2024 IEEE Conference on Control Technology and Applications (CCTA)
August 21-23, 2024. Newcastle upon Tyne, UK

2024 IEEE Conference on Control Technology and Applications (CCTA) | 979-8-3503-7094-2/24/$31.00 ©2024 IEEE | DOI: 10.1109/CCTA60707.2024.10666642

Optimal Real-Time Human Attention Allocation and Scheduling in a Multi-human and Multi-robot Collaborative System

Ningshi Yao* Renke Wang

Abstract— **Our earlier work developed a contention-resolving model predictive control (or MPC) framework to optimally schedule a single human operator's attention to collaborate with multiple robots. In this paper, we present a generalized design that can allocate and schedule the limited human attention in a multi-human and multi-robot collaborative system. We establish a new analytical timing model for multi-resource and multi-task real-time systems, where collaboration tasks can be either preemptive or non-preemptive. Then we derive the condition to predict the moments when contentions occur among the collaboration requests to humans. The contention-resolving MPC is triggered at contention moments to dynamically schedule human attention and determine the robot with which each human operator should collaborate. The optimal schedule, which aims to maximize the overall robots performance, is computed using an event-triggered and sampling-based approach with a weighted decision tree. To further improve computation efficiency, we present a new rule to merge leaves and simplify the decision tree. This paper also introduces a reset time for contention-resolving MPC, shortening the time horizon to search the optimal solution and reduce computational requirements. The effectiveness of the proposed method is verified through simulations.**

I. INTRODUCTION

With the rapid progress of robotic technology, it can be envisioned that humans will collaborate with robots in the very near future in many scenarios [1], such as smart manufacturing, underground mining, search and rescue, and surveillance tasks. In those scenarios, robots can work for long operation hours on repetitive tasks, providing consistent and precise performance beyond human capability, while human operators are better at intuitive decision making and working with uncertainties. The multi-human multi-robot (or MH-MR) collaboration then has the advantage of combining the strength of both robots and humans and becomes a promising setup for future work spaces. Research on collaboration between humans and robots has gained a lot of momentum [2]–[5]. However, when it is applied, one constraint for humans is limited attention capacity. Psychology studies [6], [7] has revealed that one human can only efficiently pay attention to two to four items simultaneously. Due to this limitation, each human should collaborate with which subset of robots needs to be determined [8].

Given the variety of possible combinations of heterogeneous robots and the scale of MH-MR systems, developing a generalized framework for human attention allocation is challenging [9]–[14]. A multi-level programming model was proposed to allocate multiple robot and human agents to maximize the effectiveness of the entire system with limited resources [15]. Gombolay et al. [16] present a centralized task assignment algorithm using a mixed-integer program solver. In addition to human-robot allocation, properly schedule human attention to better assist robots is another important problem for MH-MR systems. Well-known scheduling policies like rate monotonic scheduling (or RMS), which schedules the system with the smallest period first, and earliest deadline first (or EDF) [17], which schedules the system with the most urgent deadline first, can be used for multi-resource real-time systems. But these algorithms are designed only for real-time scheduling purposes and are optimal for minimizing the mean waiting time [18]. Other scheduling strategies, such as first come first serve [19], the shortest job first [20] and the highest trust first scheduling [21] were proposed to ensure fairness, efficiency or maintain trust level between humans and robots but cannot guarantee optimal performance. There are fewer recent studies [22]–[24] that use optimization-based methods to solve scheduling problems in human and robot collaboration. However, these existing methods rely on optimization solvers or genetic algorithms to obtain the optimal solution, which results in difficulties for real-time computing when considering a relatively large number of collaboration tasks or repeating collaboration requirements.

Our previous work suggested a contention-resolving MPC design that is a promising general method to solve real-time scheduling and task allocation problem with the applications in traffic intersection management [25], [26], networked control systems [27]–[29], and a single-human and multi-robot collaboration system [30]. However, all those works so far only focus on scheduling systems with a single shared resource. The real-time scheduling problem with multiple shared resources, more general and common in reality, is much more challenging and complex. Our previous work assumed purely non-preemptive human-robot collaboration, meaning it cannot be interrupted until finished. However, in reality, robot collaboration involves both preemptive and non-preemptive tasks. An example of preemptive collaboration is a robot requesting human's help to detect targets from its onboard camera, which can be interrupted by other more important tasks. Such a mix of preemptive and non-preemptive tasks cannot be handled by our previous work.

In this paper, we present a contention-resolving MPC to dynamically schedule multiple shared resources for the FIRST time, applied to an MH-MR collaboration system

This work is supported by NSF grant ECCS-2218517.

Ningshi Yao and Renke Wang are with the Department of Electrical and Computer Engineering, George Mason University, Fairfax, VA, 22030, USA. Email: nyao4@gmu.edu, rwang26@gmu.edu
*Send correspondence to Ningshi Yao.

where each human is regarded as a shared resource. Our method aims to find the optimal allocation of collaboration tasks and the attention schedule of each human operator to maximize the overall performance of the robots. The contributions of this paper are as follows.

1. We model the MH-MR systems as parallel machines in the job scheduling and develop a generalized timing model to rigorously describe time evolution in a multi-resource multi-task real-time system. To the best of our knowledge, this is the first analytical model for multiple resources. Timing models for a single resource system from our previous works are special cases of this generalized timing model.
2. The new proposed timing model unifies the definition of significant moments for preemptive and non-preemptive collaborative tasks, previously studied separately in [21] and [30]. It predicts the critical timings when contentions occur, considering both preemptive and non-preemptive tasks in the collaboration system. At each contention time, the contended robots and available human operators are identified based on timing states values computed by the new timing model.
3. We develop a generalized contention-resolving MPC to optimize scheduling for the MH-MR collaborative system. We present a new rule to merge leaves in decision tree construction, significantly reducing the number of leaves and branches. Also, we propose a novel reset time concept, converting the search for minimal cost path in the entire tree into searching for minimal cost segments in several sub-trees. Such conversion improves the computation efficiency of the contention-resolving MPC algorithm. The effectiveness of the proposed method is verified by simulations.

II. Problem Formulation

We consider M human operators collaborating with N robots $(1 \leq M < N)$. Since human operators have higher intelligence in decision making and adaptation to uncertainties than robots, we assume that robots can improve their performance with human help through collaboration.

A. Robot Performance Model

For a robot i where $i = 1, ..., N$, we consider two modes depending on whether a human collaborates with it or not: the autonomous mode and the interactive mode. The robot performance, denoted as $P_i(k)$ where k represents a discrete time step, can be quantified according to [31] as

$$P_i(k) = u_i(k)\left[(1-k_{i,H})P_i(k-1) + k_{i,H}P_{i,\max}\right] + [1-u_i(k)]\left[(1-k_{i,R})P_i(k-1) + k_{i,R}P_{i,\min}\right], \quad (1)$$

where $P_{i,\min}$ and $P_{i,\max}$ are the minimal and maximal bounds of the robot i's performance value. The parameters $k_{i,R}$ and $k_{i,H}$ are coefficients for autonomous and collaborative modes, respectively, and we assume $0 < k_{i,H} < k_{i,R} < 1$. The control variable $u_i(k)$ is a binary integer that indicates whether robot i is in autonomous mode, or in collaborative mode with a human. When a robot is in autonomous mode, $u_i(k) = 0$ and the performance value $P_i(k)$ is a convex combination of $P_i(k-1)$ from the previous time step and the lower bound $P_{i,\min}$ due to the range of $k_{i,R}$. Thus, $P_i(k)$ will decrease in the autonomous mode. When a robot is in collaborative mode, $u_i(k) = 1$ and the performance value $P_i(k)$ is a convex combination of $P_i(k-1)$ and the upper bound $P_{i,\max}$. Thus, $P_i(k)$ will increase in collaborative mode. Given that the initial performance values are within $[P_{i,\min}, P_{i,\max}]$, it can be easily shown that $P_i(k)$ is always within $[P_{i,\min}, P_{i,\max}]$ for any k.

B. Collaboration Task Model

Each robot will perform a sequence of repeating tasks, denoted as $\{\tau_{i,1}, \tau_{i,2}, ..., \tau_{i,n_i}, ...\}$, where i is the robot index and n_i is the task index. We assume that all tasks are periodic with a known period T_i. For any n_i, $C_i(n_i)$ is the collaboration time that robot i requires to collaborate with one human operator for task τ_{i,n_i}, satisfying $1 \leq C_i(n_i) < T_i$ for all i and n_i. In an MH-MR system, collaboration task allocation and human attention scheduling can be modeled as a parallel machine scheduling [18] with following assumptions.

Assumption 1: Each human operator has the same set of skills and expertise, i.e., the human resources are identical. Therefore, the duration of the collaboration time $C_i(n_i)$ does not depend on which human operator robot i is assigned.

Assumption 2: Each collaboration cannot be divided and accomplished by two or more than two human operators simultaneously, meaning that each robot can only collaborate with one human at any time.

The time robot i starts to execute the n_ith task is denoted by $\alpha_i(n_i)$. At each $\alpha_i(n_i)$, the robot performance $P_i(\alpha_i(n_i))$ is reset to $P_i^0(n_i) \in [P_{i,\min}, P_{i,\max}]$ because each task in the task sequence can be very different. A collaboration completion time $\gamma_i(n_i)$ is the time step when robot i finishes collaborating with a human operator. Since we modeled the systems in discrete time, all T_i, C_i, α_i, and γ_i are integers. In addition to the timing parameters, we also introduce a parameter $\phi_i(n_i)$, indicating whether a collaboration task τ_{i,n_i} is non-preemptive. Define $\phi_i(n_i) = 0$ if τ_{i,n_i} is preemptive and $\phi_i(n_i) = 1$ if it is non-preemptive.

Due to *Theorem 1* in [30], to maximize overall robot performance, the optimal strategy for each robot is to obey the condition of immediate access (or CIA), meaning that every robot should start collaboration immediately at $\alpha_i(n_i)$ if possible. However, the CIA condition is not always possible for all robots due to limited resources.

C. Task Allocation and Human Attention Scheduling

To design the allocation, we introduce a selector variable $v_{i,m}(k)$ where $i = 1, \ldots, N$ and $m = 1, \ldots, M$.

Definition 1: The selector variable $v_{i,m}(k)$ is a binary integer. If the current task of robot i is assigned to human operator m at k, then $v_{i,m}(k) = 1$. Otherwise, $v_{i,m}(k) = 0$.

All selector variables for human m lead to a vector $\mathbf{V}_m(k) = [v_{1,m}(k), ..., v_{N,m}(k)]^\top$, denoting the schedule for human m at k. If $v_{i,m}(k) = 1$, human m is scheduled to collaborate with robot i at k. Due to the limitation of human attention capacity, we make the following assumption:

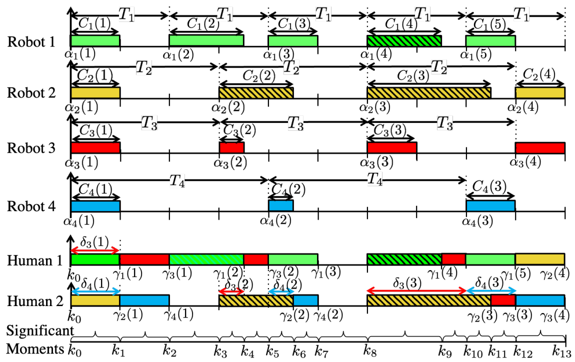

Fig. 1. Illustration of scheduling 4 robots to 2 humans. The upper four sub-figures show collaboration tasks of each robot where the colored rectangles denote preemptive tasks and shaded rectangles denote non-preemptive tasks. The lower two sub-figures show the humans attention schedule.

Assumption 3: At any given time, each human operator can collaborate with at most one robot, i.e.,

$$\sum\nolimits_{i=1}^{N} v_{i,m}(k) \leq 1, \text{ for any } m \text{ and } k. \quad (2)$$

This is valid because, in most human-robot collaborations, humans must manually control or pay close attention to monitor robots' behavior. It is risky for a human to work with two robots at the same time.

All column vectors $\mathbf{V}_m(k)$ together contribute to a selector matrix $\mathcal{V}(k)=[\mathbf{V}_1(k), ..., \mathbf{V}_m(k), ..., \mathbf{V}_M(k)]$. Based on *Assumption 2*, we have the constraint

$$\sum\nolimits_{m=1}^{M} v_{i,m}(k) \leq 1, \text{ for any } i \text{ and } k. \quad (3)$$

Then based on the definitions of $v_{i,m}(k)$ and $u_i(k)$,

$$u_i(k) = \sum\nolimits_{m=1}^{M} v_{i,m}(k), \text{ for any } i \text{ and } k. \quad (4)$$

Remark 1: When $M=1$, we have $u_i(k)=v_{i,1}(k)$, which means that $v_{i,1}(k)$ and $u_i(k)$ are equivalent if there is only one human operator in the collaboration system, which is consistent with the scheduling model used in [30].

Due to constraints (2) and (3), we derive $\sum_{i=1}^{N}\sum_{m=1}^{M} v_{i,m}(k) \leq M$ for any k, meaning that at most M robots can be scheduled to humans at any time. If the number of robots collaborating with a human at k plus those that have started new task requests at k exceeds M, a contention occurs. Thus, some robots cannot collaborate with humans immediately, delaying their task completion time. To quantify the delay, we introduce the delay variable $\delta_i(n_i) \geq 0$ so that

$$\gamma_i(n_i) = \alpha_i(n_i) + \delta_i(n_i) + C_i(n_i). \quad (5)$$

Consider an example shown by Figure 1 where 4 robots request to collaborate with 2 humans at k_0. Since the total number of requests at k_0 is greater than that of humans, contention occurs at k_0. Let robot 1 collaborate with human 1 and robot 2 with human 2 at k_0, robots 3 and 4 wait to humans later. Due to the occupation of humans by robots 1 and 2, robots 3 and 4 have time delays $\delta_3(1)$ and $\delta_4(1)$, shown by red and blue arrows on the left. If we exchange the schedule between robots 2 and 4, i.e., exchange the value of $v_{2,1}(k)$ and $v_{4,1}(k)$ for $k \in [\alpha_1(1), \alpha_1(2)]$, robot 2 has a time delay but robot 4 will not. This simple example shows that time delay variables $\delta_i(n_i)$ depend on the value of $v_{i,m}(k)$. In Section III, we present a timing model that can accurately compute $\delta_i(n_i)$ given $v_{i,m}(k)$ for all i, m, and k.

D. Optimization Formulation

A robot collaboration task allocation and human attention scheduling problem can be formulated with the selector matrix $\mathcal{V}(k)=[\mathbf{V}_1(k), ..., \mathbf{V}_m(k), ..., \mathbf{V}_M(k)]$. Given the initial robot performance $(P_1^0(n_1), ..., P_i^0(n_i), ..., P_N^0(n_N))$ for all i and n_i, the optimal task allocation and scheduling problem is to find the optimal $\mathcal{V}^*(k)$ by solving the optimization problem in a planning horizon $[k_0, k_f]$

$$\min_{\mathcal{V}(k)} \sum_{i=1}^{N} \sum_{k=k_0}^{k_f} [P_{i,\max} - P_i(k)] \text{ s.t. } (1), (2), (3) \text{ and } (4), \quad (6)$$

where the cost function aims to increase the robot performance as much as possible to reach its upper bound. Equations (1) and (4) are system dynamics. (2) and (3) are the contention constraints. Since $\mathcal{V}(k)$ is a matrix of binary integers, the problem is binary optimization. It is a non-convex optimization problem with $M \times N \times (k_f - k_0)$ decision variables. If we consider a relatively large number of robots and a long time horizon, the optimization problem in (6) will be very difficult to solve.

Instead of solving (6), which is commonly adopted for MH-MR allocation problems [22]–[24], for each time step, we formulate this problem in an equivalent way

$$\begin{aligned}
&\min_{\mathcal{V}(k)} \sum_{i=1}^{N} \sum_{k=k_0}^{k_f} [P_{i,\max} - P_i(k)] \text{ s.t. } (1), \qquad (7)\\
&u_i(k)=0, k \in [\alpha_i(n_i), \alpha_i(n_i)+\delta_i(n_i)(\mathcal{V}(k)) - 1],\\
&u_i(k)=1, k \in [\alpha_i(n_i)+\delta_i(n_i)(\mathcal{V}(k)), \gamma_i(n_i)(\mathcal{V}(k))] \text{ and}\\
&u_i(k)=0, k \in [\gamma_i(n_i)(\mathcal{V}(k))+1, \alpha_i(n_i+1)-1]\\
&\text{for all } n_i \text{ such that } k_0 \leq \alpha_i(n_i) \text{ and } \alpha_i(n_i+1) \leq k_f,
\end{aligned}$$

where the notations $\delta_i(n_i)(\mathcal{V}(k))$ and $\gamma_i(n_i)(\mathcal{V}(k))$ mean that these time instants are implicit functions of the decision variables in matrix $\mathcal{V}(k)$. The contention constraints (2)-(4) are embedded with the implicit functions to compute $\delta_i(n_i)$ and $\gamma_i(n_i)$, which will be presented in the next section.

The benefit of this equivalent formulation is that it can enable an event-triggered MPC design, which uses an analytical timing model to efficiently compute $\delta_i(n_i)$ and $\gamma_i(n_i)$ for any i and n_i. The event-triggered MPC only needs to calculate the decision variables at the event times instead of every time step, which can significantly reduce the computation.

III. Dynamic Timing Model

The critical events for real-time scheduling of human attention are contentions. Contentions can only occur at $\alpha_i(n_i)$, the time when a new request is generated from a robot, or $\gamma_i(n_i)$, the time when a robot finishes the collaboration with a human. Thus, the moments $\alpha_i(n_i)$ and $\gamma_i(n_i)$ are more *significant* than other times in real-time scheduling. We developed a method called significant moment analysis (or

SMA), which studies the system's timing behavior based on the *significant moments*, enabling us to establish a dynamic model to compute when contentions will occur.

A. Timing States

To develop a dynamic model that describes how timing evolves, we define the timing states as $Z(k) = (D(k), R(k), O(k), \text{ID}(k))$ same as in [21]. $D(k) = (d_1(k), ..., d_i(k), ..., d_N(k))$ is the deadline variable, where $d_i(k)$ is how long after time k the next generation of task τ_{i,n_i} will occur. $R(k) = (r_1(k), ..., r_i(k), ..., r_N(k))$ is the remaining time variable, where $r_i(k)$ is the remaining time needed to complete the collaboration of the task generated the most recently τ_{i,n_i} after k. $O(k) = (o_1(k), ..., o_i(k), ..., o_N(k))$ is the response variable, where $o_i(k)$ is how long the completion of the task τ_{i,n_i} has been delayed from its most recent request time $\alpha_i(n_i)$ to k. The difference between this paper and [21] lies in $\text{ID}(k)$.

Definition 2: The index variable $\text{ID}(k)$ is a set $\{id_1(k), \ldots, id_m(k), \ldots, id_M(k)\}$, where $id_m(k)$ is the index of the robot that is collaborating with the human operator m at time k. If $id_m(k) = i$, it means that human m is collaborating with robot i at k, and $id_m(k) = 0$ implies that no robot is occupying human m's attention at time k.

To establish a timing model for the MH-MR collaborative system, we need to redefine the collaboration time C_i and the preemptive or non-preemptive parameter ϕ_i of a task.

Definition 3: For all i, $n_i \geq 0$, we set $C_i(k) = C_i(n_i)$ and $\phi_i(k) = \phi_i(n_i)$ for $k \in [\alpha_i(n_i), \alpha_i(n_i+1))$.

The evolution rules for $Z(k)$ can be expressed mathematically. These equations lead to a dynamic model to describe how time evolves based on specific human attention allocation. It is analytical and efficient to compute, which supports the implementation of contention-resolving MPC.

B. Timing Model

We divide $[k_0, k_f]$ into sub-intervals by the *significant moments*. Let k_w and k_{w+1} be two successive significant moments. Due to the definition, the collaboration generation or completion only occurs at k_w or k_{w+1}, but not at any time within $[k_w+1, k_{w+1}-1]$. In Figure 1, k_0 to k_{13} are significant moments of the example and how to compute them is as follows.

Initially, we set the beginning of the optimization horizon k_0 as the first significant moment. Then, using mathematical induction, other significant moments can be computed iteratively based on timing states values. Assume that the significant moment k_w has been calculated.

Case 1: If none of the human operators is working with any robot at k_w, i.e., $id_m(k_w) = 0$ for all m, i.e., $\sum_{m=1}^{M} id_m(k_w) = 0$, then the next significant moment is the nearest collaboration generation time. So, the difference between them is $k_{w+1} - k_w = \min\{d_1(k_w), ..., d_N(k_w), k_f - k_w\}$.

Case 2: If the attention of some humans is occupied by robot $id_m(k_w)$ at k_w, i.e., $\sum_{m=1}^{M} id_m(k_w) > 0$, then in addition to the above requirement, $k_{w+1} - k_w$ should be less than or equal to $r_{id_m}(k_w)$ so that the closest completion time $k_w + r_{id_m}(k_w)$ of the collaboration task from robot $id_m(k_w)$ to human m is not less than k_{w+1}. Here $r_{id_m}(k)$ is a simplified notation for the remaining time $r_{id_m(k)}(k)$ of timing state variable id_m at any k. Similar simplifications apply to $d_{id_m(k)}(k)$, $o_{id_m(k)}(k)$, $\phi_{id_m(k)}(k)$, and $v_{id_m(k),m}(k)$. If there is more than one human whose attention is occupied by robots, let $r_{\min}(k_w)$ be the smallest remaining time of $r_{id_m}(k_w)$ among all the tasks at k_w, i.e., $r_{\min}(k_w) = \min\{r_{id_m}(k_w) : id_m(k_w) > 0\}$. Summarizing the above two cases, we have

$$k_{w+1} - k_w = \tag{8}$$
$$\mathsf{sgn}\left[\sum_{m=1}^{M} id_m(k_w)\right] \min\{r_{\min}(k_w), d_1(k_w), ..., d_N(k_w), k_f - k_w\}$$
$$+ \left\{1 - \mathsf{sgn}\left[\sum_{m=1}^{M} id_m(k_w)\right]\right\} \min\{d_1(k_w), ..., d_N(k_w), k_f - k_w\}$$

for all w where $\mathsf{sgn}(q) = 1$ if $q > 0$ and $\mathsf{sgn}(q) = 0$ if $q = 0$.

After dividing the optimization horizon into sub-intervals $[k_w, k_{w+1})$, the evolution of $Z(k)$ within any sub-interval can be derived at k_w and within interval (k_w, k_{w+1}).

At significant moment k_w, timing states will have jumps. The changes of the state vector $(d_i(k_w), r_i(k_w), o_i(k_w))$ are the same for preemptive and non-preemptive tasks, depending on whether a new task of robot i is generated at k_w.

Case 1: If a new task of robot i is generated at k_w, which means that the deadline variable of robot i satisfies $d_i(k_w - 1) = 1$, then three timing states are reset to be

$$d_i(k_w) = T_i, r_i(k_w) = C_i(k_w), o_i(k_w) = 0. \tag{9}$$

Case 2: If $d_i(k_w - 1) > 1$, i.e., the next task of robot i will not generate at k_w, there is no jump in the timing states of robot i. The deadline $d_i(k_w - 1)$ decreases by 1 for one time step. The response variable $o_i(k_w)$ will increase by 1 if the collaboration has not been finished, i.e., $r_i(k_w - 1) > 0$. If the collaboration has been completed before k_w, i.e., $r_i(k_w - 1) = 0$, then $o_i(k_w) = o_i(k_w - 1)$. For the remaining time variable $r_i(k_w)$, if robot i is collaborating with a human at $k_w - 1$, i.e., $i \in \text{ID}(k_w - 1)$, then $r_i(k_w - 1)$ decreases by 1 for one time step. If i is not in the index set $\text{ID}(k_w - 1)$, then $r_i(k_w - 1)$ remains the same. Let $\mathbb{1}(\cdot)$ be an indicator function which is defined to be 1 if the condition $i \in \text{ID}(k_w - 1)$ holds and 0 otherwise. We can present the changes as

$$d_i(k_w) = d_i(k_w - 1) - 1, r_i(k_w) = r_i(k_w - 1) - \mathbb{1}(i \in \text{ID}(k_w - 1)),$$
$$\text{and } o_i(k_w) = o_i(k_w - 1) + \text{sgn}(r_i(k_w - 1)). \tag{10}$$

For the index variable id_m at k_w, if $r_{id_m}(k_w - 1) > 1$, it means that robot $id_m(k_w - 1)$ has not completed the collaboration with human m at k_w. If the current task of robot id_m is non-preemptive, i.e., $\phi_{id_m}(k_w - 1) = 1$, then $id_m(k_w) = id_m(k_w - 1)$ as the collaboration cannot be interrupted until it finishes. If the current task of robot id_m is preemptive, i.e., $\phi_{id_m}(k_w - 1) = 0$, $id_m(k_w)$ must switch to the robot scheduled to collaborate with human m at k_w, i.e., robot i with $v_{i,m}(k_w) = 1$. If $r_{id_m}(k_w - 1) = 1$, robot $id_m(k_w - 1)$ will complete collaboration at k_w, then $id_m(k_w)$ also switches to robot with $v_{i,m}(k_w) = 1$ for either preemptive or non-preemptive tasks. Combining the above cases, the evolution rule for $id_m(k)$ at k_w is

$$\begin{aligned} id_m(k_w) = {} & id_m(k_w{-}1)\mathrm{sgn}[r_{id_m}(k_w{-}1){-}1]\phi_{id_m}(k_w{-}1) \\ & + \underset{i}{\mathrm{argmax}}[v_{i,m}(k_w)]\big\{1{-}\mathrm{sgn}[r_{id_m}(k_w{-}1){-}1]\phi_{id_m}(k_w{-}1)\big\}. \end{aligned} \quad (11)$$

For any time $k_w+\epsilon \in [k_w+1, k_{w+1}-1]$, the timing state $id_m(k_w+\epsilon)$ does not change since $k_{w+1}-k_w \le r_{id_m}(k_w)$ for any m. If $id_m(k_w) \neq 0$, meaning that robot $id_m(k_w)$ is assigned to human m in $[k_w+1, k_{w+1}-1]$, we have

$$\begin{aligned} & d_{id_m}(k_w{+}\epsilon) = d_{id_m}(k_w) - \epsilon, r_{id_m}(k_w{+}\epsilon) = r_{id_m}(k_w) - \epsilon, \\ & o_{id_m}(k_w{+}\epsilon) = o_{id_m}(k_w) + \epsilon, \text{and } v_{id_m,m}(k_w{+}\epsilon) = 1. \end{aligned} \quad (12)$$

For all i such that $i \notin \mathrm{ID}(k_w)$, we have

$$\begin{aligned} & d_i(k_w+\epsilon) = d_i(k_w) - \epsilon, \ r_i(k_w+\epsilon) = r_i(k_w) \quad (13) \\ & o_i(k_w+\epsilon) = o_i(k_w) + \mathrm{sgn}(r_i(k_w))\epsilon, \text{and } v_{i,m}(k_w{+}\epsilon) = 0. \end{aligned}$$

The analytical equations from (8) to (13) together form the generalized timing model for MH-MR collaboration scheduling, represented by $\mathbb{H}(\cdot)$. Given the initial state $Z(k_0)$, the task timing parameter $C_i(n_i)$, the period T_i, the non-preemptiveness parameter $\phi_i(n_i)$ for all i and n_i, and the value of $\mathcal{V}(k_0 \sim k)$, a simplified notation for decision variables $\mathcal{V}(\kappa)$ for any $\kappa \in [k_0, k]$, this analytical model can compute $Z(k)$ at each k as

$$Z(k) = \mathbb{H}(k; Z(k_0), (C_i(n_i), T_i, \phi_i(n_i))_{i=1,\ldots,N}, \mathcal{V}(k_0 \sim k)).$$

Based on the definition of the response variable $O(k)$, the time delay $\delta_i(n_i)$ can be calculated as

$$\delta_i(n_i) = o_i(\alpha_i(n_i{+}1){-}1) + \mathrm{sgn}(r_i(\alpha_i(n_i{+}1){-}1)) - C_i(n_i).$$

Remark 2: Through (8) to (13), the only case where the values of the timing states depend on the decision variable $v_{i,m}$ occurs in (11) at the significant moments k_w. Thus, we only need to determine decision variables $v_{i,m}$ at k_w.

IV. Contention-Resolving MPC Algorithm

In this section, we convert the problem formulated by (7) into a path planning problem among a weighted decision tree, which can be solved iteratively. Finding an optimal solution to (7) is equivalent to finding a path from the root to the end leaf of the decision tree with minimal cost.

A. Construction of Decision Tree

Due to *Remark* 2, decision variables $v_{i,m}(k)$ only need to be determined at significant moments, which trigger the decision tree generation. Figure 2 shows an example of a decision tree where each layer of leaves represents one significant moment computed by (8). Each leaf is associated with a significant moment. For an arbitrary leaf indexed by l with the significant moment k_w, a branch pointing out from leaf l is associated with a specific choice of $v_{i,m}(k_w)$ and connects to a leaf in the next layer. Like the decision tree proposed for a single human scheduling in [30], the construction of the whole tree is not needed for the contention-resolving MPC algorithm for the MH-MR

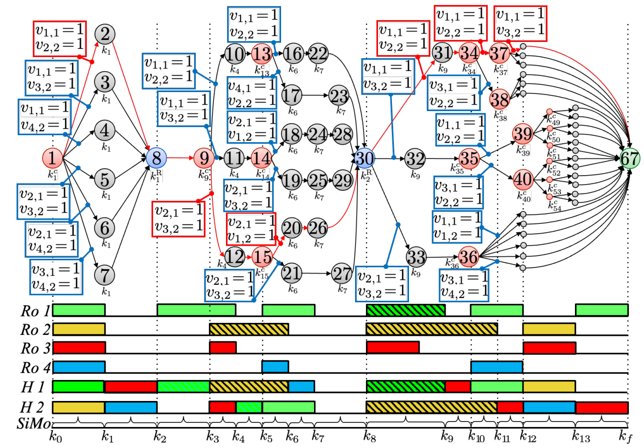

Fig. 2. Decision tree for solving integer optimization in a finite time window. Grey circles denote the leaves without contention, red circles denote the leaves with contention, and blue circles denote the merged leaves. The green circle denotes the end of time. The decision tree expands along the arrowed branches. Colored rectangles show an example of the task allocation and the attention schedule for humans along the path with red arrows.

system. However, to clearly present our solution, here we briefly describe how the tree can be fully constructed.

For each leaf l, we identify $\Lambda(l)$ as a subset of robots requesting collaboration or currently collaborating with a human by timing states. If a request from robot i is generated at k_w, we must have $r_i(k_w) = C_i(k_w)$ due to (9) and (10). If robot i was collaborating with a human right before k_w, then $0 < r_i(k_w) < C_i(k_w)$. In both cases, we have $r_i(k_w) > 0$, so $\Lambda(l) = \{i : r_i(k_w) > 0, \ i = 1, \ldots, N\}$. The branches number from leaf l depends on whether contentions occur at k_w associated with l. The following *Lemma* can check it.

Lemma 1: A contention happens at a significant moment k_w if and only if $\sum_{i=1}^{N} \mathrm{sgn}\,[r_i(k_w)] > M$.

Proof. Since each robot in the set $\Lambda(l)$ must satisfy $r_i(k_w) > 0$, the total number of elements in $\Lambda(l)$ can be written as $\sum_{i=1}^{N} \mathrm{sgn}(r_i(k_w))$. If this number is greater than the number of humans, then there is a contention occurring at k_w. □

Based on *Lemma* 1, k_0, k_3, k_5, k_8, k_{10}, k_{11} and k_{12} of Figure 1 are the significant moments when contention occurs.

If no contention occurs at k_w, i.e., $\sum_{i=1}^{N} \mathrm{sgn}\,[r_i(k_w)] \le M$, the assignment of robots to humans does not affect the robot performance $P_i(k)$ in $[k_w, k_{w+1}]$, as humans are identical and $C_i(k)$ are the same for each. Any feasible $v_{i,m}(k_w)$ will contribute equally to the cost function in (7). Leaves with no contention are denoted by grey circles in Figure 2, with only one branch extending from them. Robots with ongoing non-preemptive tasks at k_w are represented as set $\Lambda_{\mathrm{NP}}(l) = \{id_m(k_w{-}1) : [r_{id_m}(k_w{-}1){-}1]\phi_{id_m}(k_w{-}1) > 0\}$, and the decision variable $v_{i,m}(k_w)$ for robots in $\Lambda_{\mathrm{NP}}(l)$ are determined at k_w as they have to continue the collaboration with the human assigned previously, i.e., $v_{i,m}(k_w) = v_{i,m}(k_w - 1)$ for all $i \in \Lambda_{\mathrm{NP}}(l)$. Denote the set of humans occupied by robots in $\Lambda_{\mathrm{NP}}(l)$ as $\Psi_{\mathrm{NP}}(l)$, represented as $\Psi_{\mathrm{NP}}(l) = \{m : [r_{id_m}(k_w{-}1){-}1]\phi_{id_m}(k_w{-}1) > 0\}$. Then the rest of the robots that request collaboration at k_w are denoted as $\Lambda(l) \setminus \Lambda_{\mathrm{NP}}(l)$, with $N_{\mathrm{P}}(l)$ as their number. The humans available at k_w are represented as $\{1, \ldots, M\} \setminus \Psi_{\mathrm{NP}}(l)$, with

$M_{\rm P}(l)$ as their number. Since no contention occurs, we have $N_{\rm P}(l) \le M_{\rm P}(l)$. Then, a feasible choice of $v_{i,m}(k_w)$ is to arrange robots with preemptive tasks and available humans by increasing indices, assigning the first $N_{\rm P}(l)$ humans to each robot in the increasing index order.

However, if contention occurs at k_w, i.e., $\sum_{i=1}^{N}{\rm sgn}[r_i(k_w)] > M$, we denote the k_w as contention time k_l^c, where l denotes its corresponding leaf. In Figure 2, red circles represent the leaves with the contentions. At k_l^c, the performance of robots without immediate collaboration will decrease, so different choices of $v_{i,m}(k_w)$ lead to different costs in $[k_w, k_{w+1}]$. Hence, each contention leaf will have multiple branches, and we only need to design $v_{i,m}(k_w)$ when contentions occur. Same as the leaves with no contention, the decision variable $v_{i,m}(k_w)$ for robots in $\Lambda_{\rm NP}(l)$ with non-preemptive tasks is determined at k_w and cannot be chosen freely. For the rest of robots in $\Lambda(l)\setminus\Lambda_{\rm NP}(l)$, we choose $M_{\rm P}(l)$ robots and assign available humans to them. Thus, there will be $N_{\rm P}(l)!/\,[M_{\rm P}(l)!(N_{\rm P}(l)-M_{\rm P}(l))!]$ branches from each contention leaf. In Figure 2, a contention occurs to all four robots ($N_{\rm P}(1)=4$) at k_0 and both humans are available ($M_{\rm P}(1)=2$). So, we choose 2 robots out of 4 to collaborate with humans at k_0, leading to 6 choices of $v_{i,m}(k_0)$ corresponding to 6 branches.

Define a branch cost associated with each branch. Let the q-th branch from leaf l_1 and denote the decision variable matrix $\mathcal{V}(k)$ associated with q as $\mathcal{V}_q$, constants for all i and $k \in [k_w, k_{w+1}]$. Due to $\mathcal{V}_q$, timing states at k_w can be computed by (9),(10) and (11), and the next significant moment k_{w+1} scheduled under $\mathcal{V}_q$ can be computed by (8), leading to a new leaf l_2 at k_{w+1}. Then we can compute robots performance $\sum_{i=1}^{N}\sum_{k=k_w}^{k_{w+1}}[P_{i,\max}-P_i(k)]$ in $k\in[k_w, k_{w+1}]$ by the determined $\mathcal{V}_q$ and let it be the cost of branch (l_1, l_2). See Section IV.B of [30] for the specific formula.

Remark 3: The leaf generation rule in this paper enables a simplification method to merge leaves and reduce branches number, which will be introduced next.

B. Tree Structure Simplification

The decision tree will grow exponentially with a larger number of robots or a longer time horizon $[k_0, k_f]$. Thus, we propose rules to simplify it.

Lemma 2: For any two leaves l_a and l_b with the same associated significant moment k_w, if $\sum_{i=1}^{N} r_i(k_w) = 0$ for both leaves, l_a and l_b can be merged.

Proof. Since $\sum_{i=1}^{N} r_i(k_w) = 0$ means all robots have completed collaboration at k_w, they will start new collaborations with new initial performance $P_i^0(n_i)$ after k_w. That is, the system behavior after k_w will be the same for both leaves, allowing them to be merged and treated as one. □

Such a leaf merging rule can help simplify the structure of decision tree. Take the example at k_0 in Figure 2 again as an illustration. There are 6 different ways to allocate robots to humans at k_0 to resolve contention. Under each allocation, all robots can complete their first task before or right at k_2. After k_2, they will have new tasks. So, different task allocations at k_0 do not affect the system timing or robots' performance after k_2, allowing us to merge all the leaves at k_2 together. The same merging occurs at time k_8.

If all leaves at a significant moment can be merged into a single leaf, we call this significant moment a reset time, denoted as $t_p^{\rm R}$ where p is the index. Searching for the minimal path among the entire tree can be changed to find the minimal cost segments on the partial tree between each two successive reset times, simplifying the computation complexity for contention-resolving MPC. The optimal path for the entire tree in Figure 2 can be partitioned into three segments, which are the lowest cost paths from leaf 1 to 8, leaf 8 to 30, and leaf 30 to 67.

Based on the simplified decision tree with branch costs, the integer optimization problem in (7) can now be converted to the problem of finding a path from k_0 to k_f such that the whole cost along the path is the lowest. In our previous work, we presented an optimal path search algorithm that leverages the A-star algorithm to search for an optimal path in the decision tree. See [27], [28] for more details.

V. Simulation Results

We simulate 4 robots collaborating with 2 humans. The starting and ending times are $k_0 = 0$ and $k_f = 120$, respectively. The initial performance values are $P_i^0(n_i)=0.7$ for all i and n_i. The parameters for the performance model are $k_{i,R} = 0.25$ for all i, and $[k_{1,H}, k_{2,H}, k_{3,H}, k_{4,H}] = [0.2, 0.15, 0.13, 0.1]$. The lower and upper bounds are $P_{i,\min}=0.65$ for $i<4$, $P_{4,\min}=0.6$, and $P_{i,\max}=0.75$ for all i. The periods are $[T_1, T_2, T_3, T_4]=[20, 30, 30, 40]$. The collaboration times are $C_1(1)=C_1(3)=C_1(5)=C_1(6)=10$, $C_1(2)=C_1(4)=15$, $C_2(1)=C_2(4)=10$, $C_2(2)=15$, $C_2(3)=25$, $C_3(1)=C_3(3)=C_3(4)=10$, $C_3(2)=5$, $C_4(1)=C_4(3)=10$ and $C_4(2)=5$. Parameters for non-preemptiveness are $\phi_1(4)=\phi_2(2)=\phi_2(3)=1$ and the other $\phi_i=0$. Applying the algorithm, the humans attention occupation result is shown in Figure 3. The robot performance is shown in Figure 4. The total computation of our method only took 0.1182 seconds. Five contentions occur in $[0, 120]$ and the computation time to find the optimal solution is 0.41 seconds. The cost under optimal schedule considering the original cost function in (6) is 34.235, less than the RMS scheduling strategy cost of 36.086, where the priority of 4 robots is in descending order, meaning that robot 1 always has the highest priority. Our proposed method showed better performance than the state-of-the-art scheduling strategy, and the optimal solution can be obtained in real-time.

VI. Conclusions and Future Work

Coordinating multiple humans and robots to work together is a challenging problem and attracts increasing research interest. We present a novel method to allocate human resources to achieve maximal robot performance. In the future, we aim to extend this work to more general scenarios where humans have different capacities on specific tasks.

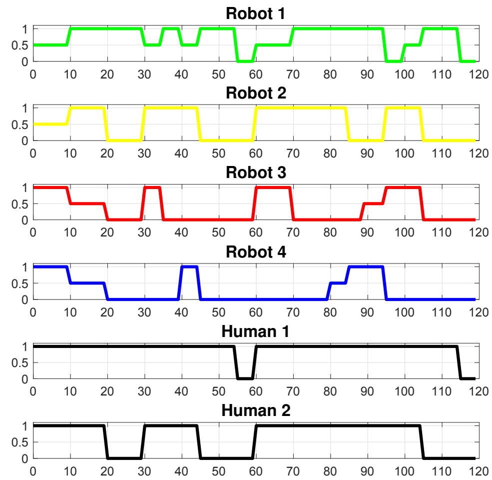

Fig. 3. Human attention occupation to collaborate with four robots. The y axis value 1 means that the robot is collaborating with a human, 0 means that the robot is not requesting the collaboration, and 0.5 means that the robot's collaboration request is delayed by a contention.

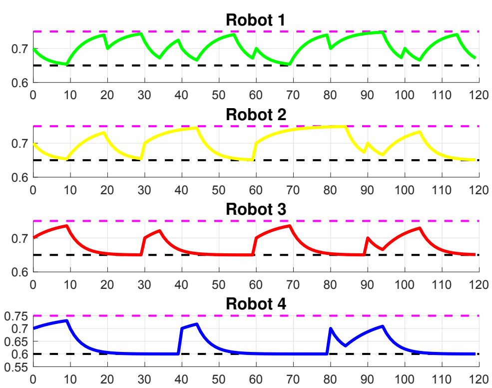

Fig. 4. Performance of four robots under the optimal schedule. The magenta dashed line represents $P_{i,\max}$ and the black dashed line represents $P_{i,\min}$.

REFERENCES

[1] A. Ajoudani, A. M. Zanchettin, S. Ivaldi, A. Albu-Schäffer, K. Kosuge, and O. Khatib, "Progress and prospects of the human–robot collaboration," *Autonomous Robots*, vol. 42, no. 5, pp. 957–975, 2018.

[2] L. Zhang and R. Vaughan, "Optimal robot selection by gaze direction in multi-human multi-robot interaction," in *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2016, pp. 5077–5083.

[3] V. Villani, F. Pini, F. Leali, and C. Secchi, "Survey on human–robot collaboration in industrial settings: Safety, intuitive interfaces and applications," *Mechatronics*, vol. 55, pp. 248–266, 2018.

[4] E. Matheson, R. Minto, E. G. Zampieri, M. Faccio, and G. Rosati, "Human–robot collaboration in manufacturing applications: A review," *Robotics*, vol. 8, no. 4, p. 100, 2019.

[5] F. Semeraro *et al.*, "Human–robot collaboration and machine learning: A systematic review of recent research," *Robotics and Computer-Integrated Manufacturing*, vol. 79, p. 102432, 2023.

[6] G. A. Miller, "The magical number seven, plus or minus two: Some limits on our capacity for processing information." *Psychological review*, vol. 63, no. 2, p. 81, 1956.

[7] N. Carr, *The shallows: What the Internet is doing to our brains*. WW Norton & Company, 2011.

[8] T. Yu, J. Huang, and Q. Chang, "Optimizing task scheduling in human-robot collaboration with deep multi-agent reinforcement learning," *Journal of Manufacturing Systems*, vol. 60, pp. 487–499, 2021.

[9] M. Majji and R. Rai, "Autonomous task assignment of multiple operators for human robot interaction," in *2013 American Control Conference*. IEEE, 2013, pp. 6454–6459.

[10] L. Johannsmeier and S. Haddadin, "A hierarchical human-robot interaction-planning framework for task allocation in collaborative industrial assembly processes," *IEEE Robotics and Automation Letters*, vol. 2, no. 1, pp. 41–48, 2016.

[11] T. Mina, S. S. Kannan, W. Jo, and B.-C. Min, "Adaptive workload allocation for multi-human multi-robot teams for independent and homogeneous tasks," *IEEE Access*, vol. 8, pp. 152 697–152 712, 2020.

[12] P. Gupta and V. Srivastava, "On robust and adaptive fidelity selection for human-in-the-loop queues," in *2021 European Control Conference (ECC)*, 2021, pp. 872–877.

[13] A. Ali, H. Azevedo-Sa, D. M. Tilbury, and L. P. Robert, "Heterogeneous human–robot task allocation based on artificial trust," *Scientific reports*, vol. 12, no. 1, pp. 1–15, 2022.

[14] F. Chen, K. Sekiyama, F. Cannella, and T. Fukuda, "Optimal subtask allocation for human and robot collaboration within hybrid assembly system," *IEEE Transactions on Automation Science and Engineering*, vol. 11, no. 4, pp. 1065–1075, 2013.

[15] M. S. Malvankar-Mehta and S. S. Mehta, "Optimal task allocation in multi-human multi-robot interaction," *Optimization Letters*, vol. 9, no. 8, pp. 1787–1803, 2015.

[16] 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.

[17] C. L. Liu and J. W. Layland, "Scheduling algorithms for multiprogramming in a hard-real-time environment," *Journal of the ACM*, vol. 20, no. 1, pp. 46–61, 1973.

[18] R. W. Conway, W. L. Maxwell, and L. W. Miller, *Theory of Scheduling*. Courier Corporation, 2003.

[19] U. Schwiegelshohn and R. Yahyapour, "Analysis of first-come-first-serve parallel job scheduling," in *SODA*, vol. 98. Citeseer, 1998, pp. 629–638.

[20] S.-Y. Chien, Y.-L. Lin, P.-J. Lee, S. Han, M. Lewis, and K. Sycara, "Attention allocation for human multi-robot control: Cognitive analysis based on behavior data and hidden states," *International Journal of Human-Computer Studies*, vol. 117, pp. 30–44, 2018.

[21] X. Wang, Z. Shi, F. Zhang, and Y. Wang, "Dynamic real-time scheduling for human-agent collaboration systems based on mutual trust," *Cyber-Physical Systems*, vol. 1, no. 2-4, pp. 76–90, 2015.

[22] Z. Shi, L. Wang, P. Liu, and L. Shi, "Minimizing completion time for order scheduling: Formulation and heuristic algorithm," *IEEE Transactions on Automation Science and Engineering*, vol. 14, no. 4, pp. 1558–1569, 2015.

[23] K. Bogner, U. Pferschy, R. Unterberger, and H. Zeiner, "Optimised scheduling in human–robot collaboration–a use case in the assembly of printed circuit boards," *International Journal of Production Research*, vol. 56, no. 16, pp. 5522–5540, 2018.

[24] A. Casalino, A. M. Zanchettin, L. Piroddi, and P. Rocco, "Optimal scheduling of human–robot collaborative assembly operations with time petri nets," *IEEE Transactions on Automation Science and Engineering*, vol. 18, no. 1, pp. 70–84, 2021.

[25] N. Yao, M. Malisoff, and F. Zhang, "Contention-resolving model predictive control for coordinating automated vehicles at a traffic intersection," in *2019 IEEE Conference on Decision and Control (CDC)*. IEEE, 2019, pp. 2233–2238.

[26] N. Yao and F. Zhang, "Contention-resolving model predictive control for an intelligent intersection traffic model," *Discrete Event Dynamic Systems*, pp. 1–31, 2021.

[27] N. Yao, M. Malisoff, and F. Zhang, "Contention resolving optimal priority assignment for event-triggered model predictive controllers," in *Proceedings of the 2017 American Control Conference*. IEEE, 2017, pp. 2357–2362.

[28] Y. Ningshi, M. Michael, and Z. Fumin, "Contention-resolving model predictive control for coupled control systems with a shared resource," *Automatica*, vol. 122, p. 109219, 2020.

[29] N. Yao and F. Zhang, "Event-triggered scheduling and control co-design for networked control systems with sub-schedulability," in *2022 American Control Conference (ACC)*. IEEE, 2022, pp. 1733–1738.

[30] Y. Ningshi and Z. Fumin, "Optimal real-time scheduling of human attention for a human and multi-robot collaboration system," in *2020 American Control Conference (ACC)*. IEEE, 2020, pp. 30–35.

[31] J. W. Crandall, M. A. Goodrich, D. R. Olsen, and C. W. Nielsen, "Validating human-robot interaction schemes in multitasking environments," *IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans*, vol. 35, no. 4, pp. 438–449, 2005.