Energy Efficient Multi-Robot Task Allocation Constrained by Time Window and Precedence

Lixuan Zhang*, *Graduate Student Member, IEEE,* Jianzhuang Zhao*, *Graduate Student Member, IEEE,* Edoardo Lamon, *Member, IEEE,* Yabin Wang, *Graduate Student Member, IEEE,* and Xiaopeng Hong†, *Senior Member, IEEE,*

***Abstract*—To meet the demands in terms of energy-efficient and fast production and delivery of goods, robotic fleets began to populate warehouses and industrial environments. To maximize the profitability of the operations, multi-robot systems are required to coordinate agents avoiding downtime efficiently. In this paper, agent coordination is formulated as a multi-robot task allocation (MRTA) problem with time and precedence constraints. The method capitalizes on a graph method to build a measure graph reflecting the sparsity of tasks and a precedence graph, which includes the task constraints, to group the tasks into batches. A batch solver is provided to obtain the final solutions to the MRTA. In this way, the sustainability and environmental impact of logistics operations can be improved by reducing the number of robots needed to complete tasks and also by assigning tasks closest to the robot location, reducing the amount of time and the total energy required for the robots to complete the job. Extensive experiments on both uniformly distributed and sparse data sets prove the effectiveness of the proposed algorithm compared to state-of-the-art algorithms such as MIP and TePSSI.**

***Note to Practitioners*—This paper was motivated by the problem of minimizing the energy consumption of multi-robot systems in the execution of complex tasks, which requires, in the most general case, the motion of the robot to a target location and further on-site operations. This scenario is particularly relevant in smart, automated warehouses, where mobile robots are repeatedly demanded to store or dispatch goods in a structured environment, where operation duration and future requests are known a priori. The paper formulates this problem by means of a batched multi-robot task allocation (BMRTA) optimization, which can include time windows and precedence constraints jointly. First, the task constraints are encoded into two graphs and then combined to group subtasks together in batches. Then, each batch is solved separately, minimizing the overall energy required to achieve the tasks in the batch. Although the optimality of the solution is ensured only locally, i.e., within the same batch, the task clustering improves the computational efficiency with respect to global approaches, especially in large-sized problems. Experimental results demonstrated that when comparing BMRTA with literature approaches such as MIP and TePSSI, not only the energy consumption but also the total travel distance can be minimized while the total duration of the tasks remains comparable.**

***Index Terms*—Multi-robot coordination, task allocation, energy-efficient.**

L. Zhang, Y. Wang, and X. Hong are with the Faculty of Electronic and Information Engineering at Xi'an Jiaotong University. X. Hong is also with Harbin Insititute of Technology. J. Zhao and E. Lamon are with the Human-Robot Interfaces and Interaction (HRI^2), Istituto Italiano di Tecnologia, Genoa, Italy. J. Zhao is also with the Dept. of Electronics, Information, and Bioengineering, Politecnico di Milano, Milano, Italy and E. Lamon is also with the Dept. of Information Engineering and Computer Science, Università di Trento, Trento, Italy. (e-mail: lxzhang@ieee.org, {jianzhuang.zhao, edoardo.lamon}@iit.it, iamwangyabin@stu.xjtu.edu.cn, hongxiaopengieee.org).

* Lixuan Zhang and Jianzhaung Zhao are co-first authors.

† Corresponding author: Xiaopeng Hong.

I. INTRODUCTION

MULTI-ROBOT systems have become increasingly significant in a wide range of applications, such as surveillance, exploration and rescue, and logistics. In particular, in manufacturing scenarios, robotized autonomous warehouses aim to maximize the velocity of inventory storage and commodity dispatch. The optimal coordination of the agents' operations can result in the enhancement of the supply chain's efficiency and sustainability. Multi-Robot Task Allocation (MRTA) algorithms, regarded as a typical Group Role Assignment with Constraints (GRA+) problem [1], [2], require optimally assigning goals from a given set of tasks (roles) to a team of robotic agents (group), based on specified criteria [3]. These criteria can include but are not limited to, the elapsed time to complete all tasks, the number of robots necessary to perform the task, and the aggregate energy consumption [4]–[6].

Extensive investigation has been carried out in the area of MRTA [5], [7]–[9]. Some of the most prevalent approaches involve mixed integer programming (MIP) [10], auction-based methods [11]–[16], and the genetic algorithms (GA) [17], [18]. However, MRTA remains a formidable and unresolved challenge. Firstly, it has been proved that the MRTA problem is NP-Complete [5], [19], [20]. This means that, when the scale of the problem is large, in terms of the number of simultaneous requests/tasks and team size, it is impossible to solve the problem accurately within a reasonable time [21], [22]. Secondly, practical applications involve the simultaneous execution of multiple tasks, which, in turn, might depend on the success of other ones. Thus, tasks generally are not independent and imply time windows and precedence constraints. On the one hand, the time window constraint requires the tasks to be started and finished in a set of predefined time intervals. On the other hand, the precedence constraint requires tasks to be executed in order. For example, in a robotized cleaning task, it is usually required to perform floor sweeping before mopping, and mopping before buffing. Precedence constraints are important and violating them may lead to unreasonable results in many practical tasks. Nevertheless, most algorithms focus on solving time window constraints based on MIP [10], constraint programming (CP) [23], or temporal sequential single-item

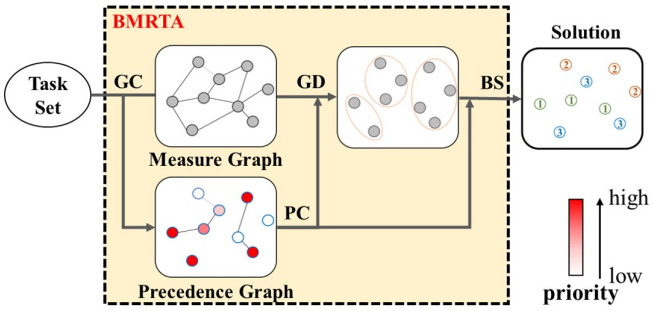


Fig. 1. The framework of the proposed batched MRTA (BMRTA) algorithm, which includes time windows and precedence constraints. The acronyms are "GC": Graph Construction, "GD": Graph Decomposition, "BS": Batch Solver, and "PC": Precedence Constraints. The method constructs two graphs, namely the measure graph and the precedence graph according to the temporal and precedence constraints. Then, graph decomposition splits the tasks into subsets and solves each of them sequentially. The colored numbers within the Solution block suggest the IDs of different robots, while the colors in the precedence graph indicate priorities: the warmer, the most critical.

auction algorithm (TeSSI) [24]. Only a few methods, like the iterated auction with prioritization iterated algorithm (pIA) [24], leverage the precedence constraints. Finally, including energetic considerations in the development of a multi-robot system can yield huge environmental and economic benefits. In such a period of increasing energy demand and transition towards renewable sources, mature MRTA algorithms have to deal with robot energy consumption in the direction of ever more sustainable solutions. However, not quite enough research has brought this aspect to the attention of the scientific community.

Moreover, typically MRTA problems require the inclusion of time windows and precedence constraints jointly (TWPC-MRTA). This is the case of a multi-robot system in an intelligent E-commerce warehouse. Each robot can be assigned specific tasks and routes with time window constraints, which require the pickup and drop-off of goods to be finished during a predefined period. Moreover, procedural constraints, such as specific handling instructions or packing protocols, might require the team to pick and deliver a specific item before another.

In response to the need for satisfying these aspects together, this paper investigates the MRTA problem with time windows and precedence constraints (TWPC-MRTA) jointly from the energy consumption perspective. We first observed that the tasks for multi-robot systems are generally sparse but locally dense in the temporal domain. In the intelligent warehouse example, the robotic action is requested in correspondence with new orders. Once a big new order is placed, several goods might have to be shipped, corresponding to more tasks for the robotic team. The distribution of tasks appears generally sparse but locally dense at the end, which drove us to a divide-and-rule solution. Thus, we proposed a graph-based batch TWPC-MRTA algorithm (which, for simplicity, will be called BMRTA from now on) that divides a complex set of tasks into subsets by clustering two different graphs: a measure graph, which represents the temporal relations between tasks, and a precedence graph, which includes the task-related precedence constraints, and then solves each subset by batch scheduling through MIP. Finally, the fleet energy consumption was modeled and included in each MIP as the cost function that needs to be minimized. The overall framework of the proposed algorithm is shown in Fig. 1.

To summarize, we proposed a solution based on graph construction and decomposition for an MRTA problem, including temporal and precedence constraints, to maximize the task completion rate with a minimum total energy cost of robots. In particular, the contributions of this paper include:

- A graph-based batch solution to the MRTA problem subject to the simultaneous time window and task-precedence constraints;
- The design of the measure graph to quantify the tasks' sparsity in the time domain;
- A task partitioning algorithm based on graph decomposition for batch generation.

II. RELATED WORK

The MRTA problem traces back to the multiprocessor task scheduling problem [21], [25]. While the latter focuses on minimizing the makespan, ensuring real-time performance, MRTA aims at minimizing the total completion time of all tasks, constrained by task execution times and agents' capabilities, communication, and coordination. According to [5], MRTA problems can be briefly divided by three criteria as follows:

- *Single-task robots (ST) or multi-task robots (MT)*: ST means that the robot can perform only one task at the same time, while MT robots can perform multiple tasks.
- *Single-robot tasks (SR) or multi-robot tasks (MR)*: SR means the task needs one robot to complete it, while MR requires multiple robots.
- *Instantaneous assignment (IA) or time-extended assignment (TA)*: IA indicates that each task is assigned only based on the current state of the system while TA considers the potential future state of the system.

Except for the ST-SR-IA problem, MRTA problems are NP-Complete or even NP-hard [19]. A plethora of algorithms have been proposed to solve the MRTA problem, such as the mixed integer program (MIP) [10], [26], [27], constraint programming (CP) [23], heuristic methods [28], [29], meta-heuristic [30], Monte Carlo tree search [31], and auction-based algorithm [12], [13], [15], [24], [32], [33]. Despite the extensive attention that has been given to the literature, limited research has focused on addressing the MRTA problem while concurrently considering both temporal and precedence constraints. Existing state-of-art studies such as CP and TeSSI only focus on one constraint while disregarding the other.

The problem of handling temporal and precedence constraints is addressed also by studies on the Vehicle Routing Problem (VRP) [34]–[36]. For instance, VRP formulations were exploited for planning caregivers' operations in visiting different patients [35]. However, VRP does not include task duration in the problem formulation [34], [37], which is paramount in the long-term assignment of tasks as in MRTA.

To the best of our knowledge, there are currently few studies solving the TWPC-MRTA problem using the MIP [10]

and TePSSI algorithms [33]. The MIP requires a complex mathematical formulation and performs poorly when the scale of the problem becomes large. Differently, the Temporal- and Precedence-constrained Sequential Single-Item auction (TePSSI) [33] can cope more efficiently with large-scale problems. However, the stochastic order of task auction in TePSSI presents a challenge in obtaining robust optimal solutions.

To improve the algorithm performance, recent research introduced clustering to reduce the search space of the MRTA [18], [38], [39]. Nevertheless, in these simplified conditions, coping with task constraints is an open issue.

In MRTA problems, researchers tend to energy-related considerations through the objective function of the optimization problem. In [40], a mobile robot energy model was presented, considering three major energy consumption factors: the sensor system, control system, and motion system, i.e. the different states of robots in MRTA (standby, working, etc.) have different energy consumption. Nevertheless, existing approaches are not able to deal with such an accurate model. For instance, a multihop-based auction algorithm was proposed in [11], which considered the resource consumption of robots in a limited communication range with a fixed absorbed power. Also in [41], a unique heuristic-based energy utility function was exploited by the Iterative Local Search-based (ILS) approach. However, none of them included the time window and task-precedence constraints simultaneously, which is essential for MRTA applications that are targeted by this work.

TABLE I
THE NOTATIONS IN FORMULATION.

Equation	Description
Regular Notation	
$\mathcal{T}$	The set of tasks.
$\mathcal{T}_d$	The set of dummy task.
$\mathcal{R}$	The set of robots.
n_t	The number of tasks in $\mathcal{T}$.
n_r	The number of robots in $\mathcal{R}$.
r_i	The subscript indicates the i-th task in $\mathcal{R}$.
t_j/t_k	The subscript indicates the j-th/k-th task in $\mathcal{R}$.
G_p	The directed acyclic graph that represents the precedence constraints among tasks.
E	The set of the edges in G_p.
$\mathrm{est}(t)$	The earliest start time of task t.
$\mathrm{twl}(t)$	The length of the time window of task t.
$\mathrm{dur}(t)$	The task duration of task t.
$\mathrm{d}(t_j, t_k)$	The distance between task t_j and task t_k.
Decision Variable	
$A_{t_j}^{r_i}$	The binary assignment variables, which is 1 if task t_j is allocated to robot r_i, otherwise is 0.
$Y_{t_j}^{r_i}$	The binary assignment variables, which is 1 if task t_j is the last task allocated to robot r_i, otherwise is 0.
$X_{t_j,t_k}^{r_i}$	The binary ordering variables, which is 1 if task t_j is executed before task t_k in the schedule table of robot r_j, otherwise is 0.
$S(t_j)$	The nonnegative start time of task t_j.
$F(t_j)$	The nonnegative finish time of task t_j.
Z	The nonnegative makespan variable, which is defined as the latest finish time of any task in the corresponding task set.
DIS^{r_i}	The distance traveled by robot r_i.
DUR^{r_i}	The total duration time of tasks finished by robot r_i.
IDT^{r_i}	The total idle time cost by robot r_i, which refers to the time that the robot runs but does nothing.
WEC^{r_i}	The working energy consumption of robot r_i, which refers to the energy consumed for regular work.
SEC^{r_i}	The standby energy consumption of robot r_i, which refers to energy used during the waiting period.
EC	The total energy cost of the robots.
Model Parameter	
M	The parameter of Big M method.

III. PROBLEM FORMULATION

In this section, we formulate the TWPC-MRTA problem from the perspective of energy cost. Specifically, the energy cost model is presented first. The TWPC-MRTA is constructed based on this model later. It is noted that the main notations in this section are summarized in Table I for convenience.

A. Multi-Robot Energy Cost

Given $\mathcal{R}$, a set of n_r robots, the energy cost is defined as the total energy consumed by all robots to complete the tasks. In general, the energy cost of robots is formulated into 2 parts: working state and standby state energy consumption. Working energy (WEC) consumption is the energy consumed for regular work, including traveling to the task location and processing the task. Standby energy (SEC) consumption is the energy used during the waiting period. Therefore, the general total energy cost (EC) of multiple robots can be expressed as:

$$EC = \sum_{r_i \in \mathcal{R}} (WEC^{r_i} + SEC^{r_i}) \tag{1}$$

B. TWPC-MRTA

Given $\mathcal{T}$, a set of n_t tasks with time windows and precedence constraints. For the tasks to be allocated, the time window of task t, where $t \in \{1, \cdots, n_t\}$, is constrained by the *earliest start time* $\mathrm{est}(t)$, the *length of the time window* $\mathrm{twl}(t)$, and the *task duration* $\mathrm{dur}(t)$. The task processed in time interval $[\mathrm{est}(t), \mathrm{est}(t) + \mathrm{twl}(t)]$ and more than $\mathrm{dur}(t)$ time units will be acceptable (or feasible). It is noted that the $\mathrm{twl}(t)$ is indirectly also the maximum processing time of the task. The precedence constraints are usually represented by a directed acyclic graph (DAG) $G_p = (\mathcal{T}, E)$, where the edge in the edge set E refers to the precedence constraints. For instance, if the task pair $(t_1, t_2) \in E$, then the priority of t_1 is higher than t_2 and t_1 must be executed before than t_2 [24], [42], [43]. In addition, we have a robot set $\mathcal{R}$ with n_r homogeneous robots. Each robot in $\mathcal{R}$ has an initial position.

We design an energy-aware model for the TWPC-MRTA problem, where a valid solution consists of an assignment of tasks to robots and a schedule for each robot. These schedules must satisfy the constraints of *duration*, *time windows*, and *precedence*. The objective of this model is to maximize the number of finished tasks and then minimize the energy cost. Thus, we formulate the problem as a bi-objective optimization program, defined as follows.

$$\min \left[EC, \ - \sum_{r_i \in \mathcal{R}, t_j \in \mathcal{T}} A_{t_j}^{r_i} \right]^T \tag{2}$$

s.t.

$$Z \geq F(t_j),\ \forall t_j \in \mathcal{T}, \tag{3}$$

$$\sum_{r_i \in \mathcal{R}} A_{t_j}^{r_i} = 1,\ \forall t_j \in \mathcal{T}_d, \tag{4}$$

$$\sum_{r_i \in \mathcal{R}} A_{t_j}^{r_i} \leq 1,\ \forall t_j \in \mathcal{T}, \tag{5}$$

$$\sum_{t_j \in \mathcal{T}_d} A_{t_j}^{r_i} = 1,\ \forall r_i \in \mathcal{R}, \tag{6}$$

$$\sum_{t_j \in \mathcal{T} \cup \mathcal{T}_d} Y_{t_j}^{r_i} = 1,\ \forall r_i \in \mathcal{R}, \tag{7}$$

$$\sum_{t_j \in \mathcal{T} \cup \mathcal{T}_d, t_j \neq t_k} X_{t_j,t_k}^{r_i} = A_{t_k}^{r_i},\ \forall t_k \in \mathcal{T}, r_i \in \mathcal{R}, \tag{8}$$

$$\sum_{t_k \in \mathcal{T}, t_j \neq t_k} X_{t_j,t_k}^{r_i} + Y_{t_j}^{r_i} = A_{t_j}^{r_i}, \forall t_j \in \mathcal{T} \cup \mathcal{T}_d, r_i \in \mathcal{R}, \tag{9}$$

$$F(t_j) + \mathrm{d}(t_j, t_k) - M(1 - X_{t_j,t_k}^{r_i}) \leq S(t_k), \quad \forall t_j, t_k \in \mathcal{T} \cup \mathcal{T}_d, r_i \in \mathcal{R}, \tag{10}$$

$$0 \leq S(t_j) \leq \infty,\ \forall t_j \in \mathcal{T} \cup \mathcal{T}_d, \tag{11}$$

$$0 \leq F(t_j) \leq \infty,\ \forall t_j \in \mathcal{T} \cup \mathcal{T}_d, \tag{12}$$

$$F(t_j) - S(t_j) \geq \mathrm{dur}(t_j),\ \forall t_j \in \mathcal{T}, \tag{13}$$

$$S(t_k) - F(t_j) \geq 0,\ \forall (t_j, t_k) \in \mathcal{T}_{\mathrm{prec}}, \tag{14}$$

$$\sum_{r_i \in \mathcal{R}} A_{t_j}^{r_i} - \sum_{r_i \in \mathcal{R}} A_{t_k}^{r_i} \geq 0,\ \forall (t_j, t_k) \in \mathcal{T}_{\mathrm{prec}}, \tag{15}$$

$$S(t_j) \geq \mathrm{est}(t_j),\ \forall t_j \in \mathcal{T}, \tag{16}$$

$$F(t_j) \leq \mathrm{est}(t_j) + \mathrm{twl}(t_j) + M(1 - \sum_{r_i \in \mathcal{R}} A_{t_j}^{r_i}),\ \forall t_j \in \mathcal{T}, \tag{17}$$

$$DIS^{r_i} = \sum_{t_j \in \mathcal{T}. t_k \in \mathcal{T}} \mathrm{d}(t_j, t_k) X_{t_j,t_k}^{r_i},\ \forall r_i \in \mathcal{R} \tag{18}$$

$$DUR^{r_i} = \sum_{t_j \in \mathcal{T}} \mathrm{dur}(t_j) A_{t_j}^{r_i},\ \forall r_i \in \mathcal{R} \tag{19}$$

$$IDT^{r_i} = Z - DIS^{r_i} - DUR^{r_i},\ \forall r_i \in \mathcal{R} \tag{20}$$

$$A_{t_j}^{r_i} \in \{0, 1\},\ \forall t_j \in \mathcal{T} \cup \mathcal{T}_d,\ r_i \in \mathcal{R}, \tag{21}$$

$$Y_{t_j}^{r_i} \in \{0, 1\},\ \forall t_j \in \mathcal{T}, r_i \in \mathcal{R}, \tag{22}$$

$$X_{t_j,t_k}^{r_i} \in \{0, 1\},\ \forall t_j \in \mathcal{T} \cup \mathcal{T}_d, r_i \in \mathcal{R}. \tag{23}$$

In this model, a dummy task set $\mathcal{T}_d$ with no time windows or precedence constraints is added at the robot's initial locations for the conciseness of modeling. The precedence constraints set $\mathcal{T}_{\mathrm{prec}}$ consists of the task pairs with precedence constraints between each other. The distance between t_j and t_k is represented by $\mathrm{d}(t_j, t_k)$. The decision variables include the nonnegative start time $S(t_j)$, finish time $F(t_j)$, makespan variable Z, traveled distance DIS^{r_i}, task duration time DUR^{r_i}, and idle time IDT^{r_i}; the binary assignment variables $A_{t_j}^{r_i}$, $Y_{t_j}^{r_i}$ and ordering variables $X_{t_j,t_k}^{r_i}$. The variable $A_{t_j}^{r_i} = 1$ if task t_j is allocated to robot r_i, $Y_{t_j}^{r_i} = 1$ if t_j is the last task allocated to r_i, and $X_{t_j,t_k}^{r_i} = 1$ if t_j is executed before t_k in the schedule table of r_j.

Moreover, equations (4)-(5) indicate that each task only needs to be executed once at most by a robot. Equations (6)-(10) ensure that each robot can only execute one task at one time, and (10) also guarantees that each robot can arrive at the location of the task on time. Equation (13) defines the task *duration* constraints to ensure that a robot shall allocate enough time to finish the task. Equations (14)-(15) and (16)-(17) are the *precedence* constraints and *time windows* constraints, respectively, to ensure the task to satisfy both of these two constraints jointly. For any task pair $(t_j, t_k) \in \mathcal{T}$, equation (14) ensures that the high priority task t_j finishes before the low priority task t_k starts; equation (15) assures all tasks with higher priority than t_k finishes before t_k starts. For the precedence constraints, equations (16)-(17) jointly keep the processed task in its acceptable time interval $[\mathrm{est}(t), \mathrm{est}(t) + \mathrm{twl}(t)]$. Finally, the equations (18)-(20) define the calculation of distance, duration, and idle time respectively, which will be used to calculate energy cost EC in Section V-A.

It is worth mentioning that the model for the multi-robot task allocation problem with precedence constraints (PC-MRTA) [24] shares parts of the equations with our model except (4), (5), (13)-(20). Nevertheless, our model is distinct from the PC-MRTA model, which only focuses on the *precedence* constraint from the following four aspects. Firstly, our model introduces the *time windows* constraint according to the earliest start time and time window length parameters as shown in (16) and (17), which is clearly beyond the scope of PC-MRTA. Secondly, in PC-MRTA, the constraint $\sum_{r_i \in \mathcal{R}} A_{t_j}^{r_i} = 1$ shall be hold for $\forall t_j \in \mathcal{T} \cup \mathcal{T}_d$. It is too strict in practice to satisfy when the time window constraint is valid at the same time. As a result, we replace the constraint with more feasible (4) and (5). Thirdly, when allocating the tasks, it is probable that some tasks can not be allocated and thus fail. Therefore, our model adds the constraint (15) to ensure that the task will not be scheduled when its predecessor fails. Finally, the proposed model is a bi-objective program model, in which we introduce the formulation of multi-robot energy cost and incorporate it into our objective function. The objective in our formulation is to maximize the number of finished tasks and minimize the energy and thus is a two-dimensional vector instead of a scalar, as shown in (2). This makes us significantly different from the PC-MRTA. Together with both of the objective functions, our model prefers to finish more tasks when pursuing minimized energy consumption.

IV. METHODOLOGY

The problem as formulated in Section III is nontrivial to solve. Traditional MRTA problems are usually NP-Complete/Hard problems [19], and thus the time of using mixed integer program straightforwardly will be exponentially increasing along with the number of tasks increases [22]. Moreover, the TWPC-MRTA problem introduces extra constraints beyond traditional MRTA, which makes it much more challenging to solve.

To solve this challenging problem, we first investigate the characteristics of the problem. In our study on MRTA, a phenomenon of *generally sparse but locally dense* is observed in the temporal domain. This phenomenon forms the basis for clustering the entire task set into a few clusters. As a result,

TABLE II
THE NOTATIONS IN METHODOLOGY.

Equation	Description
Regular Notation	
G_d	The measure graph that describes the temporal distance among tasks.
E_d	The edge set of G_d.
W_d	The weight matrix of G_d.
$\Delta_{i,j}$	The element of W_d which refers to the temporal distance between task t_i and task t_j.
B	The task batch refers to the task set subset.
S	The sequence of the task batches.
T_I	The initial task set produced by Algorithm 1 refers to the intermediate variable of the Task Division Algorithm.
S_t	The task sequence produced by Algorithm 2 that refers to the intermediate variable of the Task Division Algorithm.
S_d	The sequence of the temporal distance between the adjacent tasks in S_t.
Decision Variable	
c	The binary assignment variables, which is 1 if the task sequence S_d is divided at the connection before i-th task, otherwise is 0.
Model Parameter	
$n_{t-upper}$	The upper limit of the number of tasks in the batch.
$n_{t-lower}$	The lower limit of the number of tasks in the batch.

different task clusters are sparsely distributed in the temporal domain, while the tasks within a cluster are densely distributed and overlapped in the temporal domain. In the former case, methods like greedy algorithms [37] and heuristics algorithms [44] are sufficient to make a schedule for the tasks of different task clusters due to the relatively large distance between them. For the latter, more complicated algorithms with higher accuracy and acceptable efficiency shall be designed.

Through the above analysis, we propose a graph-based batch MRTA (BMRTA) algorithm for solving the TWPC-MRTA problems. The basic idea is to treat different tasks with different strategies according to their temporal-spatial distributions. As a result, we automatically divide the task set into clusters (batches) of appropriate sizes (*task division*), each of which can be efficiently solved by an accurate solver like MIP (*batch solving*). The proposed approach consists of three key steps. As shown in Fig. 1, we first construct a *measure graph* for estimating the distance between tasks. We then perform *task division* based on the measure graph. Finally, we design a *batch solver* to solve the methods inside batches. Moreover, it is noted that the summary of the main notations in this section is provided in Table II for convenience.

A. Graph Construction

Two graphs are constructed. Besides the precedence graph defined in Section III-B, we introduce a measure graph that describes the distance between tasks. A measure graph is a weighted directed acyclic graph $G_d = (\mathcal{T}, E_d, W_d)$, where $\mathcal{T}$ is the vertex set, E_d is the edge set, and the weight set W_d is the weight matrix. The element of W_d, namely $W_d(t_i, t_j) = \Delta_{i,j}$, represents the difference between tasks i and j.

Next, we define the weight matrix. A natural way is to formulate the difference between tasks i and j as the temporal difference between the start time of the two tasks, namely $S(t_i) - S(t_j)$. However, the exact execution time can only be obtained after the task assignment and scheduling are made. To escape this dilemma, we introduce a data-driven method that enables estimating the $S(t_i) - S(t_j)$ from experience (historical data).

More specifically, the difference between tasks i and j are defined as a linear combination of the difference of *earliest start time*, *time window length*, and *task duration* between tasks i and j , as follows.

$$\Delta_{i,j} = \begin{bmatrix} k_1 & k_2 & k_3 \end{bmatrix} \begin{bmatrix} \mathrm{est}(t_i) - \mathrm{est}(t_j) \\ \mathrm{twl}(t_i) - \mathrm{twl}(t_j) \\ \mathrm{dur}(t_i) - \mathrm{dur}(t_j) \end{bmatrix}, \tag{24}$$

where $\boldsymbol{K} = [k_1 \quad k_2 \quad k_3]^T$ is linear model coefficient.

We provide a data-driven way for determining $\boldsymbol{K}$. In the case of n_t tasks, n_r robots and no precedence constraints, n_s training samples with different distribution are generated. And then the MIP method is used to solve these n_s training samples. From the solution of training samples, we can get the start time of the task $S(t)$ and use it as the measurement of the temporal distance between tasks, so that the problem of modeling the temporal distance between t_i and t_j is transformed into the fitting problem between $\Delta_{i,j}$ and $S(t_i) - S(t_j)$. A simple solution to this problem (used in this article) is the least-squares method, namely

$$\min_{K} \parallel \boldsymbol{X}\boldsymbol{K} - \boldsymbol{Y} \parallel^2, \tag{25}$$

where

$$\boldsymbol{X}\boldsymbol{K} = \begin{bmatrix} \Delta_{1,2} & \Delta_{1,3} & \cdots & \Delta_{n_{s-1},n_s} \end{bmatrix}^T,$$
$$\boldsymbol{Y} = (S(t_i) - S(t_j))_{\frac{n_s(n_s-1)}{2} \times 1}, 1 \le i < j \le n_s.$$

By solving (25), we have

$$\boldsymbol{K} = (\boldsymbol{X}^T\boldsymbol{X})^{-1}\boldsymbol{X}^T\boldsymbol{Y}. \tag{26}$$

And then, we can get the average of $\boldsymbol{K}$ on n_s training samples

$$\boldsymbol{K}^* = \frac{1}{n_s}\sum_{i=1} n_s \boldsymbol{K}_i. \tag{27}$$

Finally, the weight matrix W_d of the measure graph for the current task is obtained by substituting the learned $\boldsymbol{K}^*$ into (24). It is clear that the weight matrix W_d is the antisymmetric matrix. A reasonable explanation for this is the $\Delta_{i,j}$ in W_d indicates that the task t_i is ahead of task t_j for $\Delta_{i,j}$ time units in the schedule of all robots. If $t_i \prec t_j, \Delta_{i,j} < 0$; $t_j \prec t_i$, $\Delta_{i,j} > 0$, where $t_i \prec t_j$ is denoted as the task t_i is scheduled before t_j. When $\Delta_{i,j} < 0$, the larger the $|\Delta_{i,j}|$, the earlier the t_i is scheduled than t_j; when $\Delta_{i,j} > 0$, the larger the $|\Delta_{i,j}|$, the later the t_i is scheduled than t_j.

B. Task Division

The batch problem of the task set can be modeled as a partition problem, and the goal of the partition problem is to find a feasible solution to the problem. The optimal partition problem for the BMRTA can be described as finding a sequence of dividing the task set into a series of task subsets:

$$S = (B_0, B_1, \cdots, B_{N_B}). \tag{28}$$

This sequence is required to satisfy the constrain

$$\forall B_i, B_j \in S(i < j), \forall t_i \in B_i, t_j \in B_j, t_i \prec t_j, \tag{29}$$

where S is the ordered sequence of divided task subsets; $t_i \prec t_j$ indicates that in the optimal allocation, t_i is scheduled before t_j.

Algorithm 1 Initial tasks allocation

Input: input parameters $\mathcal{T}, E, \text{est}(t), d(t_1, t_2) \forall t, t_1, t_2 \in \mathcal{T}$
Output: T_I ▷ initial task set
$T_u \leftarrow \mathcal{T}$, ▷ unallocated task set
$E_u \leftarrow E$, ▷ precedence constraints among unallocated tasks
$AT_{r,t} \leftarrow \min[\text{est}(t), d(r,t)], r \in \mathcal{R}, t \in T_u$. ▷ compute heuristic distance
$T_I \leftarrow \varnothing$.
while $|T_I| < n_r$ **do**
  $r^*, t^* \leftarrow \arg\min_{r,t} AT_{r,t}$,
  $AT \leftarrow AT \backslash \{AT_{r^*,t^*}\}$.
  if $r^* \in T_u$ **and** GETPARENTS(E_u, t^*) **then**
    $T_u \leftarrow T_u \backslash \{t\}$,
    $AT \leftarrow AT \backslash (AT_{r^*,*} \cap AT_{*,t^*})$,
    $T_I \leftarrow T_I \cup \{t^*\}$,
    RELEASECHILDREN(E_u, t^*).
return T_I
function RELEASECHILDREN(E, t) ▷ delete the edges connecting t and its children nodes.
function GETPARENTS(E, t) ▷ return the set of nodes with t as their children.

Algorithm 2 Task sequence allocation

Input: input parameters $\mathcal{T}, E, T_I, W_d$
Output: S_t, S_d
$T_a \leftarrow T_I$, ▷ allocated task set
$T_u \leftarrow \mathcal{T} \backslash T_a$, ▷ unallocated task set
$E_u \leftarrow E$, ▷ precedence constraints among unallocated tasks
RELEASECHILDREN(E_u, t), $\forall t \in T_u$
while $|T_u| > 0$ **do**
  $AT_t \leftarrow \min_{t_i} W_d(t_i, t), t_i \in T_a, t \in T_u$. ▷ compute heuristic distance
  while TRUE **do**
    $t^* \leftarrow \arg\min_t AT_t$,
    $AT_t \leftarrow AT_t \backslash \{t^*\}$.
    if $\text{parents}(t^*) = \varnothing$ **then**
      $T_u \leftarrow T_u \backslash \{t^*\}$,
      $T_a \leftarrow \{t^*\}$,
      $S_d \leftarrow S_d \cup \{-AT_{t^*}\}$,
      RELEASECHILDREN(E_u, t^*).
      break.
$S_t \leftarrow T_a$.
return S_t, S_d
function RELEASECHILDREN(E, t) ▷ delete the edges connecting t and its children nodes.

Algorithm 3 Task sequence division

Input: input parameters $S_d, S_t, n_{t-upper}, n_{t-lower}$
Output: $B_i, B_i \in S_t$
$K \leftarrow \lfloor n_t / n_t upper \rfloor$, ▷ ideal batch size
$c^{'} \leftarrow \varnothing$. ▷ divided position set
while $K > 0$ **do**
  $\text{Idx}^* \leftarrow \arg\max_{\text{Idx}} S_d$, ▷ compute the divided position
  $S_d \leftarrow S_d \backslash \{S_{d_{\text{Idx}^*}}\}$.
  if INSERTCHECK$(c^{'} \cup \{\text{Idx}^*\}, \infty, n_{t-lower})$ **then**
    $c^{'} \leftarrow c^{'} \cup \{\text{Idx}^*\}$,
    $K \leftarrow K - 1$.
while INSERTCHECK$(c^{'} \cup \{\text{Idx}^*\}, n_{t-upper}, 0)$ **do**
  $\text{Idx}^* \leftarrow \arg\max_{\text{Idx}} S_d$,
  $c^{'} \leftarrow c^{'} \cup \{\text{Idx}^*\}$.
$S \xleftarrow{c^{'}} S_t$. ▷ divide task sequence by $c^{'}$
return S
function INSERTCHECK$(c, n_{t-upper}, n_{t-lower})$ ▷ return TRUE if the number of tasks in all subsets divided by c is not more than $n_{t-upper}$ and not less than $n_{t-lower}$.

It can be seen from the constraint of this model that the biggest difficulty in solving the optimal partition problem is that the relationship between t_i, t_j can not be obtained before the problem is solved. To solve this problem, we propose a graph decomposition algorithm. This graph decomposition algorithm makes use of the proposed measure graph and precedence graph, decomposing the measure graph into several sub-graphs. At the same time, the task set will be appropriately divided into several subsets.

The decomposition algorithm has two parts: *task sequence construction* and *task sequence division*. In this algorithm, the measure graph of the task set is set as the criterion of the distance between tasks. In the first part, Algorithm 1 together with Algorithm 2 are introduced to find a road in the measure graph. This road implies a possible schedule solution for all robots on the task set and it is unreliable when the distance between adjacent tasks is too small. In the second part, Algorithm 3 is designed to dive the sequence into several sub-sequence in places where there is a large distance between adjacent tasks. Finally, the sub-graphs will be obtained by restoring the sub-sequences to graphs.

1) Task Sequence Construction: Taking the measure graph as the criterion for measuring the relationship between tasks, the set of tasks can be transformed into the sequence of tasks S_t and the corresponding sequence of the distance between tasks S_d. The conversion scheme is: firstly determine n_r tasks to be allocated to n_r robots, as the initial task set, the allocation algorithm is shown in Algorithm 1.

After the initial set is obtained, the task sequence can be acquired. The allocation algorithm to get the task sequence is shown in Algorithm 2.

According to Algorithm 2, the task sequence S_t and the distance sequence S_d between the corresponding tasks can be obtained, where the task sequence S_t includes all tasks to be allocated and the tasks in S_t approximately meet

$$\forall t_i, t_j \in S_t (i < j), t_i \prec t_j. \tag{30}$$

2) Task Sequence Division: After obtaining the task sequence, the model shown in (28) - (29) can be solved. However, the relationship between t_i and t_j is approximate in the task sequence. The value in the corresponding sequence of distance S_d implies the reasonable degree of this approximation. The larger the value, the more reasonable the approximation; the smaller the value, the more possible that this approximation can be wrong. Therefore, to enable the task sequence to best guide the task set division as much as possible, it is necessary to divide as much as possible at the places where the value in S_d is larger. Therefore, the following division model is established:

$$\min c \cdot S_d, \tag{31}$$

s.t.

$$\sum_{k=j}^{j+n_{t-lower}} c_i \leq 1, 1 \leq i \leq n_t - n_{t-lower}, \tag{32}$$

$$\sum_{k=j}^{n_t} c_i \leq 1, n_t - n_{t-lower} + 1 \leq j \leq n_t, \tag{33}$$

$$\sum_{k=j}^{j+n_{t-upper}} c_i \geq 1, 1 \leq i \leq n_t - n_{t-upper}, \tag{34}$$

$$\sum_{k=j}^{n_t} c_i \geq 1, n_t - n_{t-upper} + 1 \leq i \leq n_t, \tag{35}$$

$$c_i = 0 \quad or \quad 1, 1 \leq i \leq n_t, \tag{36}$$

where c is the decision variable and $c_i = 1$ means that the task sequence is divided at the connection before i-th task; $n_{t-upper}$ and $n_{t-lower}$ are the upper and lower limit of the number of tasks in the batch, respectively.

By solving the model showed in (31) - (36), we can get the decision vector c, which divides task sequence S at its values of 1. For example, given $c = (0, 1, 0, 0, 1, 0)$, the task sequence $S_t = (2, 3, 5, 1, 4)$ (the number in S_t indicates the ID of the task.) will be divided into $S = (\{2\}, \{3, 5, 1\}, \{4\})$ as the form of (28).

Unfortunately, the partition problem modeled in (31) - (36) is still NP-Complete/Hard. In order to solve the problem in polynomial time, we propose an approximate solution as shown in Algorithm 3.

C. Batch Solver

After obtaining the batched task sets, the original large-scale MIP problem will be transformed into multiple small-scale MIP problems. By solving the model shown in (2) - (23) N_B times, we can get the solutions. In particular, for the task t whose batch B_i has a precursor in S_{i-1}, the start time and time window length of the task are updated as follows:

$$\max\left[\mathrm{est}(t), \max_{t'} F(t^{'})\right] \rightarrow \mathrm{est}(t), t^{'} \in \mathrm{parents}(t) \cap S_{i-1}, \tag{37}$$

$$\mathrm{twl}(t) + \min\left[0, \mathrm{est}(t) - \max_{t'} F(t^{'})\right] \rightarrow \mathrm{twl}(t), \tag{38}$$

where $t^{'} \in \mathrm{parents}(t) \cap S_{i-1}$. Otherwise, the precedence constraints of tasks between B_i and S_{i-1} may be violated.

D. Complexity Analysis

Most of the computational costs of the BMRTA algorithm come from division in the *Task Division* step and sub-problem solving in the *Batch Solver* step. If a task set with n_t tasks is divided into N_B batches ($N_B \leqslant n_t$), the total time consumed by the entire algorithm is bounded by

$$T_{BMRTA} \leqslant T_{divide} + N_B \cdot max(T_{batch}). \tag{39}$$

The dividing algorithm mainly includes Algorithms 1-3. The complexity of Algorithms 1-3 are $O(n_r^2)$, $O(n_t^2)$, $O(n_t^2)$ respectively. On the other hand, the complexity of the sub-problem-solving algorithm is $O(n_t C)$, where C is the upper bounding of the complexity of the sub-problem. Since the scale of the sub-problem is limited, C is a finite constant (increases with the size of the problem). As a result, the overall complexity of the algorithm is $O(n_r^2 + n_t^2 + n_t C)$.

V. EXPERIMENTS AND RESULTS

A. Experimental Settings

To our best knowledge, there is no widely recognized application scenario to evaluate the algorithms. Meanwhile, there are no publicly available realistic data sets for the TWPC-MRTA problem or the vehicle routing problem with similar constraints.

In this paper, we design a delivery task in a smart warehouse application to test the proposed method in which mobile robots must first transport from the start location to the task location in order to receive or deliver goods without violating the time window and precedence constraints. A 100×100 meter (m) map with obstacles inside (the dark grey brick shape) is constructed to simulate the real warehouse. We assume that the robots are homogeneous, and the initial location of robots and all the task locations are randomly picked in the map. The path between the task and the task/robot is computed by the classical A Star algorithm [45]. The moving speed of the robot is assumed to be 1 meter per second (m/s) and the start-stop time is ignored. The overall setup of this scenario can be found in Fig.2.

For the temporal and precedence constraints, we generate two data sets to evaluate our algorithm, namely the uniform distribution (UD) data set and cluster distribution (CD) data set. The first data set is generated according to the rule introduced by [15]. For temporal constraints, every task will be assigned an *earliest start time*, a *length of the time window*, and a *task duration*, which are randomly drawn from $U(25, 400), U(100, 1200), U(20, 40)$ respectively. As randomization is important in testing MRTA algorithm [43], the precedence constraints between tasks are also generated randomly. The algorithm to generate precedence graphs is described in [42]. The second data set is the modified version of the first data, in which we pan the distribution of the earliest start time of the time window. Specifically, we change the distribution of the earliest start time from $U(25, 400)$ to $U(25 + k\Delta_t, 400 + k\Delta_t)$ to simulate the generally sparse but locally dense characteristics. In our experiment, $\Delta_t = 40, k = \mathrm{Index}_t/3$. These two data sets will be released at the link https://github.com/zhanglixuan0720/TWPC-MRTA.

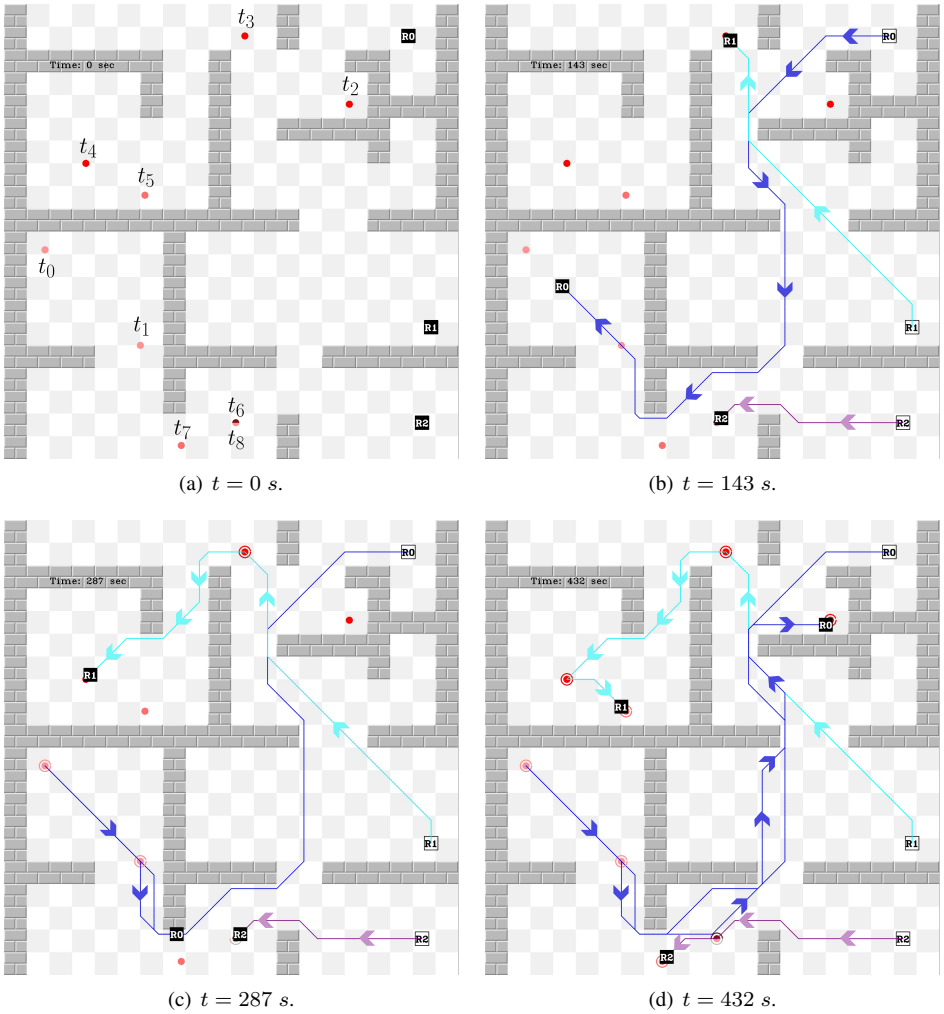
For these two data sets, we set the number of robots $n_r = 3, 5, 10$ in three cases, in the four instances of the number of tasks $n_t = 9, 18, 27, 36$, each combination of (n_r, n_t) generates 100 test samples, and a total of 1200 test samples are generated.

In our setting, all robots are homogeneous, with the same power value P and a constant movement speed of 1 m/s. Here, we use the measured power percentage (motion power: 94.7%, sensor power: 1.26%, control power: 4.04%) from [40], when the mobile robot moves with smooth constant velocity 1m/s. In the smart warehouse scenario, we reasonably suppose that the robots run at full power during regular working state and disable the motion system in standby (idle) mode (only the sensor system and control system working). Therefore, the working energy consumption is defined as (40), in which DIS is the total distance traveled by robots, and DUR is the total task duration finished by robots. The standby energy consumption is defined as (41), in which the IDT is the total idle time cost by robots. Finally, the total energy consumption EC of multi-robot in (1) can be rewritten as (42), in which we omit the constant '1' and the total power P without losing the correctness for simplicity[1].

$$WEC = (DIS/1 + DUR)P \tag{40}$$

$$SEC = 0.053 \cdot IDT \cdot P \tag{41}$$

[1]The total power P of the energy cost reported in Tab.III and Tab.IV is also omitted for simplicity.



(a) $t = 0$ s. (b) $t = 143$ s.

(c) $t = 287$ s. (d) $t = 432$ s.

Fig. 2. The snapshots of the experiment under setting $n_r = 3, n_t = 9$. The task allocation and scheduling strategy are solved by the BMRTA algorithm. (a) The initial layout of the map, robots (R0, R1, R2), and tasks (from t_0 to t_8). The dark grey brick shape with white stripes represents the obstacles. The black square with 'R' represents the current robot's location and ID. The red round shape represents the task, and different saturation indicates different priorities. The higher the saturation, the higher the priority. The round shape located in the lower middle of the map with two different colors indicates two tasks (t_6:up half part, t_8:bottom half part) with the same location distinguished by different brightness. (b) The snapshot of the working process when the timestamp is 143 seconds. The hollow black square represents the initial location of the robot. Lines and corresponding arrows indicate the paths and movement directions of different robots with different colors, respectively. At this time, R0 was going for t_0. R1 and R2 arrived at the location of t_3 and t_6, respectively. (c) The snapshot of the working process when the timestamp is 287 seconds. The successfully finished task is marked by the red hollow circle (e.g., t_3), while the failed task is marked by the grey hollow circle (e.g., t_8). At this moment, R0 finished t_0 and t_1, going for t_2. R1 completed t_3 and reached the location of t_4. While R2 still stayed at the location of t_6. (d) The end of the working processing. The makespan of this experiment is 432 seconds. The task completed rate is 8/9, which means 8 tasks (R0: t_0, t_1, t_2, R1: t_3, t_4, t_5, R2: t_6, t_7) are successfully finished while 1 task (t_8) failed.

$$\begin{aligned} EC &= \sum_{r_i \in \mathcal{R}} (WEC^{r_i} + SEC^{r_i}) \\ &= \sum_{r_i \in \mathcal{R}} (DIS^{r_i} + DUR^{r_i} + 0.053 \cdot IDT^{r_i}) \end{aligned} \tag{42}$$

Additionally, the bi-objective model described in (2)-(23) is solved by the hierarchical approach , which requires a predefined priority for each objective and solving them in decreasing priority order (from higher to lower). In our implementation, which exploits Gurobi 9.5.2 [46], we set the number of allocated tasks (i.e., $\sum A_{t_j}^{r_i}$) objective as the highest priority and the energy cost (i.e., EC) as the second priority. Therefore, the solver will first optimize the number of allocated tasks and optimize the energy cost without affecting the number of assigned tasks.

B. Experimental Results and Analysis

Three algorithms (MIP, TePSSI, and BMRTA) are tested on two data sets (UD and CD). We evaluated these algorithms from five metrics: the *task completed rate*, *energy cost*, *makespan*, *distance*, and *idle time*. It is worth mentioning that only the *task completed rate* and *energy cost* are considered in the objective function (equation (2)), and the other three metrics are also reported to further characterize the performances of the proposed method. The *task completed rate* is the ratio of finished tasks over the total number of tasks. When a task can not be allocated or finished due to the limited number of robots or time window constraints, the task will be failed. The *energy cost* is the energy cost by all robots, which is calculated by (42) using *distance* and *idle time*. The *makespan* of one test

TABLE III
THE COMPARISON OF TASK COMPLETE RATE, ENERGY COST, MAKESPAN, DISTANCE, AND IDLE TIME

n_r			3		5		10	
n_t			9	36	9	36	9	36
Task Completed Rate ↑	UD	MIP	**86.11%**	-	**87.89%**	-	**93.11%**	-
		TePSSI	84.22%	83.67%	87.89%	92.97%	93.11%	98.25%
		BMRTA	**86.11%**	**89.14%**	**87.89%**	**95.78%**	**93.11%**	**98.69%**
	CD	MIP	**83.22%**	-	**89.22%**	-	**96.56%**	-
		TePSSI	83.11%	81.64%	89.00%	91.11%	96.56%	98.06%
		BMRTA	**83.22%**	**89.42%**	**89.22%**	**93.53%**	**96.56%**	**98.28%**
Energy Cost (Joule) ↓	UD	MIP	**511.18**	-	**432.13**	-	**356.89**	-
		TePSSI	753.96	2131.53	727.32	2400.73	767.70	2776.99
		BMRTA	**511.18**	**1871.58**	**432.13**	**1673.65**	**356.89**	**1486.10**
	CD	MIP	**465.69**	-	**416.45**	-	**361.17**	-
		TePSSI	685.11	2213.00	741.99	2552.07	785.61	3136.58
		BMRTA	**465.69**	**1903.14**	**416.45**	**1639.64**	**361.17**	**1467.69**
Makespan (Seconds) ↓	UD	MIP	503.27	-	475.98	-	445.55	-
		TePSSI	**494.23**	**892.31**	**466.33**	674.61	**439.49**	**467.32**
		BMRTA	503.27	894.42	475.98	**650.14**	445.3	519.18
	CD	MIP	554.8	-	487.91	-	502.68	-
		TePSSI	**540.96**	1152.96	**477.32**	**908.09**	**497.29**	**812.02**
		BMRTA	554.8	**1129.42**	487.91	912.96	502.31	831.48
Distance (Meters) ↓	UD	MIP	**280.94**	-	**195.9**	-	**101.05**	-
		TePSSI	528.08	1234.96	491.09	1422.56	511.86	1720.86
		BMRTA	**280.94**	**918.74**	**195.9**	**670.5**	**101.05**	**425.16**
	CD	MIP	**240.72**	-	**184**	-	**113.43**	-
		TePSSI	460.49	1339.61	509.94	1593.33	537.87	2092.56
		BMRTA	**240.72**	**953.89**	**184**	**658.88**	**113.43**	**422.29**
Idle Time (Seconds) ↓	UD	MIP	998.68	-	1947.82	-	4098.66	-
		TePSSI	**728.77**	**545.44**	**1604.38**	**972.37**	**3627.25**	**1896.26**
		BMRTA	998.68	811.73	1947.82	1577.1	4096.16	3705.75
	CD	MIP	1198.75	-	2023.15	-	4665.68	-
		TePSSI	**937.81**	**1245.92**	**1644.66**	**1988.43**	**4187.34**	**4983.67**
		BMRTA	1198.75	1485.17	2023.15	2925.21	4661.98	6847.16

TABLE IV
THE COMPARISON OF TASK COMPLETE RATE, ENERGY COST, MAKESPAN, DISTANCE, AND IDLE TIME (CONTINUATION)

n_r			3		5		10	
n_t			18	27	18	27	18	27
Task Completed Rate ↑	UD	MIP	-	-	**94.67%**	-	96.56%	-
		TePSSI	86.33%	88.70%	93.28%	93.48%	96.56%	97.96%
		BMRTA	**90.89%**	**92.07%**	94.61%	**96.04%**	**96.56%**	**98.11%**
	CD	MIP	-	-	**97.28%**	-	95.11%	-
		TePSSI	88.50%	86.11%	95.17%	93.00%	95.11%	97.33%
		BMRTA	**89.94%**	**89.07%**	97.11%	**94.96%**	**95.11%**	**97.37%**
Energy Cost (Joule) ↓	UD	MIP	-	-	**851.99**	-	729.63	-
		TePSSI	1294.74	1810.28	1413.52	1889.67	1582.92	2211.76
		BMRTA	**1008.75**	**1467.58**	887.97	**1283.13**	**729.63**	**1071.28**
	CD	MIP	-	-	**865.11**	-	735.06	-
		TePSSI	1361.48	1794.45	1493.13	2074.04	1612.33	2459.76
		BMRTA	**952.38**	**1415.69**	891.58	**1265.19**	**735.06**	**1070.11**
Makespan (Seconds) ↓	UD	MIP	-	-	**467.66**	-	467.39	-
		TePSSI	641.92	776	485.08	585.81	**453.67**	**464.42**
		BMRTA	**601.68**	**758.64**	490.61	**555.24**	467.15	474.46
	CD	MIP	-	-	**611.6**	-	591.7	-
		TePSSI	763.96	955.47	623.13	**787.79**	**582.48**	**712.33**
		BMRTA	**740.74**	**939.08**	645.9	791.84	591.7	717.46
Distance (Meters) ↓	UD	MIP	-	-	**337.98**	-	207.81	-
		TePSSI	821.36	1080.94	906.61	1146.54	1061.1	1426.96
		BMRTA	**512.27**	**712.84**	374.24	**525.34**	**207.81**	**285.52**
	CD	MIP	-	-	**350.03**	-	214.37	-
		TePSSI	896.64	1103.1	987.84	1324.16	1091.64	1668.63
		BMRTA	**482.83**	**697.07**	377.29	**503.83**	**214.37**	**278.68**
Idle Time (Seconds) ↓	UD	MIP	-	-	1486.36	-	3944.32	-
		TePSSI	**631.07**	**517.77**	**1011.93**	**1039.43**	**2953.83**	**2432.49**
		BMRTA	796.34	808.39	1565.13	1493.12	3941.92	3673.37
	CD	MIP	-	-	2192.94	-	5181.99	-
		TePSSI	**930.45**	**1072.01**	**1622.57**	**1864.96**	**4212.52**	**4663.59**
		BMRTA	1269.89	1401.6	2337.97	2694.06	5181.99	6104.54

TABLE V
THE CHANGE OF TASK COMPLETED RATE AND ENERGY COST WITH $n_{t-upper}$

Data	n_t \ $n_{t-upper}$	9	18	27	36
UD	7	87.11% / 432.42	93.33% / 907.05	93.78% / 1305.05	93.11% / 1677.23
	12	87.89% / 432.13	94.67% / 885.23	93.63% / 1294.27	94.31% / 1699.73
	17	87.89% / 432.13	94.61% / 887.97	96.04% / 1283.13	95.78% / 1673.65
CD	7	88.89% / 422.75	96.56% / 907.11	94.30% / 1323.58	92.97% / 1713.96
	12	89.22% / 416.45	97.11% / 888.26	94.59% / 1306.78	93.44% / 1695.85
	17	89.22% / 416.45	97.11% / 891.58	94.96% / 1265.19	93.53% / 1639.64

instance is defined as the latest finish time of any task in the corresponding task set. The *distance* is the physical distance traveled by all robots when finishing given tasks. The time that robots run but do nothing is defined as *idle time*. In this experiment, all programs are implemented in Python 3.7 and tested on an Intel(R) Core(TM) i7-8750H CPU. The snapshots of the example experiment under setting $n_r = 3, n_t = 9$ are shown in Fig. 2.

The experiment results of the UD and CD data sets on these five metrics are shown in Table III-IV[2], where the '↑' indicates the higher the corresponding metric, the better, while the '↓' indicates the lower the corresponding metric, the better. It is worth mentioning that the other metrics will be better when the *task completed rate* decreases. Therefore, these metrics should be considered jointly, and the latter predominates.

It can be seen from the results that our algorithm outperforms previous methods on the *task completed rate* and *energy cost* as well as gets comparable results with TePSSI on *makespan*. Specifically, the results have significant improvements on *energy cost* for both the UD and CD data sets and have a slightly more improvement on the CD data set. These indicate that our algorithm can reduce energy consumption and performs better on data sets with cluster structure in the temporal domain. It is also worth mentioning that no result for MIP is implied by '-' when $n_t \geq 18$ because the MIP can not achieve an optimal solution within acceptable computational efficiency for large-scale problems. With the incomplete solution acquired by MIP, the final result for our problem may be invalid. Therefore, we list MIP here as our baseline method and pay more attention to comparing TePSSI and BMRTA.

As shown in Table III, the overall results of the BMRTA on *task completed rate* are better than that of the TePSSI. The BMRTA (UD:89.14%, CD:89.42%) maximally outperforms TePSSI (UD:83.67%, CD:81.64%) by 6.5%[3] ($n_r = 3, n_t = 36$) on UD data sets and by 9.5% ($n_r = 3, n_t = 36$) on CD data sets. For both the UD and CD data sets, the BMRTA outperforms TePSSI by 2.0% on average. In addition, we discover that the BMRTA achieves better results when the average number of tasks processed by robots is large. For example, on the UD data set, the BMRTA (89.14%) outperforms TePSSI (83.67%) by 6.5% when $n_r = 3, n_t = 36$. However, the BMRTA (93.11%) gets almost the same result as TePSSI (93.11%) when $n_r = 10, n_t = 9$. In this regard, we reasonably claim that, for *task completed rate*, the difficulty of allocating tasks will decrease if given enough robots and is positively correlated with the average number of tasks processed by robots. This indicates the proposed algorithm can perform better on more difficult tasks and helps increase efficiency, thus demonstrating the excellence of our algorithm.

Compared with the TePSSI, the results of BMRTA on *energy cost* significantly improve. For the UD data set, the BMRTA (729.63) maximally reduces the *energy cost* by 53.9% (TePSSI:1582.92, $n_r = 10, n_t = 18$) and by 35.9% on average. For the CD data set, the BMRTA (1070.11) maximally reduces the *energy cost* by 56.5% (TePSSI:2459.76, $n_r = 10, n_t = 27$) and by 39.5% on average. For both data sets, the BMRTA reduces the *energy cost* by 37.7% on average. All of these indicate the BMRTA is much better than TePSSI on *energy cost* and has considerable advantages in reducing energy consumption. Although the idle time of the TePSSI is lower than the BMRTA, the distance of the former one is much longer than the latter, resulting in massive energy consumption of the TePSSI. This interesting phenomenon is due to the robot's motion system's much higher power percentage than the sensor and control system.

For the *makespan*, we discover that the BMRTA gets comparable results with TePSSI. As shown in Table III, for some test samples (e.g., $n_r = 5, n_t = 9$, UD data set), the TePSSI (466.33) outperforms BRMTA (475.98), while for others (e.g., $n_r = 5, n_t = 36$, UD data set, TePSSI:674.61, BMRTA:650.14), it is the opposite. According to the results statistics on *makespan*, the *makespan* of BMRTA is 0.5% higher than TePSSI for all test samples. This does not prove that TePSSI or BMRTA has better performance than each other on *makespan*. Therefore, combining the previous discussion, our algorithm generally outperforms TePSSI on *task completed rate* and *energy cost*, helps save energy consumption, and increases efficiency while keeping comparable time consumption with TePSSI. All these prove the effectiveness of the BMRTA algorithm and its advantages over the MIP and TePSSI algorithms.

In addition, we make $n_{t-upper}$ (the maximum number of tasks of the batch) as an independent variable to study the influence of $n_{t-upper}$ on the effect of our BMRTA algorithm. We conduct this experiment in case $n_r = 5$, which has more average number of tasks processed by robots with higher persuasion in the MRTA problem. Additionally, we set $n_{t-lower}$ as $\frac{n_{t-upper}}{2}$ for experimental conciseness without loss of generality. The results for the change of the *task completed*

[2]Due to the limited page size, we split the result into two tables. In these tables, the representative cases of $n_t = 9, 36$ are shown in Table III, and the rest cases of $n_t = 18, 27$ are shown in Table IV.

[3]Unless otherwise specified, all percentages are relative percentages.

rate and *energy cost* with $n_{t-upper}$ are shown in Table V. As shown in this table, the overall performance of BMRTA is positively related to hyperparameter $n_{t-upper}$. The higher the $n_{t-upper}$, the better the performance. For example, on the UD data set, the *task completed rate* of $n_{t-upper} = 17$ (95.78%) outperforms that of $n_{t-upper} = 12$ (94.31%) by 1.5% and that of $n_{t-upper} = 7$ (93.11%) by 2.9% ($n_t = 36$). For UD dataset, taking $n_{t-upper} = 7$ (907.05) as the benchmark, the *energy cost* of $n_{t-upper} = 12$ (885.23) is reduced by 2.4% and that of $n_{t-upper} = 17$ (887.92) is reduced by 2.1% ($n_t = 18$). With the increase of $n_{t-upper}$, the performance of BMRTA will significantly increase. Still, it will increase the algorithm's complexity and potential runtime. Therefore, it is more practical to a trade-off between performance and runtime in real warehouses and industrial environments.

VI. CONCLUSIONS

We study the challenging Multi-Robot Task Allocation problem with the time window and precedence constraints (TWPC-MRTA). For the TWPC-MRTA, we propose a novel formulation of a multi-robot energy cost model and an algorithm named Batch MRTA (BMRTA) to solve this problem efficiently. In BMRTA, we construct a measure graph to provide cues to divide the task sets. We then provide a graph decomposition-based method to divide the tasks into batches and design a batch solver based on our multi-robot energy cost formulation to obtain the final solution to TWPC-MRTA. We design several experiments to demonstrate the effectiveness of BMRTA. Extensive experiments show that BMRTA outperforms state-of-the-art algorithms such as MIP and TePSSI and performs excellently in reducing energy consumption. However, we acknowledge that there is a strict relation between the effectiveness of the method in clustering and assigning efficient tasks and the sparsity of the problem, which is highly dependent on the environment. Future work could investigate more accurately the relationship which exists between the two components of the problem.

ACKNOWLEDGMENT

This work is funded by the National Key Research and Development Project of China (2019YFB1312000), the National Natural Science Foundation of China (No. 62076195). We acknowledge the support of the MUR PNRR project FAIR - Future AI Research (PE00000013) funded by the NextGenerationEU.

REFERENCES

[1] H. Zhu, "Role-based collaboration and e-cargo: Revisiting the developments of the last decade role-based collaboration (rbc) is an emerging computational methodology that uses roles as the prim," *IEEE Systems, Man, and Cybernetics Magazine*, vol. 1, no. 3, pp. 27–36, 2015.

[2] ——, *E-CARGO and role-based collaboration: Modeling and solving problems in the complex world*. John Wiley & Sons, 2021.

[3] B. P. Gerkey and M. J. Mataric, "Multi-robot task allocation: analyzing the complexity and optimality of key architectures," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA 2003)*. IEEE, 2003, Conference Proceedings.

[4] S. C. Botelho and R. Alami, "M+: a scheme for multi-robot cooperation through negotiated task allocation and achievement," in *Proceedings of the 1999 IEEE International Conference on Robotics & Automation*. IEEE, 1999, Conference Proceedings.

[5] B. P. Gerkey and M. J. Matarić, "A formal analysis and taxonomy of task allocation in multi-robot systems," *The International Journal of Robotics Research*, vol. 23, no. 9, pp. 939–954, 2004.

[6] R. Sengupta and R. Nagi, "Decentralized makespan minimization for uniformly related agents," in *2021 IEEE 17th International Conference on Automation Science and Engineering (CASE)*, 2021, pp. 2140–2145.

[7] Q. Guan, X. Hong, W. Ke, L. Zhang, G. Sun, and Y. Gong, "Kohonen self-organizing map based route planning: A revisit," in *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, pp. 7969–7976.

[8] E. Nunes, M. Manner, H. Mitiche, and M. Gini, "A taxonomy for task allocation problems with temporal and ordering constraints," *Robotics and Autonomous Systems*, vol. 90, pp. 55–70, 2017.

[9] D. Liu, B. Huang, and H. Zhu, "Solving the tree-structured task allocation problem via group multirole assignment," *IEEE Transactions on Automation Science and Engineering*, vol. 17, no. 1, pp. 41–55, 2020.

[10] J. Clausen, "Branch and bound algorithms-principles and examples," *Department of Computer Science, University of Copenhagen*, pp. 1–30, 1999.

[11] D.-H. Lee, S. A. Zaheer, and J.-H. Kim, "A resource-oriented, decentralized auction algorithm for multirobot task allocation," *IEEE Transactions on Automation Science and Engineering*, vol. 12, no. 4, pp. 1469–1481, 2015.

[12] S. Ponda, J. Redding, C. Han-Lim, J. P. How, M. Vavrina, and J. Vian, "Decentralized planning for complex missions with dynamic communication constraints," in *2010 American Control Conference*. IEEE, 2010, Conference Proceedings.

[13] E. Nunes and M. L. Gini, "Multi-robot auctions for allocation of tasks with temporal constraints," in *AAAI*, 2015, Conference Proceedings.

[14] L. Luo, N. Chakraborty, and K. Sycara, "Distributed algorithms for multirobot task assignment with task deadline constraints," *IEEE Transactions on Automation Science and Engineering*, vol. 12, no. 3, pp. 876–888, 2015.

[15] E. Nunes, M. McIntire, and M. Gini, "Decentralized allocation of tasks with temporal and precedence constraints to a team of robots," in *Proceedings of the 2016 IEEE International Conference on Simulation, Modeling, and Programming for Autonomous Robots*. IEEE, 2016, Conference Proceedings.

[16] X. Bai, A. Fielbaum, M. Kronmüller, L. Knoedler, and J. Alonso-Mora, "Group-based distributed auction algorithms for multi-robot task assignment," *IEEE Transactions on Automation Science and Engineering*, pp. 1–12, 2022.

[17] M. Badreldin, A. Hussein, and A. Khamis, "A comparative study between optimization and market-based approaches to multi-robot task allocation," *Advances in Artificial Intelligence (16877470)*, 2013.

[18] F. Janati, F. Abdollahi, S. S. Ghidary, M. Jannatifar, J. Baltes, and S. Sadeghnejad, *Multi-robot Task Allocation Using Clustering Method*. Springer International Publishing, 2017, pp. 233–247.

[19] G. A. Korsah, A. Stentz, and M. B. Dias, "A comprehensive taxonomy for multi-robot task allocation," *The International Journal of Robotics Research*, vol. 32, no. 12, pp. 1495–1512, 2013.

[20] M. G. Lagoudakis, E. Markakis, D. Kempe, P. Keskinocak, A. Kleywegt, S. Koenig, C. Tovey, A. Meyerson, and S. Jain, "Auction-based multi-robot routing," in *Proceedings of Robotics: Science and Systems*. Robotics: Science and Systems Foundation, 2005, Conference Proceedings.

[21] J. Bruno, E. G. Coffman Jr, and R. Sethi, "Scheduling independent tasks to reduce mean finishing time," *Communications of the ACM*, vol. 17, no. 7, pp. 382–387, 1974.

[22] M. R. Garey and D. S. Johnson, ""strong"np-completeness results: Motivation, examples, and implications," *Journal of the ACM (JACM)*, vol. 25, no. 3, pp. 499–508, 1978.

[23] A. Ham, "Transfer-robot task scheduling in job shop," *International Journal of Production Research*, vol. 59, no. 3, pp. 813–823, 2020.

[24] M. McIntire, E. Nunes, and M. Gini, "Iterated multi-robot auctions for precedence-constrained task scheduling," in *Proceedings of the 2016 International Conference on Autonomous Agents & Multiagent Systems*, 2016, Conference Proceedings, pp. 1078–1086.

[25] R. Hwang, M. Gen, and H. Katayama, "A comparison of multiprocessor task scheduling algorithms with communication costs," *Computers & Operations Research*, vol. 35, no. 3, pp. 976–993, 2008.

[26] M. C. Gombolay, R. J. Wilcox, and J. A. Shah, "Fast scheduling of robot teams performing tasks with temporospatial constraints," *IEEE Transactions on Robotics*, vol. 34, no. 1, pp. 220–239, 2018.

[27] G. Notomista, S. Mayya, Y. Emam, C. Kroninger, A. Bohannon, S. Hutchinson, and M. Egerstedt, "A resilient and energy-aware task

allocation framework for heterogeneous multirobot systems," *IEEE Transactions on Robotics*, vol. 38, no. 1, pp. 159–179, 2022.
[28] Ü. Bilge and G. Ulusoy, "A time window approach to simultaneous scheduling of machines and material handling system in an fms," *Operations Research*, vol. 43, no. 6, pp. 1058–1070, 1995.
[29] H. Mitiche, D. Boughaci, and M. Gini, "Efficient heuristics for a time-extended multi-robot task allocation problem," in *First International Conference on New Technologies of Information and Communication*. IEEE, 2015, Conference Proceedings.
[30] T. Li, S. Feng, J. Song, and J. Liu, "Robust bi-level programming model for robot task assignment of "rack-to-picker" picking system," *OPERATIONS RESEARCH AND MANAGEMENT SCIENCE*, vol. 28, no. 12, pp. 25–34, 2019.
[31] B. Kartal, E. Nunes, J. Godoy, and M. Gini, "Monte carlo tree search for multi-robot task allocation," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 30, 2016, Conference Proceedings.
[32] C. Tovey, M. G. Lagoudakis, S. Jain, and S. Koenig, *The Generation of Bidding Rules for Auction-Based Robot Coordination*. Springer-Verlag, 2005, pp. 3–14.
[33] E. Nunes, M. McIntire, and M. Gini, "Decentralized multi-robot allocation of tasks with temporal and precedence constraints," *Advanced Robotics*, vol. 31, no. 22, pp. 1193–1207, 2017.
[34] A. Dohn, M. S. Rasmussen, and J. Larsen, "The vehicle routing problem with time windows and temporal dependencies," *Networks*, vol. 58, no. 4, pp. 273–289, 2011.
[35] S. R. A. Haddadene, N. Labadie, and C. Prodhon, "A grasp $\times$ ils for the vehicle routing problem with time windows, synchronization and precedence constraints," *Expert Systems with Applications*, vol. 66, pp. 274–294, 2016.
[36] D. Bredström and M. Rönnqvist, "Combined vehicle routing and scheduling with temporal precedence and synchronization constraints," *European Journal of Operational Research*, vol. 191, no. 1, pp. 19–31, 2008.
[37] A. R. Mosteo and L. Montano, "A survey of multi-robot task allocation," *Instituto de Investigacin en Ingenierła de Aragn (I3A), Tech. Rep*, 2010.
[38] L. Liu and D. A. Shell, "Large-scale multi-robot task allocation via dynamic partitioning and distribution," *Autonomous Robots*, vol. 33, no. 3, pp. 291–307, 2012.
[39] C. Sarkar, H. S. Paul, and A. Pal, "A scalable multi-robot task allocation algorithm," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2018, Conference Proceedings.
[40] L. Hou, L. Zhang, and J. Kim, "Energy modeling and power measurement for mobile robots," *Energies*, vol. 12, no. 1, p. 27, 2018.
[41] A. Djenadi and B. Mendil, "Energy-aware task allocation strategy for multi robot system," *International Journal of Modelling and Simulation*, vol. 42, no. 1, pp. 153–167, 2022.
[42] G. Melançon, I. Dutour, and M. Bousquet-Mélou, "Random generation of directed acyclic graphs," *Electronic Notes in Discrete Mathematics*, vol. 10, pp. 202–207, 2001.
[43] L.-C. Canon, M. E. Sayah, and P.-C. Héam, "A comparison of random task graph generation methods for scheduling problems," in *Euro-Par 2019: Parallel Processing: 25th International Conference on Parallel and Distributed Computing, Göttingen, Germany, August 26–30, 2019, Proceedings 25*. Springer, 2019, pp. 61–73.
[44] M. M. Solomon, "Algorithms for the vehicle routing and scheduling problems with time window constraints," *Operations Research*, vol. 35, no. 2, pp. 254–265, 1987.
[45] D. Delling, P. Sanders, D. Schultes, and D. Wagner, *Engineering Route Planning Algorithms*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2009, pp. 117–139.
[46] L. Gurobi Optimization, "gurobi," 2021.

Lixuan Zhang (S'20) received his B.Eng degree in Automation from Xi'an Jiaotong University, Xi'an, China, in 2021. He is currently pursuing his Ph.D. degree with the Institute of Computing Technology (ICT), Chinese Academy of Sciences (CAS), Beijing, China. His research mainly focuses on computer vision, machine learning and robotics.

Jianzhuang Zhao (S'22) received his B.Eng degree in mechanical engineering from Zhengzhou University, Zhengzhou, China, in 2018, and the M.Eng degree in the same major from the Shaanxi Key Laboratory of Intelligent Robots, Xi'an Jiaotong University, Xi'an, China, in 2021. He is currently pursuing his Ph.D. degree with the Human-Robot Interfaces and Interaction (HRI^2) Lab at the Italian Institute of Technology (IIT), Genova, Italy, and Politecnico di Milano, Milano, Italy. He was a finalist for the best paper award on mobile manipulation at IROS 2022. His research focuses on robot learning, mobile manipulation, and human-robot collaboration.

Edoardo Lamon (M'18) received the Ph.D. degree in robotics and automation from the University of Pisa, Pisa, Italy, in 2021. He is currently an assistant professor at University of Trento and a researcher at the Human-Robot Interfaces and Interaction (HRI^2), Istituto Italiano di Tecnologia, Genoa, Italy. He is contributing to several research and technology-transfer projects, such as the Horizon-2020 European project SOPHIA, the ERC grant Ergo-Lean, and the Horizon-2020 European project CONCERT. Dr. Lamon was awarded as finalist for the European best thesis in robotics at the Georges Giralt PhD Award 2022, as finalist for the best paper award on mobile manipulation at IROS 2022, for the best paper at I-RIM (2021 & 2019), and for the MECSPE Solution Award 2020. His interests lie in the intersection of human-robot collaboration, artificial intelligence, robot control and learning, with the aim of boosting robotics in human-populated environments.

Yabin Wang (S'22) received his B.S. degree in naval architecture and marine engineering from Dalian University of Technology, Dalian, China, in 2018, and the M.E. degree in software engineering from Xi'an Jiaotong University, Xi'an, China, in 2021. Currently, he is working toward the PhD degree at Xi'an Jiaotong University, Xi'an, China. His research interests include incremental learning, pattern recognition, machine learning, and deep learning.

Xiaopeng Hong (M'13, SM'23) received his Ph.D. degree from HIT in 2010. He is a professor at Harbin Institute of Technology (HIT), P. R. China. He had been a distinguished research fellow at Xi'an Jiaotong University, P. R. China, and an adjunct professor at the University of Oulu, Finland. He has authored over 50 articles in journals and conferences such as IEEE T-PAMI, CVPR, ICCV, and AAAI. His studies about subtle facial movement analysis was reported by International media like MIT Technology Review. He was the co-author of a 'top paper award' paper in ACM Multimedia 2023 and also the 2020 'IEEE Finland Section best student conference paper'. His current research interests include incremental learning, crowd counting, and affective computing.