# Resource Scheduling for Human-Machine Collaboration in Multiagent Systems

Yifeng Zhou, Kai Di, Zichen Dong, Yichuan Jiang

# Resource Scheduling for Human-Machine Collaboration in Multiagent Systems

Yifeng Zhou*, Kai Di, Zichen Dong, and Yichuan Jiang

School of Computer Science and Engineering, Southeast University, Nanjing 211189, China
*Correspondence:yfzhou@seu.edu.cn

**Abstract.** To solve the human-machine resource scheduling problem in multiagent systems, we represent the complex constraints among subtasks by a directed graph and model the transmission effect of uncertainty of task execution in the modeling of human-machine collaboration. We prove that the human-machine resource scheduling problem is NP-hard and propose a heuristic scheduling algorithm to solve the problem. The algorithm determines the task priority based on cumulative delay risk estimation and time period division, so as to produce the resource scheduling scheme. Through experiments employing PSPLIB dataset, we validate the performance of the proposed heuristic algorithm by comparing with an optimal solution and the algorithm allocating resources based on the waiting time of tasks.

**Keywords:** Multiagent system · Resource scheduling · Human-machine collaboration.

## 1 Introduction

Resource scheduling is one of the key problems of agent collaboration in multi-agent systems [3, 6]. It mainly focuses on how to effectively schedule resources to maximize the utilization of resources, so as to improve the task completion rate and optimize the system utility [5]. In the human-machine collaboration scenario, i.e., the set of agents is composed of resources of human and machine, there are complex constraints among subtasks of a task, and it is difficult to estimate the execution time when human resources execute the task; these problems bring challenges to the resource scheduling in human-machine collaboration in multiagent systems [13, 14].

Previous related work pays little attention to the transmission effect of uncertainty when there are complex constraints among subtasks on task execution in human-machine collaboration scenarios in multi-agent systems [2–4, 8, 9, 12, 13]. To solve the problem, we represent the complex constraint relationship between subtasks by a directed graph and model the transmission effect of uncertainty of task execution in the modeling of human-machine cooperative resource scheduling problem in this paper. We further prove that the problem is NP-hard, and propose a heuristic scheduling algorithm that determines the task priority based

on the cumulative delay risk estimation and time period division of tasks and then generates the resource scheduling scheme.

Through experiments employing PSPLIB dataset [1], we validate the performance of the proposed heuristic algorithm from four aspects: the task completion rate, the algorithm running time, the average waiting time, and the resource utilization rate.

## 2 Related Work

### 2.1 Resource-Constrained Project Scheduling

The resource-constrained project scheduling problem (RCPSP) [4], which is one of the most important problems in operations research and management science, focuses on the allocation of tasks with the goal of minimizing the execution time of the task set under the dependency constraints of tasks and resources. Tirkolaee et al. [12] consider a nonlinear programming modeling of a multi-objective multimodal RCPSP and propose the efficient Pareto-based metaheuristics to solve the problem, which can maximize the net present value and minimize the completion time. Lin et al. [9] formalize the multi-skill resource-constrained project scheduling problem (MS-RCPSP) considering the resources like manpower or multipurpose machine, and introduce a genetic programming hyper-heuristic algorithm to address the problem, aiming at minimizing the makespan of the project. Although these works perform well in the scheduling problem with task constraints, the impact of involvement of human resources on task execution, e.g., the uncertainty of task execution, has not been fully considered in these works.

### 2.2 Task Allocation for Human-Machine Collaboration

Task allocation in multiagent systems often focuses on designing efficient task allocation algorithms to optimize the utility of agent collaboration. Wu et al. [13] use Markov decision process (MDP) to study how machine performance affects the human-machine trust and formalize the optimization problem considering the effect of human-machine trust where an optimal task allocation policy is proposed to minimize the average cost of tasks. Cai et al. [3] model the relationship of human-machine detection success rate in a human-machine collaborative task, and propose a linear program-based efficient near-optimal solution, which can significantly improve the task performance. Compared with these works, this paper pays more attention to the complex constraints of tasks and the transmission effect of uncertainty of task execution in resource scheduling for human-machine collaboration.

## 3 Problem Formulation

In this section, the human-machine resource scheduling problem in multiagent systems is formally defined.

**Resource model for human-machine collaboration:** the resources in the human-machine collaboration in multiagent systems can be broadly classified into two categories $R = R_h \cup R_c$, i.e., the human and machine resources, where each type of resources in $R$ can be represented by $R_i$.

**Task model for human-machine collaboration:** the human-machine collaboration task generally consists of multiple segments, each requiring different types of human and machine resources. Each task in the task set $T$ can be represented by a tuple, $T_i = \{V_i, E_i, A_i, W_i, D_i, R_i\}, i \in \{1, .., |T|\}$. $V_i$ and $E_i$ denotes the set of vertices and edges of subtasks of task $T_i$; and all subtasks form a directed graph $G =< V, E >, \bigcup_{i=1}^{|T|} V_i = V, , \bigcup_{i=1}^{|T|} E_i = E$. $A_i$ and $D_i$ denote the arrival time and deadline of task $T_i$. $W_i = \{w_{i1}, \ldots, w_{ij}\}, j \in \{1, .., |T_i|\}, w_{ij} \in \mathbb{R}^+$ denotes the execution time matrix of subtasks $v_{ij}$. $R_i = \{r_{i1}, \ldots, r_{ij}\}, j \in \{1, .., |T_i|\}$ denotes the matrix of resource types required by the subtask $v_{ij}$.

Assuming that the execution time of $v_{ij}$, $w_{ij}$, obeys the Gaussian distribution of $X \sim N(\mu, \sigma^2)$. The larger $\sigma$ will lead to the higher uncertainty of completion time and the higher risk of postponement. Thus, the execution time of $v_{ij}$ is estimated as

$$\widetilde{w}_{ij} = \mu(w_{ij}) + \eta * \sigma(w_{ij}) \tag{1}$$

where $\eta$ is a constant in [0,3] (when $\eta = 3$, it can be approximated as an upper bound for the estimation).

**Optimization objective:** the task $T_i$ is completed if all its subtasks $v_{ij}$ are completed before the deadline $D_i$. The allocation scheduling policy can be denoted by $\Pi < v_{ij}, r_{ij}, t_{ij} >$, which means the resource $r_{ij}$ in $R$ is allocated to $v_{ij}$ at time $t_{ij}$. Thus for any task $T_i$, its gain can be expressed as

$$U_i = \begin{cases} 1, \forall t_{ij} + \widetilde{w}_{ij} \leq D_i \\ 0, \exists t_{ij} + \widetilde{w}_{ij} > D_i \end{cases}, v_{ij} \in T_i \tag{2}$$

The final completion time of the task $T_i$ is represented by $\tau_i$. Then by adjusting the allocation scheduling policy, the total benefit that can be obtained is expressed as

$$U = \sum_{i \in T} I\left(D_i - \tau_i\right) \tag{3}$$

where $I(\cdot)$ is the indicator function that takes the value of 1 when the variable is greater than or equal to 0, and 0 otherwise

**Human-Machine Resource Scheduling Problem (HMRSP):** Given a set of tasks $T$ , a set of resources $R$ , design an allocation strategy $\Pi < v, r, t >$ that maximizes the number of tasks to be completed by the deadline. For each $v_{ij} \in V$, at any moment $t \in [0, \tau_i]$, the binary decision variable $x_{ijt}^k$ indicates whether $v_{ij}$ starts to occupy resources $R_k$ at $t$; and for each task $T_i$, the continuous decision variable $\tau_i \in \mathbb{N}$ indicates the final completion time of task $T_i$. The human-machine resource scheduling problem (HMRSP) can be modeled as:

$$\max \frac{1}{|T|}\sum_{i=1}^{|T|} I\left(D_i-\tau_i\right) \tag{4}$$

$$\text{s.t.} \quad \sum_{t=0}^{\tau_i}\sum_{k=1}^{|R|} x_{ijt}^{k} t \geq A_i, v_{ij}\in V \tag{5}$$

$$\sum_{t=0}^{\tau_i}\sum_{k=1}^{|R|} x_{ijt}^{k}(t+\widetilde{w}_{ij}) \leq \tau_i, v_{ij}\in V, \tau_i \in \mathbb{N} \tag{6}$$

$$\sum_{t=0}^{\tau_i}\sum_{k=1}^{|R|} x_{ijt}^{k} = 1, v_{ij}\in V \tag{7}$$

$$\sum_{t=0}^{\tau_i} x_{ijt}^{r_{ij}} = 1, v_{ij}\in V \tag{8}$$

$$\sum_{t=0}^{\tau_i}\sum_{h=1}^{|R|} x_{ijt}^{h}\left(t+\widetilde{w}_{ij}\right) \leq \sum_{t=0}^{\tau_i}\sum_{h=1}^{|R|} x_{ikt}^{h} t \tag{9}$$

$$v_{ij}, v_{ik}\in V,\ j\neq k, (j,k)\in E_i \tag{10}$$

$$\sum_{i=1}^{|T|}\sum_{j=1}^{|V_i|}\sum_{t=\max\left(0,t'-w_{ij}+1\right)}^{t'} x_{ijt}^{k} \leq N_k \tag{11}$$

$$N_k = |R_k|, t'\in\{0,\ldots,\tau_i\} \tag{12}$$

$$x_{ijt}^{k}\in\{0,1\}, v_{ij}\in V, R_k\in R, t\in\{0,\ldots,\tau_i\} \tag{13}$$

**Theorem 1.** *The HMRSP problem is NP-hard when* $|R|\geq 3$.

*Proof.* First, the HMRSP problem is NP, since it is easy to construct a task assignment and scheduling scheme and then verify the maximum completion time of its tasks in polynomial time; and then the NP-hardness of the problem can be proved via reducing to the Subset Product Problem (SP) [7]. The SP problem can be described in the following form: given a set $A=\{1,2,\ldots,v\}$ and $a_i\in\mathbb{N}^+, i\in A$, does there exist a subset $A'\subseteq A$ such that $\prod_{i\in A'} a_i = \prod_{i\in A\setminus A'} a_i = \sqrt{\prod_{i\in A} a_i} \equiv B$?

To demonstrate SP $\leq_p$ HMRSP, an instance of HMRSP is constructed. Let $N=v+3, M=3, \alpha_{i1}=\alpha_i, \alpha_{i2}=\beta_i, \alpha_{i3}=\gamma_i$, all elements in the instance can be presented as follows: $\alpha_i=0,\quad \beta_i=\alpha_i-1,\quad \gamma_i=0$; $\alpha_{v+1}=0,\quad \beta_{v+1}=1,\quad \gamma_{v+1}=B+1$; $\alpha_{v+2}=2B+1,\quad \beta_{v+2}=\frac{1}{B+1},\quad \gamma_{v+2}=\frac{B^2+2B+1}{B+2}$; $\alpha_{v+3}=\frac{B^2+2B+1}{B+1},\quad \beta_{v+3}=\frac{1}{B^2+3B+2},\quad \gamma_{v+3}=0$; $D=B^2+3B+3,\quad t_0=1/2$.

Clearly, the above construction can be done in polynomial time.

– If for an instance of SP,$I_{\mathrm{SP}}$, there exists a subset $A'$ satisfying $\prod_{i\in A'} a_i = \prod_{i\in A\setminus A'} a_i = \sqrt{\prod_{i\in A} a_i}$ then for instances of the HMRSP problem, there

exists a scheduling $q$ satisfying $\tau_i \leq D$. Given a solution of $I_{\mathrm{SP}}$ to the task $v+1, v+2, v+3$ build a scheduler in the order of $q'$. The task completion time $\tau_i$ is denoted as $B^2+3B+3$, leaving two gaps for resource 2: the first is to reserve $B$ workload at moment 1 and the second at moment $B+2$ The second is to set aside a gap of length $B^2+2B$, the workload is $B+2$. Scheduling tasks with index $i \in A'$ that fill the first time gap (e.g. after task $v+1$ ), and the remaining tasks fill the second gap (after tas $v+2$ after the task). (The total load of the first task subset is then $t\prod_{i\in A'}(1+\beta_i) = 1\prod_{i\in A'} a_i = B$. The total load of the second task subset is $t\prod_{i\in A\setminus A'}(1+\beta_i) = (B+2)\prod_{i\in A\setminus A'} a_i = B^2+2B$)

– If there exists a completion time for an instance of HMRSP $\tau_i \leq D = B^2+3B+3$ of the scheduling $q$ then for instances $I_{\mathrm{SP}}$ there exists a subset $A' \subseteq A$ satisfying $\prod_{i\in A'} a_i = \prod_{i\in A\setminus A'} a_i = \sqrt{\prod_{i\in A} a_i} = B$. It is easy to verify that if such a scheduling $q$ exists, then it must contain a sequence of tasks in the order $v+1, v+2, v+3$ sequence (any other sequence consisting of three tasks $\tau_i > D$). As mentioned above, this scheduling plan leaves two gaps for resource 2. Therefore, the set of tasks $A$ must be arranged in the gaps. Then we can have that there must exist a subset $A' \subseteq A$ satisfying $\prod_{i\in A'}(1+\beta_i) \leq B, (B+2)\prod_{i\in A\setminus A'}(1+\beta_i) \leq B^2+2B$, i.e., $\prod_{i\in A'}(1+\beta_i) \leq B$, $\prod_{i\in A\setminus A'}(1+\beta_i) \leq B$. Since $\prod_{i\in A'}(1+\beta_i) = \prod_{i\in A} a_i = B^2$, it can be concluded that $\prod_{i\in A'}(1+\beta_i) = \prod_{i\in A'} a_i = B, \prod_{i\in A\setminus A'}(1+\beta_i) = \prod_{i\in A\setminus A'} a_i = B$.

## 4 Algorithm Design

### 4.1 Heuristic Scheduling Algorithm for Human-machine Collaboration

We prove the HMRSP problem is NP-hard in Section 3 and thus an optimal solution cannot be obtained in polynomial time. In the scheduling process, the priority of tasks determines the order of tasks to obtain resources, which is the key factor affecting the completion rate of tasks. Hence, we consider determining the priority of executable tasks by calculating the cumulative delay risk and dividing the execution time through task topology.

On the one hand, this paper considers a measure of precursor task execution uncertainty in the task priority calculation. Considering that the cumulative delay risk is a common metric in project scheduling robustness studies [10], the subtask $v_{mn}$'s risk weight is defined as

$$\delta_{mn} = \frac{\sigma(w_{mn})}{\sum_{v_{ij}\in V_i}\sigma(w_{ij})} \tag{14}$$

Since the delay in the completion of any precursor task will directly affect the actual start time and completion time of the current task, i.e., the delay risk of the current task will increase due to any precursor task delay. Thus, the cumulative delay risk of $v_{mn}$ is:

$$\varphi_{mn} = \delta_{mn} + max\{\delta_{mi}, (i,n) \in E_m, v_{mi}, v_{mn} \in V_m\} \tag{15}$$

The cumulative delay risk value reflects the extent that the task could be affected by the uncertainty of task execution time; and it has a cumulative feature in the task topology.

On the other hand, since tasks have corresponding deadlines, it is necessary to specify the amount of execution time that can be set for each subtask, i.e., the total execution time needs to be divided according to the task topology. In this paper, we consider the division in proportion to the delayable time. Since the critical path affects the specific execution time of the task, and the deadline is a constraint on all the subtasks, the partial critical path (PCP) division method [2] is employed in this paper. Algorithm 1 is the recursive process of solving partial critical paths.

**Algorithm 1:** *PartialCriticalPath(PCP)*

**Input:** Task set $T$, node $v_{ij}$
**Output:** Partial set of critical paths
**Initialize:** $p \leftarrow v_{ij}$, $v' \leftarrow v_{ij}$, $v'$ marked as assigned
**while** $v^{'}$ *exits unassigned predecessor node* $v_p$, **do**
  $p \leftarrow v_p + p$
  $v_p$ marked as assigned
  $v' \leftarrow v' \cup v_p$
$P \leftarrow P + p$
$v' \leftarrow v' \cup p$'s end node
**while** $v^{'} != null$ **do**
  **foreach** $v_p \in pred\left(v^{'}\right)$ *in* $G$ **do**
    **if** $v_p$ *unassigned* **then**
      $P' \leftarrow PartialCriticalPath(T^{'}, v_{ij})$
      $P \leftarrow P \cup P'$
  $v^{'} \leftarrow$ pred $\left(v^{'}\right)$ in $p$
**return** $P$

After obtaining partial critical paths, the total execution time is then needed to be divided. For the topology $G$ composed of a set of subtasks, we firstly need to calculate the earliest start time, the latest start time, and the earliest completion time for $v_{ij}$.

$v_{ij}$'s Earliest Start Time is calculated as

$$T_{ij}^{\mathrm{ES}} = \begin{cases} A_i, & pred\left(v_{ij}\right) = \varnothing \\ \max_{v_{ip} \in pred(v_{ij})} \left\{T_{ip}^{\mathrm{ES}} + \widetilde{w}_{ip}\right\}, & o.w. \end{cases}, v_{ij} \in V \tag{16}$$

$v_{ij}$'s Earliest Completion Time (ECT) is calculated as

$$T_{ij}^{\mathrm{EC}} = T_{ij}^{\mathrm{ES}} + \widetilde{w}_{ij}, v_{ij} \in V \tag{17}$$

$v_{ij}$'s Latest Completion Time (LCT) is calculated as

$$T_{ij}^{\mathrm{LC}} = \begin{cases} D_i, & succ(v_{ij}) = \varnothing \\ \min_{v_{iq} \in succ(v_{ij})} \left\{ T_{iq}^{\mathrm{LC}} - \widetilde{w}_{iq} \right\}, & o.w. \end{cases}, v_{ij} \in V \tag{18}$$

Then, the total execution time is divided by the task topology diagram structure according to the set of obtained partial critical paths (as shown in Algorithm 2). Then the deadline for $v_{ij}$ is calculated as

$$d_{ij} = T_{i1}^{\mathrm{ES}} + \frac{T_{ij}^{\mathrm{EC}} - T_{i1}^{\mathrm{ES}}}{T_{ik}^{\mathrm{EC}} - T_{i1}^{\mathrm{ES}}} \times (T_{ik}^{\mathrm{LC}} - T_{i1}^{\mathrm{ES}}), v_{ij} \in V \tag{19}$$

Algorithm 2 focuses on the division of the total execution time for the nodes on the derived partial critical path.

**Algorithm 2:** *TaskProcessing*

**Input:** Task set $T_i$
**Output:** Executable tasks set $T_i^e$, blocking tasks set $T_i^b$
**Initialize:** $T_i^e \leftarrow \varnothing$
**foreach** $v_{ij} \in V_i$ **do**
    Calculate the earliest start time $T_{ij}^{\mathrm{ES}}$ according to equation (16)
    Calculate the earliest completion time $T_{ij}^{\mathrm{EC}}$ according to equation (17)
    Calculate the latest completion time $T_{ij}^{\mathrm{LC}}$ according to equation (18)
$P \leftarrow \mathrm{PCP}(T_i, v_{\mathrm{i\ exit}})$
**foreach** $p \in P$ **do**
    **foreach** $v_{ij} \in p$ **do**
        Calculate subtask $v_{ij}$'s deadline $d_{ij}$ according to equation (19)
$T_i^e \leftarrow v_{ij}, \forall v_{ij} \in V_i, pred(v_{ij}) = \varnothing$
$T_i^b \leftarrow T_i - T_i^e$
**return** $T_i^e, T_i^b$

Through Algorithm 2, the execution time of the task can be divided to each subtask by considering partial critical path division method. However, it ignores the cumulative impact of the precursor tasks which may cause uncertain execution time; and the partial critical path division process leads to the fragmentation of the inherent backward and forward dependencies in the task topology graph. Thus we consider the following three aspects: the cumulative delay risk, the deadline, and the successor tasks of the task for the specific resource allocation. The task priority can be defined as:

$$p_{ij} = \alpha \varphi_{ij} + \beta d_{ij} + \gamma N_{ij} \tag{20}$$

where $N_{ij}$ denotes the number of successor waiting vertices in the task topology , and $\alpha \in (0,1], \beta \in [-1,0), \gamma \in (0,1]$ is the weight factor. By combining the factors of the cumulative delay risk, the deadline and the number of successor tasks, the comprehensive index to measure the resource allocation priority of each task can reflect the impact of precursor task execution uncertainty impact, the deadline urgency and the inherent topological properties of successor tasks.

Algorithm 3 starts with a priority calculation for each executable task (line 2), followed by dynamic maintenance through the priority queue. It takes out the task with the highest priority at the top of the priority queue each time for resource allocation and assigns the task with the lowest expected completion time among all resources (lines 5-7), and then generates a scheduling scheme and updating the resource occupation time (lines 8-9) and finally update the priority queue (line 10).

**Algorithm 3:** *Heuristic*

**Input:** Executable tasks set $T^e$, blocking tasks sets $T^b$, resources set $R$
**Output:** Distribution schemet $\Pi < v, r, t >$

1 **Initialize:** set resource availability time $T_R$ to 0; $\Pi \leftarrow \varnothing$
2 Calculate the priority for each $p_{ij}$ according to (20)
3 $P \leftarrow \mathrm{PCP}(T_i, v_{\mathrm{i\ exit}})$
4 Add $v_{ij} \in T^e$ to the priority queue in ascending order of priority PQ $< p_{ij}, v_{ij} >$
5 **while** *PQ is not empty* **do**
6 $v_{ij} \leftarrow$ PQ.peek
7 PQ delete $v_{ij}$
8 $r_{\min} \leftarrow \mathrm{argmin}_{h \in R_{r_{ij}}} T_{R_h}$
9 $\Pi \leftarrow \Pi + < v_{ij}, r_{\min}, T_{R_h} >$
10 $T_{R_h} \leftarrow T_{R_h} + w_{ij}$
11 Move successor task of $T^b$ of $v_{ij}$ that can perform the assignment to $T^e$ and add it to the PQ
12 **return** $\Pi$

## 5 Experimental Validation

### 5.1 Experimental Settings

The data used in the experiments are obtained from the PSPLIB dataset [1], which originated from the resource-constrained project scheduling problem [4] [12] [9]. The experiments are solved by CPLEX for integer programming. The subtask dependencies in the experiments are generated through the network complexity parameters [11]. Each task consists of an average of eight subtasks with backward and forward dependencies, and the mean execution time obeys a

normal distribution and the standard deviation being a multiple of the mean in the interval [0.1,0.6]. The deadline of the task is the minimum completion time calculated by PSPLIB based on the average execution time of the task multiplied by a factor with an offset of 40 units of time. Each subtask is set to occupy one unit of resource. The experimental performance of the algorithm are tested in four aspects: the task completion rate, the running time of the algorithm, the average waiting time of the tasks, and the resource utilization rate; and there are two methods for comparison: 1) OPT, the optimal solution (solved by CPLEX), and 2) WT, which allocates resources based on the waiting time of tasks [5, 6]. The heuristic scheduling algorithm proposed in this paper is denoted by HC.

### 5.2 Experimental Results

To validate the performance of the proposed HC algorithm, the task completion rate is tested. Fig. 1 presents the results of task completion rate obtained by HC, OPT and WT.

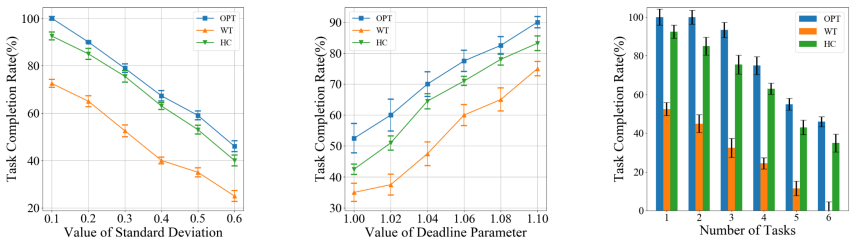
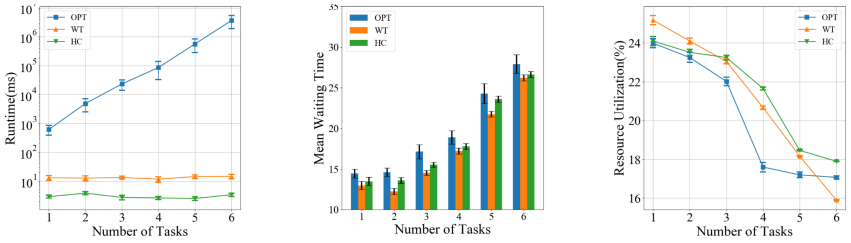



**Fig. 1.** Experimental results of task completion rate employing HC, OPT and WT algorithms.

From the results, OPT and HC perform much better than WT, and the performance of HC is close to the optimal solution. From Fig. 1(a) and 1(c), with the increase of the standard deviation of task execution time and the number of tasks, the task completion rate of different algorithms are decreasing. The potential reason is that the actual execution time of tasks fluctuates more and the delay risk of each task keeps increasing, thus leading to an increase in the cumulative delay risk of tasks and a decreasing task completion rate with a higher total number of tasks; with the number of tasks increases, the number of tasks waiting for the same resource increases, leading to an increased probability of tasks being delayed and a decreasing task completion rate. From Fig. 1(b), with the increase of task deadline, the task completion rate of different algorithms increase. The potential reason is that with the task deadline increases, the execution time for each subtask increase, the time available to deal with task delay increases, thus more tasks can be completed before the deadline, and the task completion rate keeps increasing.

In order to investigate the key features of the proposed HC algorithm, the algorithm runtime, the average waiting time and the resource utilization are also tested; Fig. 2 presents the experimental results.




(a) (b) (c)

**Fig. 2.** Experimental results of the algorithm runtime, the average waiting time and the resource utilization

Fig. 2(a) presents the tests results of the impact of the number of tasks on algorithm runtime. with the increase of the number of tasks, the OPT running time increases continuously and is much larger than HC and WT algorithms; and the running time of HC algorithm and WT algorithm are within 10ms. The potential reason is that as the number of tasks increases, the OPT state space increases exponentially; the HC and WT algorithms have much lower time complexity, thus the efficiency of HC and WT is much higher than OPT. Fig. 2(b) presents the tests results of the average waiting time. with the increase of the number of tasks, the average waiting time of the algorithms all show an increasing trend, and the average waiting time of OPT is larger than that of HC algorithm, while the average waiting time of HC algorithm is close to that of WT algorithm. The potential reason is that as the number of tasks increases, the number of tasks waiting for the same resource increases with the same amount of resources, thus leading to the increase of the average task waiting time; the WT algorithm gives priority to assign tasks with longer waiting time, the HC algorithm takes into account the cumulative delay risk and the deadline of tasks, and the OPT algorithm gives priority to task completion rate. Fig. 2(c) presents the tests results of resource utilization which indicates the ratio of resource utilization time to total task duration.with the increase of the number of tasks, the resource utilization of different algorithms all show a decreasing trend, and the resource utilization of OPT algorithm is smaller than that of WT algorithm and HC algorithm; and the resource utilization of HC algorithm gradually exceeds that of WT algorithm when the amount of tasks is larger than 3. The potential reason is that as the number of tasks increases, the number of tasks waiting for the same resource increases with the same amount of resources; thus it leads to the higher delay risk, the longer waiting time, and the lower resource utilization of tasks. The algorithm HC in this paper takes into account the cumulative delay risk and task deadline, so it allocate resources to tasks

at critical vertices with higher priority, thus it can achieve a shorter total task execution time and a higher resource utilization.

## 6 Conclusions

To solve the human-machine resource scheduling problem in multiagent systems, we represent the complex constraint relationship between subtasks by a directed graph and model the transmission effect of uncertainty of task execution in the modeling of human-machine cooperative resource scheduling problem in this paper. Since we prove the problem is NP-hard, we propose a heuristic scheduling algorithm to solve the problem. The heuristic algorithm determines the task priority based on cumulative delay risk estimation and time period division, so as to produce the resource scheduling scheme. Through experiments employing PSPLIB dataset, we investigate the performance of the proposed heuristic algorithm from four aspects: the task completion rate, the algorithm running time, the average waiting time, and the resource utilization rate, and by comparing with OPT and WT algorithms, the proposed algorithm performs close to the OPT algorithm and have a much lower running time which can be applied into large scale collaboration scenarios.

## Acknowledgements

This work was supported by the National Key Research and Development Project of China (2019YFB1405000), the National Natural Science Foundation of China (No.61807008, 61806053, 61932007, 62076060, and 61703097); and the Natural Science Foundation of Jiangsu Province of China (BK20180369, BK20180356, BK20201394, and BK20171363)

## References

1. Psplib. http://www.om-db.wi.tum.de/psplib/main.html
2. Abrishami, S., Naghibzadeh, M., Epema, D.H.: Deadline-constrained workflow scheduling algorithms for infrastructure as a service clouds. Future Generation Computer Systems **29**(1), 158–169 (2013)
3. Cai, H., Mostofi, Y.: Human–robot collaborative site inspection under resource constraints. IEEE Transactions on Robotics **35**(1), 200–215 (2018)
4. Habibi, F., Barzinpour, F., Sadjadi, S.: Resource-constrained project scheduling problem: review of past and recent developments. Journal of project management **3**(2), 55–88 (2018)
5. Jiang, Y., Zhou, Y., Li, Y.: Reliable task allocation with load balancing in multiplex networks. ACM Transactions on Autonomous and Adaptive Systems (TAAS) **10**(1), 1–32 (2015)
6. Jiang, Y., Zhou, Y., Wang, W.: Task allocation for undependable multiagent systems in social networks. IEEE Transactions on Parallel and Distributed Systems **24**(8), 1671–1681 (2012)

7. Johnson, D.S.: The np-completeness column: an ongoing guide. Journal of Algorithms **6**(3), 434–451 (1985)
8. Lin, B., Guo, W., Xiong, N., Chen, G., Vasilakos, A.V., Zhang, H.: A pretreatment workflow scheduling approach for big data applications in multicloud environments. IEEE Transactions on Network and Service Management **13**(3), 581–594 (2016)
9. Lin, J., Zhu, L., Gao, K.: A genetic programming hyper-heuristic approach for the multi-skill resource constrained project scheduling problem. Expert Systems with Applications **140**, 112915 (2020)
10. Rezaei, F., Najafi, A.A., Ramezanian, R.: Mean-conditional value at risk model for the stochastic project scheduling problem. Computers & Industrial Engineering **142**, 106356 (2020)
11. Sprecher, A., Kolisch, R.: Psplib—a project scheduling problem library. Eur. J. Oper. Res **96**, 205–216 (1996)
12. Tirkolaee, E.B., Goli, A., Hematian, M., Sangaiah, A.K., Han, T.: Multi-objective multi-mode resource constrained project scheduling problem using pareto-based algorithms. Computing **101**(6), 547–570 (2019)
13. Wu, B., Hu, B., Lin, H.: Toward efficient manufacturing systems: A trust based human robot collaboration. In: 2017 American Control Conference (ACC). pp. 1536–1541. IEEE (2017)
14. Zhou, Y., Di, K., Xing, H.: Hybrid multiagent collaboration for time-critical tasks: A mathematical model and heuristic approach. Algorithms **14**(11), 327 (2021)