials is given as $R_t = \sum_{i=1}^{t} r_{a_i,i}$, where a_i is the arm selected by the user in trial i, and $r_{a_i,i}$ is the reward from selecting arm a_i at trial i. The expected total reward after n trials is denoted as $E[R_n] = \sum_{i=1}^{n} E[r_{a_i,i}]$. Here, $E[r_{a_i,i}]$ represents the expected value of the reward from selecting arm a_i in trial i. The user's objective is to maximize this expected cumulative reward across the n trials by choosing the arm with the highest expected reward at each trial. In essence, at each trial t, the user selects arm a_t such that $a_t = \arg\max_{i\in\{1,2,...,k\}} \hat{\mu}_i(t-1)$, where $\hat{\mu}_i(t-1)$ is the estimate of the true mean reward of arm i based on the rewards obtained up to trial $t-1$.

III. DARTS - Demand Aware multi-Robot Task Scheduling

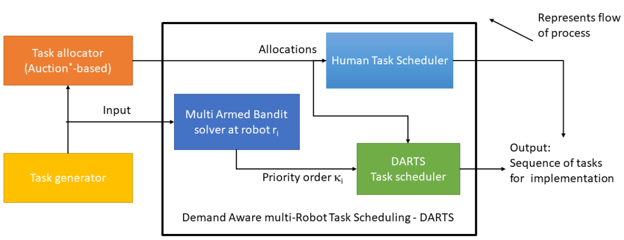

Fig. 2: Individual modules of the proposed architecture

In this section we discuss the proposed DARTS approach for multi-robot task scheduling considering the demand from the environment. Figure 2 represents the overall architecture of the proposed DARTS approach. The task generator module simulates tasks using predefined probabilities to mimic environmental demand. It's important to note that the other modules are unaware of these probability distributions. There are two important components involved in the proposed DARTS algorithm. To observe the demand from the environment, we have utilized the integration of multi-agent multi-armed bandit approach suggested by Sandula et al. [4]. A combination of epsilon-greedy and Thompson sampling algorithms were utilized to solve the bandit problem.

A. Bandit solver

We utilized a multi-agent multi-armed bandit [4] solver at each robot to determine the demand and subsequently establish the priority order. In cases where a robot, denoted as r_i, collaborates with other robots and must await their sub-task completion, we deploy a bandit solver. Thus, there exists a group of homogeneous robots, represented as $\psi_1, \psi_2, \dots, \psi_k$, with which r_i collaborates. A bandit solver is then employed at robot r_i to assess the demand and derive a priority order κ_i among ψ_{1-k}. If the agents choose the same group where the task is allocated, we give the agent a reward '1' otherwise '0'. The agents share the information obtained in this round, that is, the choice (Ψ_i) and reward ('1' or '0') with each other. The agents collaborate among themselves instead of competing to solve the bandit problem. The bandit algorithm used is a combination of Thompson sampling and ϵ-greedy algorithms. 'Beta' distribution is used to estimate the expected reward of the arms Ψ_{1-k}.

Algorithm 1 DARTS Task Scheduler ($[T_i, R_i]$,κ)

if Task is being executed **then**
  Return
end if
$\lambda = Empty\_list$
while $\kappa \neq \phi$ **do**
  $\psi_j = \kappa \rightarrow pop$
  for $k < length(T_i)$ **do**
    if $R_i[k] \rightarrow robot\_type = \psi_j$ **then**
      $Append(R_i[k]) \rightarrow \lambda$
    end if
  end for
end while
if $\lambda \neq \phi$ **then**
  $cur\_robot\_task = \lambda \rightarrow pop$
  execute T $\rightarrow (cur\_robot\_task)$
  $T_i \rightarrow pop(T_i \rightarrow (cur\_robot\_task))$
  Update current status (task is being executed)
end if

B. DARTS task scheduler

Algorithm 1 presents the pseudocode for the proposed DARTS algorithm at robot r_i. After establishing the priority order among ψ_{1-k} for robot r_i as $\kappa_i = \psi_a, \psi_b, \dots$, this module determines the sequence of tasks to be executed. The algorithm takes input in the form of task allocations, denoted as [T, R], where 'T' represents the set of tasks allocated to their corresponding indices in set 'R'. As an example, the set of tasks T_i, where T_i $\in$ T and is the 'ith' element in set T, signifies the collection of allocated tasks for robot r_i, which is the 'ith' element in set 'R'. The scheduler ensures that computation is avoided when a task is already in progress to reduce unnecessary processing. It computes the sequence for the next task only if the robot has completed its current task. The scheduler aims to arrange the tasks such that all tasks in T_i belonging to robot coalitions ψ_a are sequenced together, followed by tasks according to ψ_b, and so on as per the priority order κ_i.

C. Mixed Reality User Study

We established a mixed reality environment by combining Microsoft HoloLens, Unity, and ROS Bridge. In this

Fig. 3: Mixed Reality setup for Independent Scheduling.

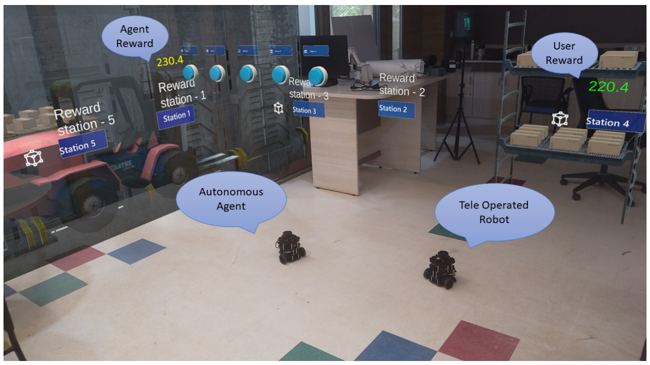

Fig. 4: Mixed Reality setup for Scheduling with Autonomous Agent Assistance.

environment, we introduced three distinct scenarios aimed at investigating the influence of human perception on task scheduling. These scenarios encompassed (1) Single-robot scheduling, (2) Multi-robot scheduling, and (3) Simulated scheduling with zero robots, each of which is described below. We collected data from all the participants across all the scenarios.

1) Participants: We enlisted 8 participants (6 males, 2 females) from our university, with an average age of 24.37 years and a standard deviation of 2.99. None of the participants had color blindness, and they all granted the required permissions and consent for the study.

2) Setup: In case 1, known as single-robot scheduling (depicted in Figure 3), participants are assigned the operation of a single robot and tasked with deducing rewards to maximize their collection. Participants control the robots using the virtual buttons, and when a robot reaches a designated station, the reward obtained is revealed in green text atop the station. During this phase, we measure the robot's travel distance, record task completion times, and calculate the rewards achieved in each iteration. The rewards in each iteration are calculated based on specific probability distributions unique to each station, all of which are kept confidential from the participants. This information remains undisclosed throughout the study.

In case 2, known as multi-robot scheduling, as shown in the Figure 4 we introduce an autonomous agent into the environment. Here, participants observe the actions of the autonomous agent and have the option to incorporate its strategies into their own task scheduling. This stage serves to compare participant performance when acting independently versus when assisted by the autonomous agent. The autonomous agent utilizes the DARTS algorithm for task scheduling. In this environment, two rewards are displayed: one collected by the autonomous agent (yellow colour) and the other by the user-controlled robot. Participants can utilize the agent's reward data to optimize their scheduling processes.

In the case 3, referred to as simulated scheduling with zero robots, the mixed reality setup remains identical to the previous cases. However, it lacks physical robots, and participants instantly observe rewards upon interacting with virtual buttons.

3) Design: The user interface, depicted in figures 3 and 4, features five virtual buttons for task scheduling. For cases 1 and 2, each button when pressed, the tele-operated robot executes a pickup-dispatch task. The robot autonomously navigates first to the base location and subsequently to the corresponding station. Upon task completion, a reward is displayed on the respective station. For the tele-operated robot, the reward is shown in green text, while for the autonomous robot, it appears in yellow text. In the simulated scheduling scenario (case 3), rewards are instantly shown on the corresponding station when users press the virtual button, as no robots are involved.

4) Procedure: Participants received an introduction to the study's objectives. Their goal was to maximize rewards. We recorded each participant's total reward, task execution time, and robot travel distance for all scenarios. The outcomes are detailed in Section IV.

IV. Results and Discussion

In this Section, we present the results obtained by testing the DARTS algorithm through comprehensive benchmarking study by deploying a combination of planning and scheduling algorithms. In subsection IV-A we explain the experimental setup. In subsection IV-B present the results and analyze the total task completion times and waiting times of the robots. In subsection IV-C we present our observing of demand aware performance of the proposed algorithm in comparison with other scheduling algorithms.

A. Experimental setup

We conducted the experiment in a ROS-Gazebo simulation environment which can be visualized in the figures 5 and 6. A total of 20 mobile robots are simulated in a warehouse-type of environment. The mobile robots are represented as $M = [M_1, M_2, ..., M_{20}]$. Each mobile robot is capable of carrying a specific type of load. Five points are chosen as pickup points, and three points are chosen as drop points. Robots $M_{[1-4]}$, $M_{[5-8]}$, $M_{[9-11]}$, $M_{[12-14]}$ and $M_{[15-20]}$ carry the loads only from pickup points P_1, P_2, P_3, P_4, and P_5, respectively. The tasks get generated from the pickup points with a Bernoulli distribution of 0.3, 0.1, 0.05, 0.05 and 0.5, respectively from P_1, P_2, P_3, P_4, and P_5 respectively. The following table I shows the probability at which tasks are

Manuscript 3680 submitted to 2024 IEEE International Conference on Robotics and Automation (ICRA). Received September 15, 2023.

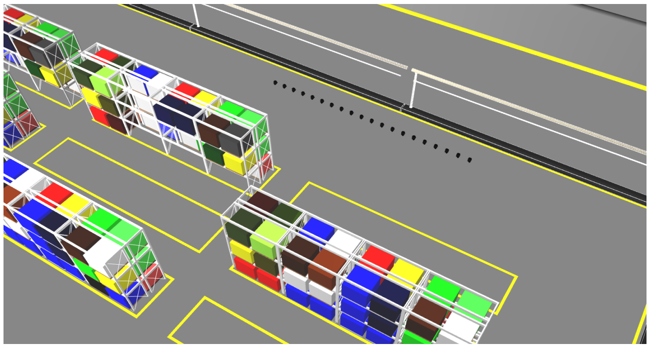

Fig. 5: Simulation of a fleet of mobile robots within a warehouse environment using ROS-Gazebo architercture.

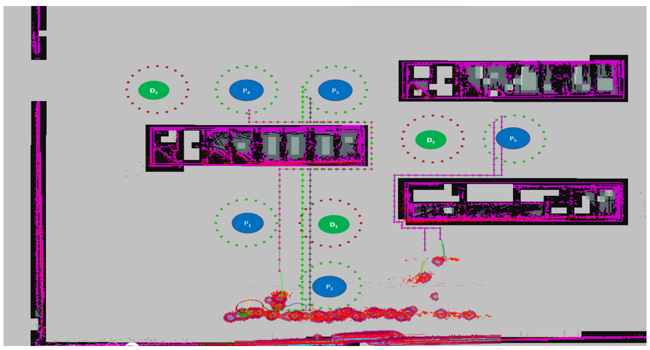

Fig. 6: ROS-Gazebo simulation in real-time displaying the trajectories of mobile robots towards their current goals.

generated from pickup points to corresponding drop points. The mobile robots move autonomously in the environment while executing their sub-tasks. We've assumed a standard collaboration time of 55 seconds at the drop point for the fixed-base robot to transfer the load from the mobile robot. The mobile robots often interfere with each other when executing tasks and navigating to their designated goal locations. So, we utilized three different multi-robot path planning algorithms A* [24], CBS [25] and ECBS [26]. This facilitated a comprehensive comparison of multiple planning and scheduling strategies.

	D1	D2	D3
P1	0.750	0.125	0.125
P2	0.800	0.100	0.100
P3	0.100	0.800	0.100
P4	0.100	0.800	0.100
P5	0.500	0.250	0.250

TABLE I: Table containing probability distribution to drop points from pickup points

B. Performance metrics and analysis

We conducted experiments encompassing all three path planning algorithms (A-star, CBS, ECBS) and four scheduling algorithms (Min-Interference [21], Rate-Monotonic [27], DARTS, First-Come-First-Serve [27]). In each simulation scenario, we ensured consistency by employing the same set of generated tasks, enabling a fair comparison among the generated scenarios. Tables II and III present the comprehensive task completion and robot waiting times for a total of 200 tasks.

$Scheduler(\downarrow)\&Planner(\rightarrow)$	astar	cbs	ecbs
Min-Interference	36931.31	40354.20	41131.31
Rate-monotonic	37329.66	37329.66	36296.45
DARTS	36958.35	40518.38	41387.86
FCFS	36982.85	39487.45	39504.90

TABLE II: Overall task completion times in seconds for various combinations of planning and scheduling algorithms.

$Scheduler(\downarrow)\&Planner(\rightarrow)$	astar	cbs	ecbs
Min-Interference	14455.12	14008.22	15062.40
Rate-monotonic	16434.75	16014.56	15193.65
DARTS	14884.81	16331.97	15499.36
FCFS	15293.10	15513.45	15295.41

TABLE III: Overall robot waiting times in seconds for various combinations of planning and scheduling algorithms.

Demand aware performance index	astar	cbs	ecbs
Min-Interference	4502.37	4318.52	4744.06
Rate-monotonic	4990.88	4934.57	4762.28
DARTS	4657.79	4869.14	4813.06
FCFS	4733.10	4715.49	4761.89

TABLE IV: Demand aware performance index in seconds for various combinations of planning and scheduling algorithms.

We observe that the robot waiting times contribute a significant portion in the total task completion times. Hence it is important to optimize the tasks which have higher demand from the environment. The Demand-Aware Performance Index (DAPI) is computed by multiplying the robot waiting times by the probability at which the corresponding task is simulated. The table III presents the DAPI results for all 12 scenarios in the benchmarking study. We noted that the proposed scheduling algorithm outperformed rate-monotonic scheduling in all planning scenarios. Additionally, the algorithm demonstrated commendable performance, on par with state-of-the-art existing multi-robot task scheduling methods.

C. Interpretation of User Study

We conducted the user study with three distinct conditions: (1) Independent scheduling(no autonomous Agent), (2) Scheduling with autonomous agent assistance and (3) Simulated scheduling environment with no robots as explained in Section III-C. In total, each participant was responsible for scheduling 45 tasks: 15 tasks independently in case 1, 15 tasks with the assistance of the autonomous agent in case 2 and additionally, participants completed 15 tasks in a simulated environment where there was no robot involved, and the reward was instantly observed by the user. Our analysis encompassed three distinct aspects

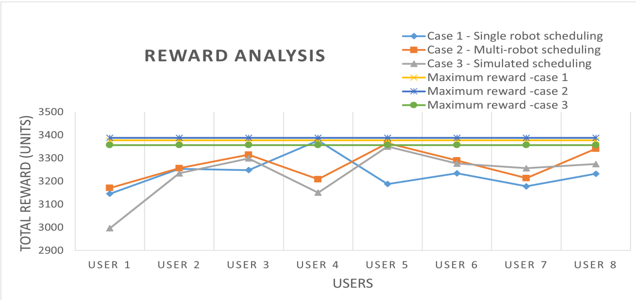

Fig. 7: Reward Analysis for all the three cases

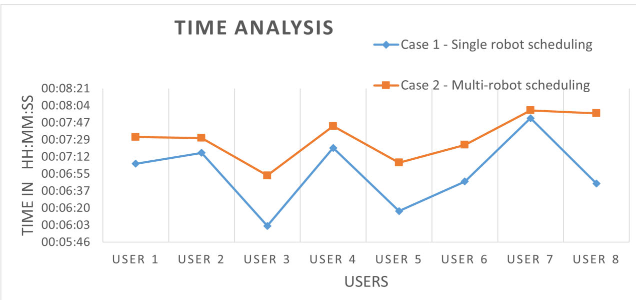

Fig. 8: Time analysis

From the graph presented in figures 7 and 9, it becomes evident that users generally prioritized lower rewards as a means to minimize the overall distance traveled by the robots, particularly in case 1. The exception to this trend was 'user 4.' However, in scenarios where scheduling involved autonomous agent assistance, users exhibited improved performance in terms of rewards and seemed less concerned about the distance traveled by the robots. Additionally, we noted that there was no substantial difference in user performance regarding rewards between cases 1 and 3.

From Figure 8, it's clear that participants favored completing tasks quickly over optimal scheduling in independent scheduling (Case 1). However, in Case 2, with autonomous agent assistance, participants took more time to complete tasks while achieving higher rewards.

Figure 9 reveals scheduling preferences of the participants. In case 1, they sought to minimize the robot's travel distance, deeming it efficient. In contrast, Case 2 presented a different

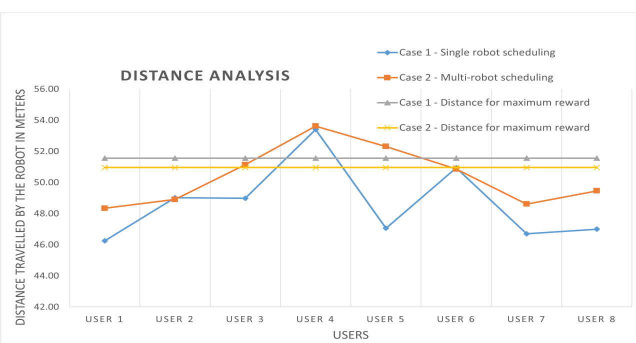

Fig. 9: Distance Analysis

scenario. Here, the autonomous agent employed the DARTS algorithm. Participants shifted their focus toward maximizing rewards. They made this adjustment after observing the rewards acquired by the autonomous agent, even if it meant the robot had to cover a greater distance. Remarkably, in case 2, the robot covered significantly more distance compared to case 1, except for 'user 4'.

We conducted a one-way ANOVA to compare users' "total reward" across three independent variables: cases 1, 2, and 3. This analysis didn't reveal any statistically significant differences. However, when we performed paired T-tests on other dependent variables, namely "total task completion time" and "total distance traveled by the tele-operated robot," we uncovered noteworthy distinctions among the independent variables cases 1 and 2. A paired $t-test$ score for "total task completion time" indicate that the users took higher time to complete the tasks in case 2, while case 1 is significantly lower ($t = -4.693$, $p < 0.050$, Cohens $d = -1.369$) than case 2. Also, the paired $t-test$ score for "total distance traveled by the tele-operated robot" indicate that the tele-operated robot travelled significantly higher distance complete the tasks in case 2, while case 1 is significantly lower ($t = -2.759$, $p < 0.050$, Cohens $d = -0.787$) than case 2. In conclusion, the data suggests that there are significant differences between the means of distances traveled by the tele-operated robot and the time taken by users across the different cases.

We conclude the results as follows. Participants initially prioritized task completion speed and minimizing travel distance when scheduling tasks independently. However, when an autonomous agent was introduced, task scheduling improved, resulting in higher rewards but with increased time consumption. Notably, one user had a distinct approach. Participants also learned that minimizing robot travel distance didn't always yield the best task scheduling outcomes, a finding supported by statistical analysis.

V. Future work & Conclusion

We introduced the DARTS algorithm, a human-on-the-loop multi-robot task scheduling approach that considers environmental demands and precedence constraints among the robots. Extensive benchmarking demonstrated that DARTS performed competitively with state-of-the-art algorithms and outperformed the rate-monotonic scheduling approach in demand-aware performance, task completion times, and robot waiting times. The mixed reality user study highlighted the advantages of including an autonomous agent for human-on-the-loop multi-robot task scheduling. Future work will focus on demand-aware multi-robot task allocation in conjunction with human-on-the-loop supervision.

References

[1] Rajan, J., Shriwastav, S., Kashyap, A., Ratnoo, A. and Ghose, D., 2021. Disaster management using unmanned aerial vehicles. In Unmanned Aerial Systems (pp. 129-155). Academic Press.

[2] Nahavandi, S., 2017. Trusted autonomy between humans and robots: Toward human-on-the-loop in robotics and autonomous systems. IEEE Systems, Man, and Cybernetics Magazine, 3(1), pp.10-17.

[3] Goodrich, M.A. and Schultz, A.C., 2008. Human–robot interaction: a survey. Foundations and Trends® in Human–Computer Interaction, 1(3), pp.203-275.
[4] AK Sandula, A Khokhar, D Ghose and P Biswas ”Demand-Aware Multi-Robot Task Scheduling with Mixed Reality Simulation,” 2023 32nd IEEE International Conference on Robot and Human Interactive Communication (RO-MAN), Busan, South Korea, 2023
[5] Papallas, R. and Dogar, M.R., 2020, May. Non-prehensile manipulation in clutter with human-in-the-loop. In 2020 IEEE International Conference on Robotics and Automation (ICRA) (pp. 6723-6729). IEEE.
[6] Manyar, O.M., McNulty, Z., Nikolaidis, S. and Gupta, S.K., 2023, May. Inverse Reinforcement Learning Framework for Transferring Task Sequencing Policies from Humans to Robots in Manufacturing Applications. In 2023 IEEE International Conference on Robotics and Automation (ICRA) (pp. 849-856). IEEE.
[7] Dhanaraj, N., Narayan, S.V., Nikolaidis, S. and Gupta, S.K., 2023, May. Contingency-Aware Task Assignment and Scheduling for Human-Robot Teams. In 2023 IEEE International Conference on Robotics and Automation (ICRA) (pp. 5765-5771). IEEE.
[8] Ng, E., Liu, Z. and Kennedy, M., 2023, May. It takes two: Learning to plan for human-robot cooperative carrying. In 2023 IEEE International Conference on Robotics and Automation (ICRA) (pp. 7526-7532). IEEE.
[9] Ford, C.J., Li, H., Lloyd, J., Catalano, M.G., Bianchi, M., Psomopoulou, E. and Lepora, N.F., 2023. Tactile-Driven Gentle Grasping for Human-Robot Collaborative Tasks. arXiv preprint arXiv:2303.09346.
[10] Sheridan, T.B., 2021. Human supervisory control of automation. Handbook of Human Factors and Ergonomics, pp.736-760.
[11] Al-Hussaini, S., Gregory, J.M. and Gupta, S.K., 2023. Generating Task Reallocation Suggestions to Handle Contingencies in Human-Supervised Multi-Robot Missions. IEEE Transactions on Automation Science and Engineering.
[12] Al-Hussaini, S., Gregory, J.M., Guan, Y. and Gupta, S.K., 2020, October. Generating alerts to assist with task assignments in human-supervised multi-robot teams operating in challenging environments. In 2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) (pp. 11245-11252). IEEE.
[13] Stavridis, S. and Doulgeri, Z., 2018, October. Bimanual assembly of two parts with relative motion generation and task related optimization. In 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) (pp. 7131-7136). IEEE.
[14] Méndez, J.B., Perez-Vidal, C., Heras, J.V.S. and Pérez-Hernández, J.J., 2020. Robotic pick-and-place time optimization: Application to footwear production. IEEE Access, 8, pp.209428-209440.
[15] Szczepanski, R., Erwinski, K., Tejer, M., Bereit, A. and Tarczewski, T., 2022. Optimal scheduling for palletizing task using robotic arm and artificial bee colony algorithm. Engineering Applications of Artificial Intelligence, 113, p.104976.
[16] Ho, Y.C. and Liu, H.C., 2006. A simulation study on the performance of pickup-dispatching rules for multiple-load AGVs. Computers industrial engineering, 51(3), pp.445-463.
[17] Kalempa, V.C., Piardi, L., Limeira, M. and de Oliveira, A.S., 2021. Multi-Robot preemptive task scheduling with fault recovery: A novel approach to automatic logistics of smart factories. Sensors, 21(19), p.6536.
[18] Kousi, N., Koukas, S., Michalos, G. and Makris, S., 2019. Scheduling of smart intra–factory material supply operations using mobile robots. International Journal of Production Research, 57(3), pp.801-814.
[19] Gao, T., Tian, Y. and Bhattacharya, S., 2021, September. Refuel scheduling for multirobot charging-on-demand. In 2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) (pp. 5825-5830). IEEE.
[20] Bischoff, E., Meyer, F., Inga, J. and Hohmann, S., 2020, October. Multi-robot task allocation and scheduling considering cooperative tasks and precedence constraints. In 2020 IEEE International Conference on Systems, Man, and Cybernetics (SMC) (pp. 3949-3956). IEEE.
[21] Zhang, Y. and Parker, L.E., 2013, May. Multi-robot task scheduling. In 2013 IEEE International Conference on Robotics and Automation (pp. 2992-2998). IEEE.
[22] Lattimore, T. and Szepesvári, C., 2020. Bandit algorithms. Cambridge University Press.
[23] Gerkey, B.P. and Mataric, M.J., 2002. Sold!: Auction methods for multirobot coordination. IEEE transactions on robotics and automation, 18(5), pp.758-768.
[24] Hart, P., Nilsson, N. Raphael, B., 1968. A Formal Basis for the Heuristic Determination of Minimum Cost Paths. IEEE Transactions on Systems Science and Cybernetics, 4(2), pp.100–107. Available at: https://doi.org/10.1109/tssc.1968.300136.
[25] Sharon, G., Stern, R., Felner, A. and Sturtevant, N.R., 2015. Conflict-based search for optimal multi-agent pathfinding. Artificial Intelligence, 219, pp.40-66.
[26] Barer, M., Sharon, G., Stern, R. and Felner, A., 2014. Suboptimal variants of the conflict-based search algorithm for the multi-agent pathfinding problem. In Proceedings of the International Symposium on Combinatorial Search (Vol. 5, No. 1, pp. 19-27).
[27] Krishna, C. M. (C. M. Shin, K. G. (1997) Real-time systems / C.M. Krishna, Kang G. Shin. New York: McGraw-Hill.