2020 IEEE International Conference on Robotics and Automation (ICRA)
31 May - 31 August, 2020. Paris, France

Real-Time Adaptive Assembly Scheduling in Human-Multi-Robot Collaboration According to Human Capability*

Shaobo Zhang, Yi Chen, Jun Zhang, and Yunyi Jia

Abstract— **Human-multi-robot collaboration is becoming more and more common in intelligent manufacturing. Optimal assembly scheduling of such systems plays a critical role in their production efficiency. Existing approaches mostly consider humans as agents with assumed or known capabilities, which leads to suboptimal performance in realistic applications where human capabilities usually change. In addition, most robot adaptation focuses on human-single-robot interaction and the adaptation in human-multi-robot interaction with changing human capability still remains challenging due to the complexity of the heterogeneous multi-agent interactions. This paper proposes a real-time adaptive assembly scheduling approach for human-multi-robot collaboration by modeling and incorporating changing human capability. A genetic algorithm is also designed to derive implementable solutions for the formulated adaptive assembly scheduling problem. The proposed approaches are validated through different simulated human-multi-robot assembly tasks and the results demonstrate the effectiveness and advantages of the proposed approaches.**

I. INTRODUCTION

In developing intelligent manufacturing, human-robot collaborations have been becoming a new paradigm to leverage the capabilities of both humans and robots. During such collaborations, efficient scheduling plays an important role in improving production efficiency.

Many approaches have been proposed for human-robot assembly scheduling. A logic mathematic method is developed to quantitatively describe the subtask allocation problem by considering the system tradeoff between the assembly time cost and payment cost[1]. In order to support safe and efficient coordination between humans and robots, a centralized algorithm is presented, which handles tightly intercoupled temporal and spatial constraints[2]. A novel framework is proposed, which integrates both physical and social human-robot interaction factors into the robot motion controller for human-robot collaborative assembly tasks[3]. To minimize the cycle time through trajectory selection, task sequence and task allocation, an integrated motion planning and scheduling method is presented in [4]. To alleviate the information overload problem of human operators, a human operator is made responsible for overseeing autonomous agents and providing feedback based on sensor data[5]. In order to balance the required product and process characteristics, an approach based on a detailed analysis of the skills of humans and robots is adopted in [6].

However, these existing approaches for human-robot cooperative assembly largely consider humans as agents with assumed or known capabilities and performance. However, most existing approaches in such scheduling mainly assume that the human capability is known and unchanged. During human-robot collaborations, while the capabilities of robots usually remain unchanged, human capabilities are subject to change due to factors such as muscle fatigue, mental fatigue, workload, and environmental effects. This may lead to performance degradation in realistic applications due to dynamic changes in human capabilities. Therefore, it is critical to investigate this calculated capability issue and real-time adaptive scheduling.

Recently some robot adaptation approaches have been proposed in human-robot interaction. A time series trust model of human coworker to robot in a collaborative manufacturing task is proposed[7] in which human performance is modeled based on muscle fatigue and recovery dynamics. A robotic scheduling and control capability is proposed, which can adapt to the changing preferences of a human while providing strong guarantees for synchronization and timing of activities[8]. Some other robot adaptation approaches are also proposed to improve the human-robot efficiency and safety based on the detection of human states such as fatigue, distraction, motions and attributes [9][10][11]. However, these robot adaptation approaches mainly focus on human-single-robot interaction. The adaptation in human-multi-robot interaction considering changing human capability still remains a challenge due to the complexity of heterogeneous multi-agent interaction.

This paper presents a real-time adaptive human-multi-robot collaboration assembly scheduling problem according to the human capability changes. We propose a human capability model by considering measurable performance indices and then incorporate this model into the human-multi-robot assembly problem. We then propose a solution to dynamically generate the assembly schedule using Genetic Algorithm (GA) according to the change of human capability during the assembly process. In the experiments, we simulate the changes in human capability and adaptively generate the schedules for multiple robots and human. The results show that the advantages of the proposed solution over existing ones from multiple evaluation perspectives.

The rest of this paper is organized as follows. In Section II, we present the assembly scheduling problem and its model, the model of human capability and the idea of adaptive assembly scheduling that incorporates human capability. In Section III, GA is adopted to solve the adaptive scheduling

*This work was supported by the National Science Foundation under grant IIS-1845779

Yunyi Jia and Yi Chen are with the Department of Automotive Engineering, Clemson University, Greenville, SC 29607 USA (e-mail: yunyij@clemson.edu, yc4@clemson.edu).

Shaobo Zhang and Jun Zhang are with Chang'an University, Xi'an 710064 China and Department of Automotive Engineering, Clemson University, Greenville, SC 29607 USA. (e-mail: shbzhang@chd.edu.cn, zhangjun@chd.edu.cn).

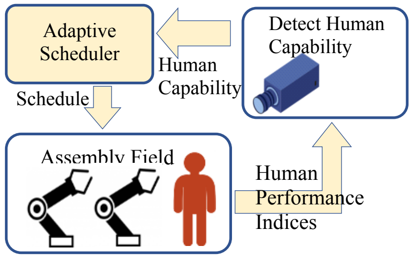


Figure 1. The framework of adaptive assembly scheduling

problem. In Section IV, simulations of numerical examples of a typical assembly scheduling problem are presented. Finally, conclusions are made in Section V.

II. Model of Human-multi-robot Assembly

A. Problem Statement and Representation

In the human-multi-robot assembling scenario, a single human collaborates with multiple robots to complete a series of assembly tasks, as shown in Fig. 1. In each task, one or more parts are assembled. The human and the robots can be modeled as heterogenous agents. Each task can contain multiple subtasks and each subtask must be conducted by at most one agent at each time instant. There may also be some temporal constraints between the subtasks but there is only one subtask that can be assigned to an agent in each time epoch. In addition, the human capability is subject to change and different from that of the robots. Given this fact, the objective of human-multi-robot assembly scheduling is to design an adaptive scheduler to find the best schedule so that all tasks can be completed in the shortest amount of time.

We will also provide a formal representation of the human-multi-robot assembly problem. Define m agents $a \in A = \{a_1, \cdots, a_m\}$, including capability-consistent robot agents and capability-varying human agent, to complete n tasks τ_i ($i = 1, \cdots, n$), which represents the set of steps required to assemble one part. The subtasks τ_i^j ($j = 1, \cdots, n_i$) is a stage of procedure of task τ_i . $d_{\tau_i^j}^a$ specifies the required processing time from agent a to subtask $\tau_i^j \in \tau$. $Seq_{<\tau_i^j, \tau_x^y>} = 1$ if subtask τ_x^y should be started after τ_i^j is accomplished. Otherwise, it is 0. The assembly scheduling problem is then to find a schedule with minimum time when all the tasks are completed.

B. Model of Assembly Scheduling

Define ${}^tA_{\tau_i^j}^a \in \{0,1\}$ as a binary decision variable for assigning an agent a to a subtask τ_i^j at the beginning of the time epoch $t \in [1, T]$. Define the estimated upper time limit as T . Define $B_{<\tau_i^j, \tau_x^y>} \in \{0,1\}$ as a binary decision variable, in which $B_{<\tau_i^j, \tau_x^y>} = 1$ if τ_i^j and τ_x^y are assigned to the same agent and τ_i^j succeeds τ_x^y . Otherwise, $B_{<\tau_i^j, \tau_x^y>} = 0$. Let $s_{\tau_i^j}$ and $f_{\tau_i^j} \in [1, T]$ represent the start time and the finish time of τ_i^j respectively. The problem of human-multi-robot assembly scheduling can then be formulated as a nonlinear mixed-integer program expressed by:

$$\min\left(\max_{\tau_i^j \in \tau} f_{\tau_i^j} \right) \tag{1}$$

subject to:

$$\sum_{a \in A} \sum_{t=1}^{T} {}^tA_{\tau_i^j}^a = 1, \quad \forall \tau_i^j \in \tau \tag{2}$$

$$\sum_{j=1}^{n_i} {}^tA_{\tau_i^j}^a \le 1, \quad \forall a \in A, \forall t \tag{3}$$

$$B_{\langle \tau_i^j, \tau_x^y \rangle} \cdot f_{\tau_i^j} < s_{\tau_x^y}, \forall \tau_i^j, \tau_x^y \in \tau \tag{4}$$

$$f_{\tau_i^j} \le T, \quad \forall \tau_i^j \in \tau \tag{5}$$

$$Seq_{\langle \tau_i^j, \tau_x^y \rangle} \cdot f_{\tau_i^j} < s_{\tau_x^y}, \forall \tau_i^j, \tau_x^y \in \tau \tag{6}$$

where the objective function in (1) represents the overall processing time of schedule that should be minimized. Equation (2) ensures that each subtask is assigned only one agent during each time epoch $[t, t+1)$, constraints given in equations (3) and (4) enforce that no agent is oversubscribed. Equation (5) ensures that the finish time of no subtask exceeds the upper limit T , and constraints of the processing order between subtasks are satisfied by (6).

In addition, $f_{\tau_i^j}, s_{\tau_i^j}$ and $d_{\tau_i^j}$, $\forall \tau_i^j \in \tau$ in (4)-(6) can be obtained using the below equations:

$$d_{\tau_i^j} = \sum_{a \in A} \sum_{t=1}^{T} \left(d_{\tau_i^j}^a \cdot {}^tA_{\tau_i^j}^a \right), \quad \forall \tau_i^j \in \tau \tag{7}$$

$${}^tb_{\tau_i^j} = \sum_{a \in A} {}^tA_{\tau_i^j}^a, \quad \forall \tau_i^j \in \tau, \forall t \tag{8}$$

$${}^tc_{\tau_i^j} = {}^tb_{\tau_i^j} + {}^{t-1}b_{\tau_i^j}, \quad \forall \tau_i^j \in \tau, t = 2, \cdots, T \tag{9}$$

$$s_{\tau_i^j} = T + 1 - \sum_{t=1}^{T} {}^tc_{\tau_i^j}, \quad \forall \tau_i^j \in \tau \tag{10}$$

$$f_{\tau_i^j} = s_{\tau_i^j} + d_{\tau_i^j}, \quad \forall \tau_i^j \in \tau \tag{11}$$

where (7) is used to get the processing time of τ_i^j , intermediate variables ${}^tb_{\tau_i^j}$ and ${}^tc_{\tau_i^j}$ in (8) and (9) are used to calculate the start time $s_{\tau_i^j}$ in (10), equation (11) can be used to derive finish time $f_{\tau_i^j}$ of τ_i^j .

III. Human Capability and Adaptive Scheduling

A. Model of Human Capability

In the above formulation, we assume that human performance and capability remain unchanged throughout the assembly process. But in realistic applications, human capabilities change over time due to many other factors such as fatigue, work intensity, absence of reasonable breaks or circadian rhythms. Therefore, in order to capture such capability changes in real-time with an effective and practical solution, we propose to model the human capability and its dynamics based on real-time measurable human performance indices. Define the human capability at time k as $c_h(k) \in [0,1]$ so that its dynamics can then be modeled as:

$$c_h(k+1) = \min\{1, \max\{0, c_h(k) + \alpha \cdot \Delta\bar{p}\}\} \quad (12)$$

$$\Delta\bar{p}(k) = \begin{cases} \mathrm{sign}\left(\Delta p_{\max}(k)\right) L_1, \text{ if } \left\|\Delta p_{\max}(k)\right\| = \max\left\{\left\|\Delta p_i(k)\right\|\right\} > L_1 \\ \dfrac{1}{N}\sum_i \Delta p_i(k), \text{ otherwise.} \end{cases} \quad (13)$$

where $P = \{p_i \mid i = 1, \cdots N\}$ represents the human performance vector which contains N performance indices, p_i represents a normalized index of one type of measurable human performance, $\Delta p_i(k)$ is the value change of p_i at time k, α is a coefficient to weigh the effects of performance change on capability change, and L_1 is a threshold for dramatic performance change. If a performance index changes dramatically (i.e., its value change is larger L_1), it indicates the human is stimulated to increase his/her capability or affected decrease his/her capability. In order to avoid the over-effect of the dramatic change, we set the maximal considerable change as L_1. In other cases, we just take the average index change into consideration for the human capability dynamics.

For the measurable human performance index design, we can consider indices which are important to the assembly performance and can also be easily captured using available sensing systems. For instance, we can design the indices as the average speed of the human hand movement, the stability (e.g., speed variance) of the human hand movement, the assembly accuracy of human operations, and the smoothness (e.g., hand trajectory variance) of human operations over a period of time. All these information can be captured using hand motion capture techniques such as vision-based motion capture systems[12], [13] or wearable hand devices such as smart gloves[14].

B. Adaptive Assembly Scheduling According to Human Capability

During the human-multi-robot assembly process, the capability of robots can be considered stable, while the human capability is usually varying. Existing approaches generate assembly scheduling solutions by mostly considering humans as agents with assumed and known capability that does not change. If such a solution is applied to realistic cases where human-capability-varies, it will lead to suboptimal performance as the optimization of the original scheduling will not hold anymore. Therefore, a real-time adaptive assembly scheduling method should be adopted.

Defining $a_1 \in A$ as the human agent, and given the variation in human capability, the processing time $d^{a_1}_{\tau^j_i}$ ($\forall \tau^j_i \in \tau$) will be no longer constant. Instead, they should be functions of human capabilities. Specifically, they should also change in real-time according to the variations of human capabilities. Based on this understanding, the process of the proposed real-time adaptive assembly scheduling algorithm is shown in Fig. 2. The algorithm will first generate an initial schedule by solving the assembly scheduling problem in Section II.B, and then during the process, periodically detect

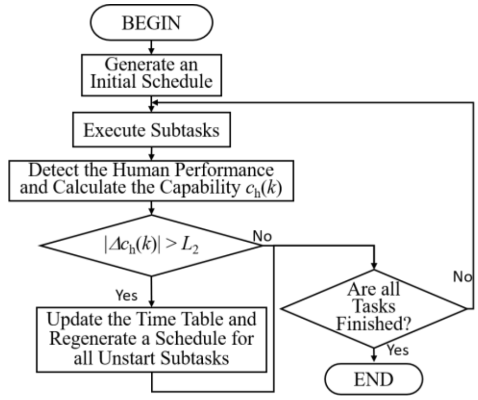


Figure 2. Flowchart of the adaptive assembly scheduling algorithm

human performance indices and compute the human capability index $c_h(k)$ using (12) and (13). If the change of $c_h(k)$ is higher than a predefined threshold L_2, it will update the processing time table for the human using

$$d^{a_1}_{\tau^j_i}(k) = d^{a_1}_{\tau^j_i}(0) / c_h(k) \quad (14)$$

which continuously increases or decreases the task processing time of human according to his/her capability variations. With these updated processing times, the assembly scheduling problem in Section II.B will be re-solved in real-time to generate the best schedule for all the remaining subtasks to be accomplished by multiple robots and human. With this real-time adaptation plan, the assembly can always be scheduled efficiently although the human capability may keep changing during the human-multi-robot assembly process. The approach to solving the assembly scheduling problem will be introduced in the following section.

C. Genetic Solution to Assembly Scheduling

The assembly scheduling problem is similar to the Job Shop Scheduling Problem (JSSP)[15], which is a nonlinear mixed-integer and NP-hard problem with high computational complexity. In order to solve it quickly with an acceptable solution, we propose to adopt the Genetic Algorithm (GA)[16], which is also often used to solve the JSSP. The GA proposed in this paper is described as Algorithm 1. The core components in the algorithm are explained in detail as below.

Algorithm 1: GA for Assembly Scheduling Problem

```
1: procedure GA for ASP
2:    initial the population P by encoding;
3:    for i ← 1 to the max generation number
4:        crossover P(i) at given probability p_c;
5:        mutation P(i) at given probability p_m;
6:        fitness calculation of each individual by decoding;
7:        select P(i+1) from P(i);
8:        record the best schedule;
9:    output best schedule;
```

(1) ***Encoding***: The encoding process is to express a solution to the problem as a chromosome. A chromosome of an individual is represented by two sections[17]: Agent Selection (AS) section and Task Sequence (TS) section. AS

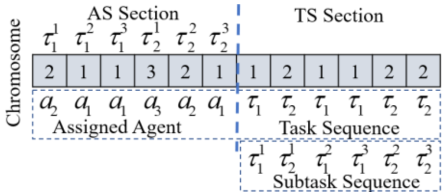


Figure 3. Encoding of a schedule for Genetic Algorithm

stores the index of the agent to which τ_i^j is assigned, and TS is the processing order of tasks. One example of encoding for three agents and two tasks with each task having three subtasks is shown in Fig. 3.

(2) ***Decoding:*** The decoding process is the mapping approach from chromosomes to solutions. It needs to maintain an idle timetable for each agent. After obtaining the subtask sequence from the TS section, it is traversed from the left to the right and the following steps repeated:

- If the subtask τ_i^j needs to be processed, get the assigned agent a_k from AS section;
- From the subtask sequence constrains, get the earliest time when τ_i^j can be processed;
- From the idle time table of a_k, search the earliest time slot which is long enough to execute τ_i^j;
- Set $s_{\tau_i^j}$, $f_{\tau_i^j}$, and update the idle time table of a_k.

The designed encoding and decoding operators can guarantee the obtained solutions satisfy the constraints (2) to (6). Crossover and mutation operators cannot destroy the feasibility of solutions, so the careful design is required.

(3) ***Crossover:*** For the AS section, randomly select 2 loci and swap the two parents' genes between them with a prescribed probability p_c; for the TS section, adopt the POX method[18], which can ensure that the number of per task index remains the same after crossover.

(4) ***Mutation:*** For the AS section, randomly select a locus, and change the gene on it to the agent index that has the shortest processing time for a given prescribed probability p_m; for the TS section, simply swap two random genes with p_m.

(5) ***Calculation of a Fitness***: Use (15) to calculate the fitness of a schedule. C is a predefined big integer.

$$F = \begin{cases} C, & \text{schedule is infeasible;} \\ C / \max\limits_{\tau_i^j \in \tau} f_{\tau_i^j}, & \text{otherwise.} \end{cases} \tag{15}$$

(6) ***Selection***: The roulette wheel method[16] is adopted to select feasible solutions.

(7) ***Initializing a Population:*** For each individual, the AS and TS sections are generated separately and then concatenated. While the AS section is generated by randomly generating a number series from 1 to m, the TS section is generated by randomly permutating the initial task index sequence $[1\cdots1\ 2\cdots2 \cdots\cdots\ n\cdots n]$ (every task index i repeats n_i times) to get the TS section.

IV. Human Capability and Adaptive Scheduling

A. Experimental Setup

In the first experiment, we assume that there are one human and two robots who work collaboratively to complete five tasks, i.e., to assemble five parts. For each task, the following subtasks are needed in sequence: pick up part, pick up a tool and assemble it. The task-subtask-robot combination is (5-3-2). In addition, the last subtasks of part 4 and part 5 must not begin until all other parts are successfully assembled. According to our formulation, there are five tasks, three agents (a_1 is the human agent, a_2 and a_3 are robot agents) and three subtasks for each task. In every task τ_i, subtask τ_i^j has to be processed before τ_i^{j+1}. τ_4^3 and τ_5^3 have to be processed before all the subtasks of τ_1 to τ_3.

We assume that the initial processing time table is shown in Table I. The numbers in the table represent time segments, where each segment is equal to 10 minutes. The processing time of each subtask by a human agent a_1 is under the assumption that the human has full capability and the number 999 means the subtask can not be assigned to the agent. During the assembly process, human performance indices are detected periodically to compute the human capability $c_h(k)$ using (12) and (13); If the change in $c_h(k)$ is higher than a predefined threshold L_2, the human processing time in the table is updated and the schedule for the subtasks that have not started yet will be regenerated.

In order to comprehensively evaluate the schedule in addition to total assembly time cost, we also define two metrics which are *task switch times* and *ratio of human work time* r_{human} as described below in detail.

- *Task switch times*: For an agent, if there is a sequence of subtasks scheduled for processing then the task switch time is the total switch times from one task to another. This indicates the fluency of the task accomplishment.
- *Ratio of human work time* r_{human}: For a schedule, this refers to the ratio of human work time over the total finish time. This indicates the minimization of human efforts during task accomplishment.

After the first experiment, three other experiments with different numbers of tasks and robots are carried out. The results show that the overall processing time and the r_{human} are decreased significantly.

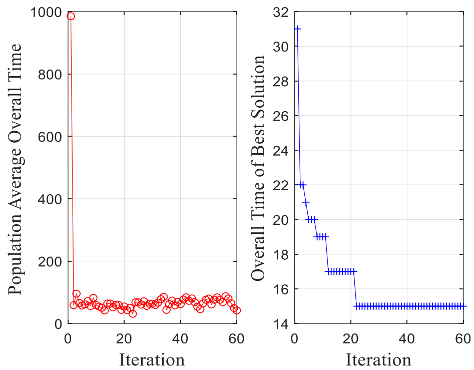
B. Experimental Results and Analysis

In the GA implement, the population size is set to 160 and the total generation number is 60. The possibility of crossover and mutation is 0.8 and 0.1 respectively. The ratio of human work time is $r_{human} = 0.8$. The GA algorithm is implemented using MATLAB. The total computation time required to generate a schedule is approximately 5 seconds on a laptop with i5-8350U processor and 8GB RAM.

The performance of GA is shown in Fig. 4. The initial population includes many infeasible solutions, so the average overall time is high. After several iterations, the feasible solution and the suboptimal solution occupy the majority of

TABLE I. THE INITIAL PROCESSING TIME TABLE

Agent	Subtask														
	τ_1^1	τ_1^2	τ_1^3	τ_2^1	τ_2^2	τ_2^3	τ_3^1	τ_3^2	τ_3^3	τ_4^1	τ_4^2	τ_4^3	τ_5^1	τ_5^2	τ_5^3
a_1	999	1	1	999	1	2	999	2	1	999	3	4	999	2	3
a_2	4	3	4	3	2	1	3	2	2	3	5	6	4	3	4
a_3	6	2	5	4	2	2	2	4	3	2	7	9	4	5	3



(a) (b)

Figure 4. GA performance

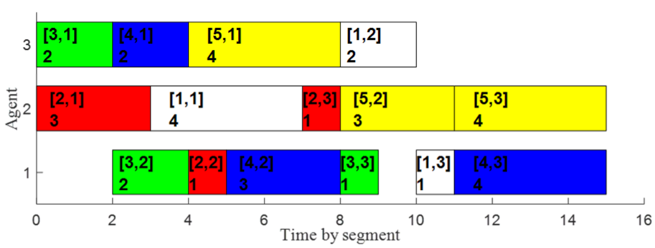


Figure 5. The initial schedule

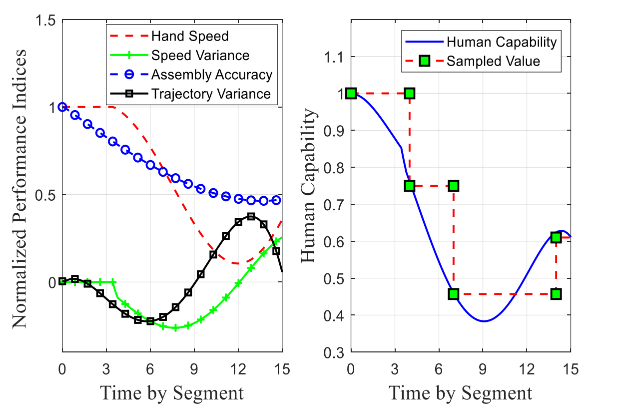
the population. Thus, the average overall time stabilizes around 37 segments. The best schedule is found in the 22nd iteration, which shows that the algorithm is stable and converges quickly.

Despite the changes in human capabilities, the initial solution of GA is shown in the form of a Gantt chart in Fig. 5. In that figure, each rectangle represents a subtask. The tuple $[i, j]$ in the rectangle represents the subtask τ_i^j, and the number below the tuple is the processing time segments for the assigned agent. The overall processing time of this initial schedule is 15 segments, the task switch time of the human agent is 5, and 6 subtasks are assigned to the human agent.

The changes of four normalized human performance indices and the calculated human capability $c_h(k)$ using (12)-(13) are shown in Fig. 6. Given this changing capability, the processing time of the human agent for specific subtask increases, resulting in an overall processing time of 24 segments (Fig. 7). Consequently, the actual ratio of human work time r_{human} becomes 0.92, which is higher than the specified value of 0.8. On the other hand, the actual ratios of the work times of the first and the second robots have dropped from 1.00 and 0.63 to 0.63 and 0.42 respectively. This results in a high workload for the human and long idle times for the robots.

Based on the simulated human capability change in Fig. 4, the adaptive assembly scheduling algorithm repeatedly generates the schedules for the subtasks, which are still not processed, at the beginning of time segments 4, 7 and 14 respectively (Fig. 8 - Fig. 10). In Fig. 8, by the time a change in human capability is detected in time segment 4, subtasks τ_2^1, τ_3^1, τ_3^2 and τ_4^1 have already finished, and τ_1^1 has already started. So the scheduler regenerates the schedule for the rest of the subtasks. Because τ_1^1 will be finished at the end of time segment 6, subtask τ_3^3 is started at 7 in the new schedule. The subtask τ_2^2 is completed, τ_4^2 and τ_5^1 are still not finished at time segment 7. The rest of the subtasks are scheduled again as shown in Fig. 9. Finally, the last two subtasks τ_4^3 and τ_5^3 are scheduled when human capability changes at time segment 14.

The final actual processing schedule is shown in Fig. 11. The overall completion time is 20, and the total human work time is 9 time segments. Therefore, $r_{human} = 0.45$. The number of subtasks assigned to the human has dropped from 6 to 3, the task switch time of the human has dropped from 5 to 2, and r_{human} is also significantly reduced. The idles times of the robots are much shorter than in the initial schedule. This allows the human to get reasonable idle time and have the opportunity to restore performance and capability.



(a) (b)

Figure 6. Human performance indices and capability

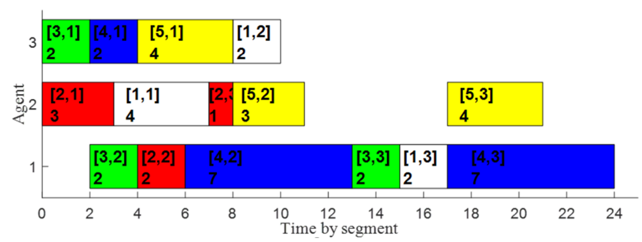


Figure 7. The actual processing induced by human capability change

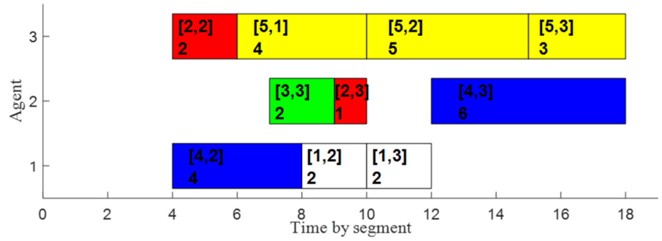


Figure 8. Regenerate schedule at time segment 4

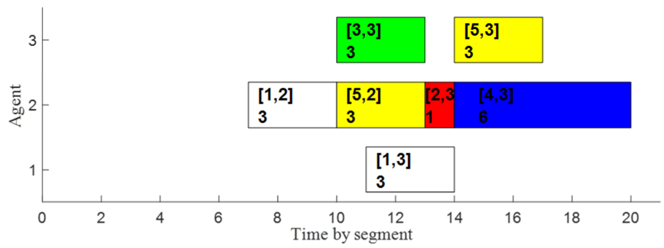


Figure 9. Regenerate schedule at time segment 7

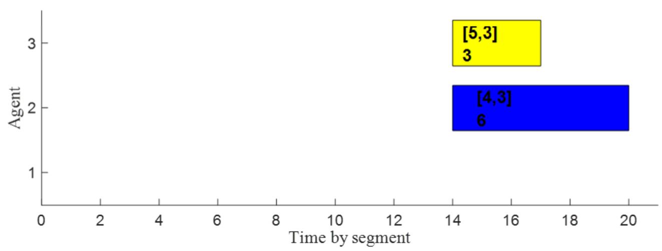


Figure 10. Regenerate schedule at time segment 14

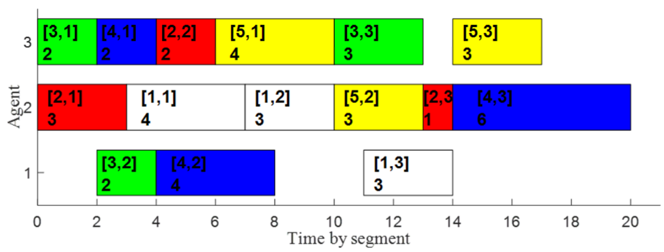


Figure 11. The schedule adapted to human capability

C. Statistical Evaluations

In addition to the first experiment (5-3-2 combination), we also designed experiments with more complicated task-subtask combinations (7-28-2, 7-35-3, and 10-50-4) as shown in Table II. The number of human and the human capability change trend remains the same as the first experiment. In addition, the last subtasks of the last three tasks cannot begin until all other parts are successfully assembled. The population size of GA is changed according to the number of subtasks in order to ensure optimum performance.

As shown in Table II, using the initial schedule, due to the changing of human capability, the actual overall time is on average 38.0% higher than the ideal case. The actual ratios of human work time r_{human} are increased to around 0.90. Using the adaptive scheduling algorithm proposed in this paper, the overall time is dropped on average by 18.8% when compared to the actual overall time of the initial schedule with changing human capability. On average, the number of subtasks

TABLE II. THE RESULT OF EXPERIMENTS WITH MORE TASKS

		Experiments		
		1	**2**	**3**
Number of Tasks		7	7	10
Number of Subtasks		28	35	50
Number of Robots		2	3	4
Population Size of GA		800	1000	12000
Initial Schedule with Full Human Capability	Overall Time	51	50	58
	Subtasks Assigned to Human	11	9	12
	Task Switch Times	8	7	9
	r_{human}	0.76	0.72	0.71
Initial Schedule with Changing Human Capability	Overall Time	67	68	85
	Subtasks Assigned to Human	11	9	12
	Task Switch Times	8	7	9
	r_{human}	0.90	0.91	0.89
Adaptive Schedule with Changing Human Capability	Overall Time	58	58	61
	Subtasks Assigned to Human	6	6	7
	Task Switch Times	5	4	5
	r_{human}	0.64	0.52	0.66

TABLE III. THE STATISTICS OF PERFORMANCE IMPROVEMENTS

	Decrease on Overall Time	**Decrease on r_{human}**	**Decrease on Human Task Switch Times**
Exp. 1	13.4%	28.9%	37.5%
Exp. 2	14.7%	42.8%	42.9%
Exp. 3	28.2%	25.8%	44.4%

assigned to humans dropped by 40.2%, and the actual ratios of human work time r_{human} is smaller. The statistics of schedule performance improvements also conclude in Table III. When compared to the initial schedule with changing human capabilities, the schedule generated by the adaptive scheduling algorithm can significantly reduce the overall processing time, the ratio of human work time and the task switch times.

V. CONCLUSIONS

Using the framework proposed in this paper, we can dynamically perform human-robot collaborative assembly scheduling according to the modeled human capability. The results of the simulated experiment show that the proposed algorithm can adjust the scheduling real-time to adapt to changing human capabilities. Adaptive scheduling can reduce the overall assembly task completion time, and allows human to get reasonable rest and improve assembly efficiency.

This paper focuses on the framework of adaptive assembly scheduling. The experiments use one set of simulated data of human capability. So we plan to detect the human performance in real-world assembly scenarios using motion capture systems, and test the proposed framework on more complex tasks. Furthermore, this paper only studies one-human and multi-robot collaboration assembly scheduling problem. Multi-human and multi-robot collaboration assembly scheduling problem adapting multi-human capability will be studied in the future work.

REFERENCES

[1] F. Chen, K. Sekiyama, F. Cannella, and T. Fukuda, "Optimal Subtask Allocation for Human and Robot Collaboration Within Hybrid Assembly System," *IEEE Trans. Automat. Sci. Eng.*, vol. 11, no. 4, pp. 1065–1075, Oct. 2014.

[2] M. C. Gombolay, R. J. Wilcox, and J. A. Shah, "Fast Scheduling of Robot Teams Performing Tasks With Temporospatial Constraints," *IEEE Trans. Robot.*, vol. 34, no. 1, pp. 220–239, Feb. 2018.

[3] B. Sadrfaridpour and Y. Wang, "Collaborative Assembly in Hybrid Manufacturing Cells: An Integrated Framework for Human–Robot Interaction," *IEEE Trans. Automat. Sci. Eng.*, vol. 15, no. 3, pp. 1178–1192, Jul. 2018.

[4] S. Pellegrinelli, A. Orlandini, N. Pedrocchi, A. Umbrico, and T. Tolio, "Motion planning and scheduling for human and industrial-robot collaboration," *CIRP Annals*, vol. 66, no. 1, pp. 1–4, 2017.

[5] Jeffrey R. Peters, Vaibhav Srivastava, Grant S. Taylor, Amit Surana, M. P. Eckstein, and Francesco Bullo, "Human Supervisory Control of Robotic Teams: Integrating Cognitive Modeling with Engineering Design," *IEEE Control Syst.*, vol. 35, no. 6, pp. 57–80, Dec. 2015.

[6] R. Müller, M. Vette, and O. Mailahn, "Process-oriented Task Assignment for Assembly Processes with Human-robot Interaction," *Procedia CIRP*, vol. 44, pp. 210–215, 2016.

[7] B. Sadrfaridpour, H. Saeidi, J. Burke, K. Madathil, and Y. Wang, "Modeling and Control of Trust in Human-Robot Collaborative Manufacturing," in *Robust Intelligence and Trust in Autonomous Systems*, R. Mittu, D. Sofge, A. Wagner, and W. F. Lawless, Eds. Boston, MA: Springer US, 2016, pp. 115–141.

[8] R. Wilcox, S. Nikolaidis, and J. Shah, "Optimization of Temporal Dynamics for Adaptive Human-Robot Interaction in Assembly Manufacturing," p. 8.

[9] Y. Jia, N. Xi, S. Liu, Y. Wang, X. Li, and S. Bi, "Quality of teleoperator adaptive control for telerobotic operations," *The International Journal of Robotics Research*, vol. 33, no. 14, pp. 1765–1781, Dec. 2014.

[10] N. Mitsunaga, C. Smith, T. Kanda, H. Ishiguro, and N. Hagita, "Adapting Robot Behavior for Human--Robot Interaction," *IEEE Trans. Robot.*, vol. 24, no. 4, pp. 911–916, Aug. 2008.

[11] T. Munzer, M. Toussaint, and M. Lopes, "Preference learning on the execution of collaborative human-robot tasks," in *2017 IEEE International Conference on Robotics and Automation (ICRA)*, Singapore, Singapore, 2017, pp. 879–885.

[12] T. B. Moeslund, A. Hilton, and V. Krüger, "A survey of advances in vision-based human motion capture and analysis," *Computer Vision and Image Understanding*, vol. 104, no. 2–3, pp. 90–126, Nov. 2006.

[13] M. Windolf, N. Götzen, and M. Morlock, "Systematic accuracy and precision analysis of video motion capturing systems—exemplified on the Vicon-460 system," *Journal of Biomechanics*, vol. 41, no. 12, pp. 2776–2780, Aug. 2008.

[14] N. Carbonaro, G. D. Mura, F. Lorussi, R. Paradiso, D. De Rossi, and A. Tognetti, "Exploiting Wearable Goniometer Technology for Motion Sensing Gloves," *IEEE J. Biomed. Health Inform.*, vol. 18, no. 6, pp. 1788–1795, Nov. 2014.

[15] H. Tamaki, T. Ono, H. Murao, and S. Kitamura, "Modeling and genetic solution of a class of flexible job shop scheduling problems," in *ETFA 2001. 8th International Conference on Emerging Technologies and Factory Automation. Proceedings (Cat. No.01TH8597)*, 2001, vol. 2, pp. 343–350 vol.2.

[16] J. H. Holland, *Adaptation in Natural and Artificial Systems: An Introductory Analysis with Applications to Biology, Control, and Artificial Intelligence*. MIT Press, 1992.

[17] G. Zhang, "Research on Methods for Flexible Job Shop Scheduling Problems," Huazhong University of Science and Technology, Wuhan, China, 2009.

[18] I. Kacem, "Genetic algorithm for the flexible job-shop scheduling problem," in *SMC'03 Conference Proceedings. 2003 IEEE International Conference on Systems, Man and Cybernetics. Conference Theme - System Security and Assurance (Cat. No.03CH37483)*, Washington, DC, USA, 2003, vol. 4, pp. 3464–3469.