

Article

An ANN-Based Approach for Real-Time Scheduling in Cloud Manufacturing

Shengkai Chen ^{1,*}, Shuliang Fang ^{1,2,*} and Renzhong Tang ¹¹ School of Mechanical Engineering, Zhejiang University, Hangzhou 310027, China; tangrz@zju.edu.cn² The State Key Laboratory of Fluid Power Transmission and Control, Zhejiang University, Hangzhou 310027, China

* Correspondence: 42@zju.edu.cn (S.C.); me_fangsl@zju.edu.cn (S.F.)

Received: 26 February 2020; Accepted: 31 March 2020; Published: 5 April 2020



Abstract: The cloud manufacturing platform needs to allocate the endlessly emerging tasks to the resources scattered in different places for processing. However, this real-time scheduling problem in the cloud environment is more complicated than that in a traditional workshop because constraints, such as type matching, task precedence, resource occupation, and logistics duration, need to be met, and the internal manufacturing plan of providers must also be considered. Since the platform aggregates massive manufacturing resources to serve large-scale manufacturing tasks, the space of feasible solutions is huge, resulting in many conventional search algorithms no longer being applicable. In this paper, we considered resource allocation as the key procedure for real-time scheduling, and an ANN (Artificial Neural Network) based model is established to predict the task completion status for resource allocation among candidates. The trained ANN model has high prediction accuracy, and the ANN-based scheduling approach performs better than the preferred method in terms of the optimization objectives, including total cost, service satisfaction, and make-span. In addition, the proposed approach has potential in the application for smart manufacturing or Industry 4.0 because of its high response performance and good scalability.

Keywords: cloud manufacturing; real-time scheduling problem; artificial neural network

1. Introduction

The development of the Internet, automation, intelligent decision support, and other technologies has driven the manufacturing industry to transform towards digitalization, networking, and intelligence [1–3]. The manufacturing environment of enterprises has gradually expanded from the traditional workshop to large-scale, highly dynamic networked version [4–7], such as the CMfg (Cloud Manufacturing), where the cloud platform has pooled massive virtualized manufacturing resources from geographically distributed providers [8,9], including hardware resources (such as machining center, lathe, assembly line, materials, etc.), software resources (such as CAD/CAE, simulation, computing capabilities), human resources, knowledge resources, and so on. Therefore, how to schedule these manufacturing resources to serve emerging manufacturing tasks in real-time is not only an urgent requirement in the CMfg environment but also a common problem of the generalized networked manufacturing models, such as Smart Manufacturing and Industrial 4.0 [10,11]. Compared with the traditional workshop environment, real-time scheduling in the CMfg is more complicated because logistics factors need to be considered when using distributed massive heterogeneous manufacturing resources [12], and complex manufacturing projects bring precedence constraints at the same time. In addition, the provider needs to take into account its internal processing plan while coordinating with the demander, and the considerations of differences in management, specifications, and protocols among the enterprises are also needed [13,14]. Although there have been many studies focused on

modeling, solving, and optimization of the scheduling problem in the workshop environment, their research results cannot be directly applied to the networked manufacturing environment because the scale of the problem is dramatically increased.

In the academic consensus, the main kinds of users in the CMfg platform can be described by a so-called “tri-group” model [4,15,16], which is summarized as:

Demander Users who publish manufacturing tasks to the platform, and request to purchase the manufacturing resource services provided by the platform.

Provider Users who register manufacturing resources on the platform, providing virtualized manufacturing resources, such as software, equipment, materials, and labor.

Operator Users who operate and manage the platform; they (1) decompose the demands into manufacturing tasks, (2) encapsulate the registered manufacturing resources through virtualization technology, and (3) match the manufacturing services and provide decision support applications.

The CMfg environment within the research scope of this article can be briefly described, as shown in Figure 1, and the meaning of the symbols is described in Table 1.

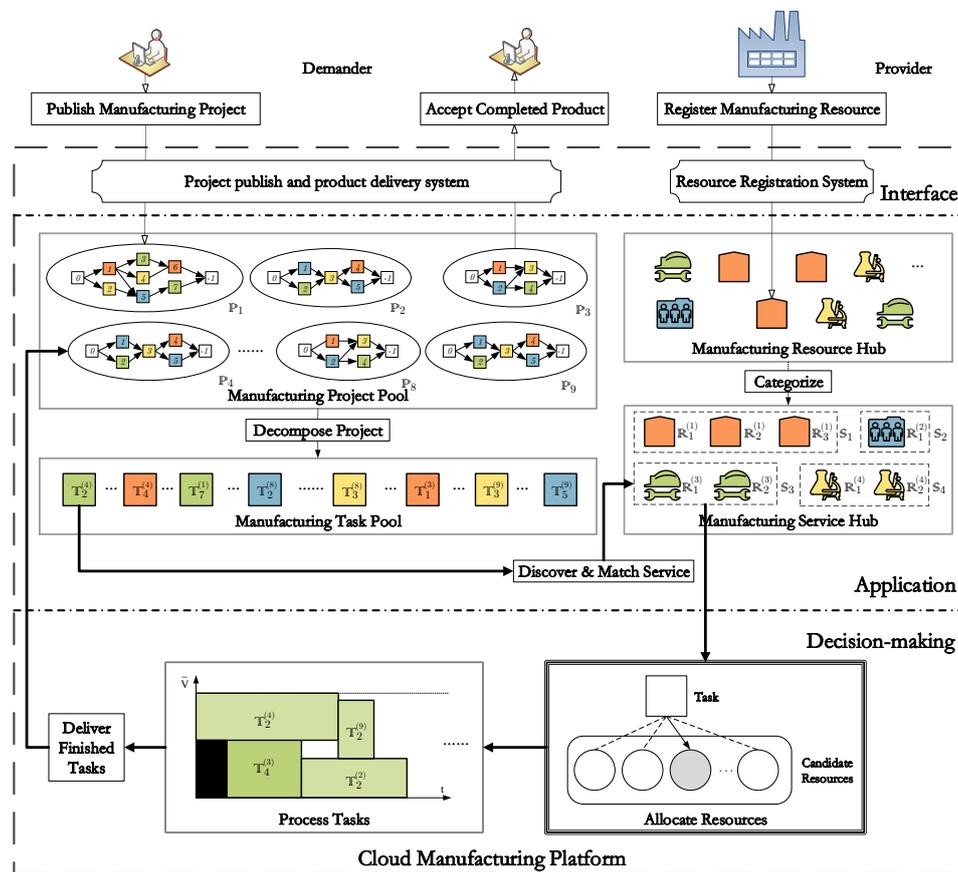


Figure 1. Architecture of the Cloud Manufacturing platform.

Table 1. Nomenclature.

Symbol	Description
$b_j^{(c)}$	The processing start time of $\mathbb{T}_j^{(c)}$.
c	Label of MP ($c \in \mathcal{C}$).
$d_j^{(c)}$	The ideal due time of $\mathbb{T}_j^{(c)}$, its upper and lower bound are $\bar{d}_j^{(c)}$ and $\underline{d}_j^{(c)}$.
$e_j^{(c)}$	The finish time of $\mathbb{T}_j^{(c)}$.
$g_j^{(c)}$	The ready time of $\mathbb{T}_j^{(c)}$.
i	Label of MP ($i \in \mathcal{I}_k$).
j	Label of MT ($j \in \mathcal{J}_c$).
k	Label of MS ($k \in \mathcal{K}$).
$l_j^{(c)}$	The logistics duration of $\mathbb{T}_j^{(c)}$.
$p_j^{(c)}$	The processing duration of $\mathbb{T}_j^{(c)}$.
$v_j^{(c)}$	The required resource capacity of $\mathbb{T}_j^{(c)}$.
$x_{i,j}^{(c)}$	0–1 decision variable to indicate if $\mathbb{T}_j^{(c)}$ will be allocated to $\mathbb{R}_i^{(k)}$ or not.
$\mathbf{y}(j, c, i, k)$	The actual completion status of $\mathbb{T}_j^{(c)}$ after processed in $\mathbb{R}_i^{(k)}$.
$\hat{\mathbf{y}}(j, c, i, k)$	The predicted completion status of $\mathbb{T}_j^{(c)}$ after processed in $\mathbb{R}_i^{(k)}$.
B_c	The publish time of \mathbb{P}_c .
$C_i^{(k)}$	The unit cost of $\mathbb{R}_i^{(k)}$.
$D_i^{(k)}(\mathbb{T}_j^{(c)})$	The preference function of $\mathbb{T}_j^{(c)}$ allocated on $\mathbb{R}_i^{(k)}$.
$L_{c,i}^{(k)}$	The logistics duration between \mathbb{P}_c and \mathbb{S}_i .
L_c	The logistics duration vector of \mathbb{P}_c , each element corresponds to a type-matched MR.
$\mathcal{M}_i^{(k)}(t)$	The in-processing set of MT in $\mathbb{R}_i^{(k)}$ at time t .
\mathbb{P}_c	MP labeled in c .
$P_i^{(k)}$	The service quality of $\mathbb{R}_i^{(k)}$.
$\mathcal{P}_j^{(c)}$	The immediate predecessor set of $\mathbb{T}_j^{(c)}$.
$\mathcal{Q}_i^{(k)}(t)$	The processing pending queue of MTs in $\mathbb{R}_i^{(k)}$ at time t .
$\mathbb{R}_i^{(k)}$	MR labeled in (i, k) .
\mathbb{S}_k	MS in type k .
$\mathcal{S}_j^{(c)}$	The immediate successor set of $\mathbb{T}_j^{(c)}$.
$\mathbb{T}_j^{(c)}$	MT labeled in (j, c) .
$V_i^{(k)}(t)$	The maximum available capacity of $\mathbb{R}_i^{(k)}$ at time t , its upper bound is $\bar{V}_i^{(k)}$.
$Z_x^{(c)}$	The object value of \mathbb{P}_c , $x \in \{d, m, u\}$ represents {make-span, cost, quality}.
$\kappa_j^{(c)}$	The required service type of $\mathbb{T}_j^{(c)}$.
$\mu_{i,x}$	The mean value of MTs in queue, $x \in \{g, p, l, v, \underline{d}, \bar{d}\}$ represents the feature mark.
$\sigma_{i,x}$	The standard deviation value of MTs in queue, $x \in \{g, p, l, v, \underline{d}, \bar{d}\}$ represents the feature mark.
$\varphi_{i,n}^{(c,j)}$	The probability for due time value located in interval n of $\mathbb{T}_j^{(c)}$ if $\mathbb{R}_i^{(k)}$ is allocated ($n = 0, 1, 2$).
$\mathbb{1}_{\text{predicate}}$	Indicator function ($\mathbb{1}_{\text{predicate}} = 1$ if the predicate is true, else 0).
$ \mathcal{S} $	The element count of set \mathcal{S} .

This architecture is mainly composed of three layers of functions, namely interface, application, and decision-making. The real-time scheduling problem comes from the decision-making layer. Accordingly, the process of the demander using the services provided by the platform can be described as follows:

1. Demanders publish MP (Manufacturing Project) to the platform;
2. Platform decomposes MP into MTs (Manufacturing Tasks);
3. Platform discovers and matches MR (Manufacturing Resource) in type-matched MS (Manufacturing Service) for these MTs;
4. Platform allocates MR for the processing of MT.

5. Providers arrange MTs using professional software and deliver the completed MTs to demanders.

Therefore, the scheduling problem in the CMfg environment is as follows: how to reasonably allocate the emerging MTs to the type-matched MRs in real-time under the constraints of task precedence, resource occupation, and logistics duration while reducing the service costs, improving the service quality, and shortening the make-span for each MP?

Since the scheduling environment of CMfg is larger and more dynamic than that of the traditional workshop [17,18], it is much more difficult to find the optimal solution. As a consequence, the focus of related research turns to find the approximate optimal solution. As a major branch of approximation methods, a search-based heuristic scheduling approach expands the current schedule with well-designed strategies. The keys to designing such an approach are the neighborhood structure and the search direction, such as Simulated Annealing, Tabu-search, Discrete Search, Genetic Algorithm, and so on [19–23]. However, these algorithms cannot be directly applied to the scheduling problem in the CMfg environment because they will take a long time for the solution searching.

In order to solve the real-time scheduling problem, researchers have proposed and developed numerous approaches and algorithms that can be divided into two categories: generative and adaptive. Generative scheduling adopts the idea of integrating partial-schedules, and the key lies in the design of priority rules. However, the priority rules for the determined manufacturing environment are not suitable for dynamic random situations [24]. Hence, adaptive methods have attracted more researchers. Adaptive scheduling, which is also called rescheduling, modifies the existing schedule dynamically according to the decision environment. Proactive and reactive are the two main modes of rescheduling [25]. Due to the uncertain factors, such as task delay, unit failure, and order insertion, the disturbances become difficult to predict, which means that the proactive rescheduling cannot be used in the real-time environments [26]. As a result, more and more researchers have begun to focus on reactive or hybrid rescheduling methods [27]. One group studied the random resource constrained project scheduling problem. They transformed the problem into a multi-stage decision problem, and made dynamic decisions by designing rescheduling strategies [28]. Another group studied the rescheduling problem with new operation insertion in a single machine environment. They set up an event-triggered rescheduling model with the goal of minimizing the absolute deviation of the maximum delay compared with the baseline [29]. For the dynamic manufacturing environment where tasks arrive randomly, a dynamic event-driven task rescheduling method is designed to avoid service rearrangement, and the constructed parallel processing strategy takes service time, logistics time, the earliest available time, and other factors into consideration at the same time for optimal service selection [30]. Researchers have also proposed two reactive scheduling strategies based on the selection and buffering to solve the random resource constrained project scheduling problem. They found that the buffer-based reaction rules combined with the baseline scheduling turned out to be an effective solution [31].

The above reviews indicate that it is difficult to efficiently allocate massive manufacturing resources in the CMfg environment for the real-time scheduling. Research on the real-time scheduling in the CMfg is scarce, and the proposed scheduling methods are based on reactive scheduling strategy, the baseline of which is set in advance and needs to be modified as the environment changes. For the dynamic CMfg environment, such scheduling methods will consume numerous computing resources. Changing schedules often results in the re-allocation of resources, which will lead to additional logistics costs. Generally, real-time scheduling using these approaches requires a trade-off between the solution quality and decision efficiency. With the development of artificial intelligence, more and more teams are trying to apply related technologies to the manufacturing scope [32]. In order to solve the scheduling problem in the manufacturing environment, many neural networks are used to determine the key parameters to improve the performance of the genetic algorithm [33,34].

In this paper, we convert the aforementioned scheduling problem into the form of estimating the optimal objective value, and an ANN-based model is established to predict the task completion status for each candidate resource as the principle for allocating manufacturing tasks. The trained ANN

model has high prediction accuracy, and the proposed ANN-based approach performs well in terms of the optimization objectives, including cost, satisfaction, and make-span. The high responsiveness of the ANN-based approach makes it suitable for the real-time scheduling in the CMfg environment.

The structure of the paper is as follows. First, we establish a mathematical model of the real-time scheduling problem in the CMfg environment in Section 2. Then, the solution framework of this problem is defined in Section 3. As the key issue in this framework, the decision of task allocation is made using a trained ANN model. Section 4 conducts comparison experiments for scheduling methods in terms of objectives and responsiveness. The application design for the proposed ANN-based approach is depicted in Section 5. Section 6 presents the conclusion.

2. Mathematical Modeling for the CMfg Scheduling Problem

According to the CMfg platform architecture (Figure 1), the real-time scheduling can be described as the procedure shown in Figure 2.

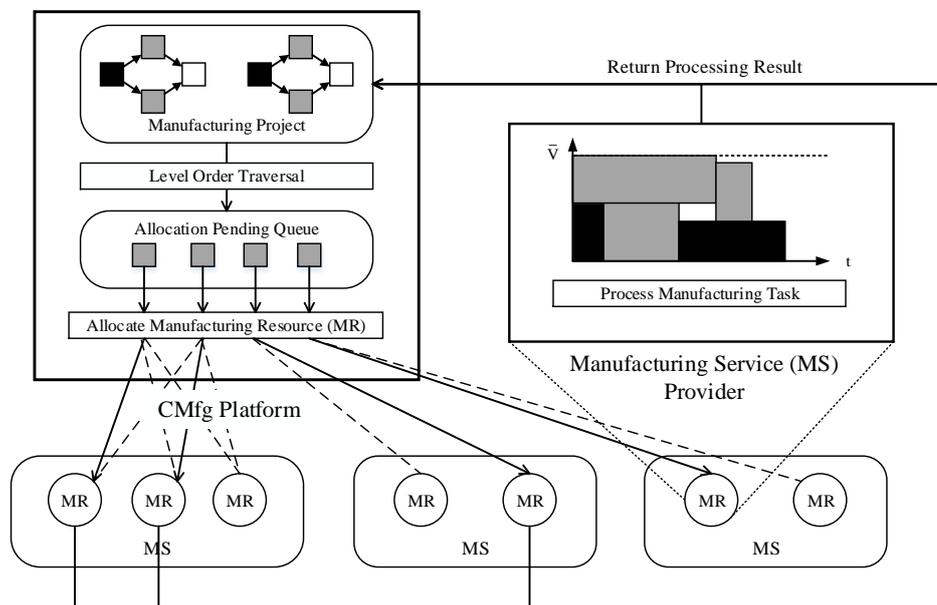


Figure 2. Real-time Schedule Procedure in the Cloud Manufacturing (CMfg) Environment.

The MPs in the CMfg platform consist of MTs with precedence sequence, the waiting ones for allocation are in white, and the ones in allocating are filled in gray, while the completed ones are filled in black. By using level-order traversal, the operator can allocate MTs to the type-matched MRs without precedence constraints, and these tasks will be manufactured by the corresponding providers.

Basic assumptions, as follows, are needed to focus on our research scope before mathematical modeling:

1. The set-up time for MT is already included in the processing time;
2. No interruption is considered in the processing of MTs;
3. The capacity of MR occupied by task processing will be released when the processing is completed;
4. Transportation logistics need to be considered before and after the processing of MT.

2.1. Formal Expression of the Main Components

The main components of the CMfg environment are MT and MR. Due to the consideration of management granularity, these two components are not directly presented to the scheduling problem.

Providers distributed in different locations register their encapsulated and virtualized MR capabilities to the CMfg platform. Then, the platform operator classifies these MRs according to

their capability types and abstracts them into MS, which is an integrated form of MRs for the demander to use. Specifically, MS of type k can be expressed by a tuple, as shown in Equation (1).

$$\mathbb{S}_k \triangleq (\mathcal{I}_k, \mathcal{Q}_k(t)) \quad \forall k \in \mathcal{K} \tag{1}$$

where \mathcal{I}_k represents the label set of MR in the same type, and $\mathcal{Q}_k(t)$ denotes the allocation pending queue of MTs at time t .

For MR labeled i in \mathbb{S}_k , its attributes are expressed as Equation (2).

$$\mathbb{R}_i^{(k)} \triangleq (C_i^{(k)}, P_i^{(k)}, V_i^{(k)}(t), \mathcal{Q}_i^{(k)}(t), \mathcal{M}_i^{(k)}(t)) \quad \forall i \in \mathcal{I}_k \tag{2}$$

where $C_i^{(k)}$ and $P_i^{(k)}$ represent unit cost and service quality, respectively. $V_i^{(k)}(t)$ is the maximum available capacity at time t , and $\bar{V}_i^{(k)}$ is its upper bound. $\mathcal{Q}_i^{(k)}(t)$ and $\mathcal{M}_i^{(k)}(t)$ denote the processing pending queue of MTs and the active MT set defined in Equation (3) at time t , respectively.

$$\mathcal{M}_i^{(k)}(t) \triangleq \{(j, c) \mid b_j^{(c)} \leq t < e_j^{(c)}, x_{i,j}^{(c)} = 1, j \in \mathcal{J}_c, c \in \mathcal{C}\} \quad \forall i \in \mathcal{I}_k \tag{3}$$

where (j, c) is the label mark of MT, $b_j^{(c)}$ and $e_j^{(c)}$, respectively, represent the start time and finish time of processing this MT. Symbol $x_{i,j}^{(c)}$ is a 0–1 decision variable defined as Equation (4).

$$x_{i,j}^{(c)} = \begin{cases} 1 & \text{if } \mathbb{T}_j^{(c)} \text{ is allocated to } \mathbb{R}_i^{(k)} \text{ for processing} \\ 0 & \text{otherwise.} \end{cases} \quad i \in \mathcal{I}_k; j \in \mathcal{J}_c; c \in \mathcal{C} \tag{4}$$

In the cloud platform, MP that is published by the demander will be pooled, and the precedence relationship of the contained MTs can be described as the activity-on-vertex network, as shown in Figure 3.

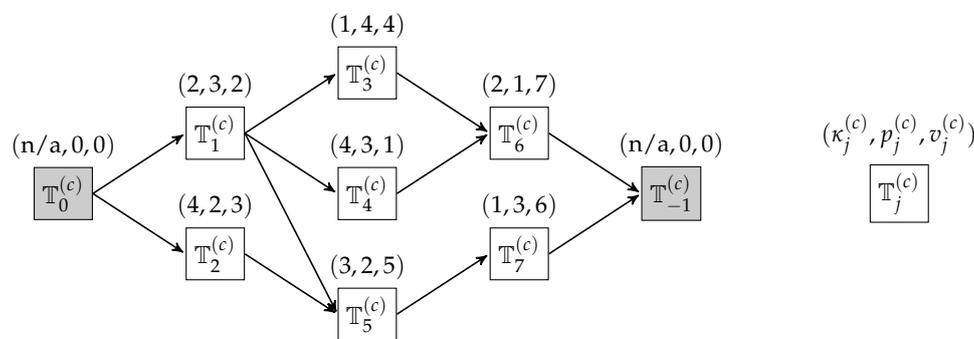


Figure 3. Activity-on-Vertex of Manufacturing Project (MP) \mathbb{P}_c (n/a means not applicable).

Where $\mathbb{T}_0^{(c)}$ and $\mathbb{T}_{-1}^{(c)}$ are dummy MTs to help stylize the MP into a single-input-single-output graph. Specifically, an MP labelled c can be expressed by a tuple shown in Equation (5).

$$\mathbb{P}_c \triangleq (\mathcal{J}_c, B_c, L_c) \quad \forall c \in \mathcal{C} \tag{5}$$

where \mathcal{J}_c represents the label set of MT included in \mathbb{P}_c , B_c denotes the publish time, and L_c is the logistics duration vector defined as Equation (6).

$$L_c \triangleq [L_{c,i}^{(k)}]_{i \in \mathcal{I}_k, k \in \mathcal{K}} \quad \forall c \in \mathcal{C} \tag{6}$$

where $L_{c,i}^{(k)}$ is the logistics duration between \mathbb{P}_c and $\mathbb{R}_i^{(k)}$.

For MT labelled j in \mathbb{P}_c , its attributes can be represented as Equation (7).

$$\mathbb{T}_j^{(c)} \triangleq (v_j^{(c)}, \kappa_j^{(c)}, p_j^{(c)}, \mathcal{P}_j^{(c)}, \mathcal{S}_j^{(c)}) \quad \forall j \in \mathcal{J}_c; c \in \mathcal{C} \tag{7}$$

where $v_j^{(c)}, \kappa_j^{(c)}$, and $p_j^{(c)}$ represent required service capacity, service capability type, and processing time, respectively. $\mathcal{P}_j^{(c)}$ and $\mathcal{S}_j^{(c)}$, respectively, denote the immediate predecessor and the successor set.

In this way, the mathematical model of real-time scheduling in the CMfg environment can be expressed as Equations (8)–(16), and it is a multi-objectives optimization problem.

$$\text{Minimize } \mathbf{Z}_{t_b:t_e} \triangleq \frac{1}{|\mathcal{C}_{t_b:t_e}|} \sum_{c \in \mathcal{C}_{t_b:t_e}} \mathbf{Z}^{(c)} \tag{8}$$

$$\text{s.t. } \left\{ \begin{array}{ll} \mathbf{Z}^{(c)} \triangleq [Z_m^{(c)} \quad Z_d^{(c)} \quad -Z_u^{(c)}]^T & c \in \mathcal{C} \tag{9} \\ \sum_{i \in \mathcal{I}_k} x_{i,j}^{(c)} \mathbb{1}_{\kappa_j^{(c)}=k} = 1 & j \in \mathcal{J}_c; c \in \mathcal{C}; k \in \mathcal{K} \tag{10} \\ \sum_{i \in \mathcal{I}_k} x_{i,j}^{(c)} \mathbb{1}_{\kappa_j^{(c)} \neq k} = 0 & j \in \mathcal{J}_c; c \in \mathcal{C}; k \in \mathcal{K} \tag{11} \\ p_j^{(c)} \leq e_j^{(c)} - b_j^{(c)} & j \in \mathcal{J}_c; c \in \mathcal{C} \tag{12} \\ x_{i,j}^{(c)} v_j^{(c)} \leq V_i^{(k)}(t) & t \in [b_j^{(c)}, e_j^{(c)}]; i \in \mathcal{I}_k; k \in \mathcal{K}; j \in \mathcal{J}_c; c \in \mathcal{C} \tag{13} \\ \sum_{(j,c) \in \mathcal{M}_i^{(k)}(t)} v_j^{(c)} \leq V_i^{(k)}(t) & t \geq 0; i \in \mathcal{I}_k; k \in \mathcal{K} \tag{14} \\ g_j^{(c)} \geq \max_{j' \in \mathcal{P}_j^{(c)}} \{e_{j'}^{(c)} + l_{j'}^{(c)}\} & j \in \mathcal{J}_c; c \in \mathcal{C} \tag{15} \\ g_j^{(c)} + l_j^{(c)} \leq b_j^{(c)} & j \in \mathcal{J}_c; c \in \mathcal{C} \tag{16} \end{array} \right.$$

where $\mathcal{C}_{t_b:t_e}$ denotes the set of arrived MPs during time interval $[t_b, t_e)$, and the objective function of Equation (8) is the accumulated mean value of objective vector $\mathbf{Z}^{(c)}$ defined as Equation (9). The composition of the objectives and the Constraints (10)–(16) will be described in detail in the rest of this section.

2.2. Optimization Objectives for Real-Time Scheduling

For each MP, the scheduling optimization objectives include three aspects, namely cost, make-span, and service quality. Without loss of generality, we proceed from the perspective of \mathbb{P}_c .

First of all, the service usage cost for any MP is incurred when the demander requests and uses specific MRs included in the type-matched MSs, it is equal to the sum of the usage costs of MTs inside the MP. The service usage cost for MT is determined by the unit cost of the allocated MR and the requirement for task processing. Specifically, if $\mathbb{T}_j^{(c)}$ is allocated to $\mathbb{R}_i^{(k)}$ for processing, the service usage cost is determined by Equation (17).

$$z_{j,m}^{(c)} = p_j^{(c)} v_j^{(c)} \sum_{i \in \mathcal{I}} x_{i,j}^{(c)} C_i^{(k)} \tag{17}$$

Then, the service usage cost for \mathbb{P}_c will be Equation (18)

$$Z_m^{(c)} = \sum_{j \in \mathcal{J}_c} z_{j,m}^{(c)} \tag{18}$$

Secondly, the make-span of MP refers to the overall processing time to complete all its MTs. It needs to consider the logistics duration for all the MTs inside the MP. According to the

activity-on-vertex of \mathbb{P}_c as shown in Figure 3, its make-span can be expressed as Equation(19), where $e_{-1}^{(c)}$ belongs to dummy $\mathbb{T}_{-1}^{(c)}$.

$$Z_d^{(c)} = e_{-1}^{(c)} \tag{19}$$

Thirdly, the quality of the product refers to the satisfaction of manufacturing \mathbb{P}_c , and it is determined by the intrinsic service quality of specific MRs. Since the satisfaction of processing $\mathbb{T}_j^{(c)}$ can be expressed as Equation (20), the quality of the product for processing \mathbb{P}_c can be defined as Equation (21), which is the lowest satisfaction obtained by processing MTs included in \mathbb{P}_c .

$$z_{j,u}^{(c)} = \sum_{i \in \mathcal{I}} x_{i,j}^{(c)} P_i^{(k)} \tag{20}$$

$$Z_u^{(c)} = \min_{j \in \mathcal{J}_c} \{z_{j,u}^{(c)}\} \tag{21}$$

2.3. Constraints for Real-Time Scheduling

Constraints for real-time scheduling problems in the CMfg environment include task precedence, resources occupation, logistics duration, and so on.

Specifically, Constraints (10) and (11) indicate the logical limitation of allocation, which means that for type-matched MRs, any MT can only be assigned to one of them for processing, and each of the MT can only be processed once. As formulated in Equation (12), the processing time of any MT needs to be guaranteed, that is, the time span between the start time $b_j^{(c)}$ and the finish time $e_j^{(c)}$ of processing is at least equal to the required processing time $p_j^{(c)}$.

During the processing of MT, Equation(13) means the available capacity of MR needs to be no less than the capacity required by the MT, and Equation (14) indicates the sum of occupied capacity by active MTs does not exceed the maximum resource capacity at time t . Take $\mathbb{R}_2^{(0)}$, for example, as shown in Figure 4, the capacity upper bound is $\bar{V}_2^{(0)} = 5$, and the rectangle in black indicates the resource occupation of the provider’s internal planning task. At time $t = 16$, the maximum available capacity of $\mathbb{R}_2^{(0)}$ becomes $V_2^{(0)}(16) = \bar{V}_2^{(0)} - 1 = 4$, so it is reasonable to manufacture $\mathbb{T}_4^{(3)}$ ($v_4^{(3)} = 3, b_4^{(3)} = 12, e_4^{(3)} = 18$) and $\mathbb{T}_2^{(1)}$ ($v_2^{(1)} = 1, b_2^{(1)} = 14, e_2^{(1)} = 19$) at this time slot since the sum of occupied capacity does not exceed ($v_2^{(1)} + v_4^{(3)} = 4 \leq V_2^{(0)}(16)$); hence, we get the active MT set as $\mathcal{M}_2^{(0)}(16) = \{(2, 1), (4, 3)\}$.

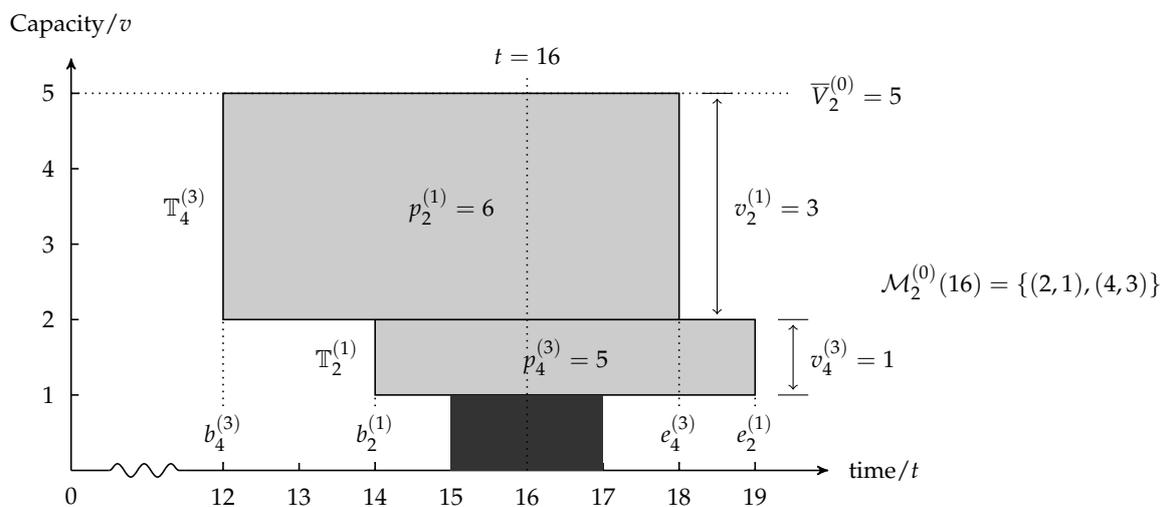


Figure 4. Capacity occupation of $\mathbb{R}_2^{(0)}$, checking at $t = 16$.

Figure 5 depicts the precedence constraints in MP that contain Equations (15) and (16), which, respectively, mean the ready time of any MT is no earlier than the complete time of its predecessors and the start time of any MT is no earlier than its ready time plus its logistics duration.

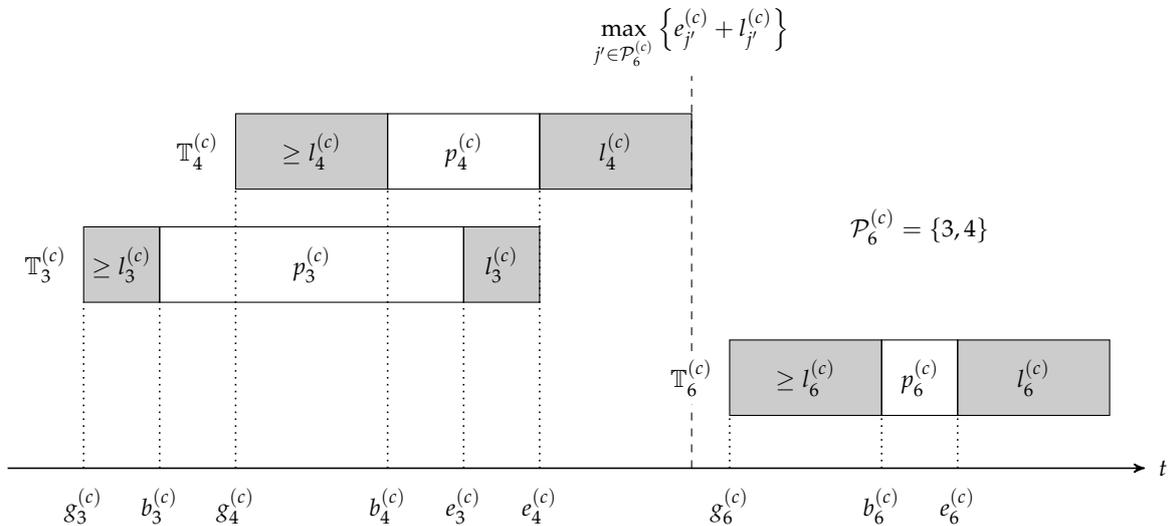


Figure 5. Precedence constraint of $T_6^{(c)}$ with logistics duration considered.

Where the auxiliary variable $l_j^{(c)}$ is defined as Equation (22).

$$l_j^{(c)} = \sum_{i \in \mathcal{I}_k} x_{i,j}^{(c)} L_{c,i}^{(k)} \quad k = \kappa_j^{(c)}; j \in \mathcal{J}_c; c \in \mathcal{C} \tag{22}$$

3. Artificial Neural Network based Resource Allocation Methodology

The main purpose of the real-time scheduling problem is to allocate MSs for every MT, and the allocation of $T_j^{(c)}$ can be depicted as Figure 6. The candidate MRs are determined by their capability type and the upper bound of available capacity as formulated in Equation (23).

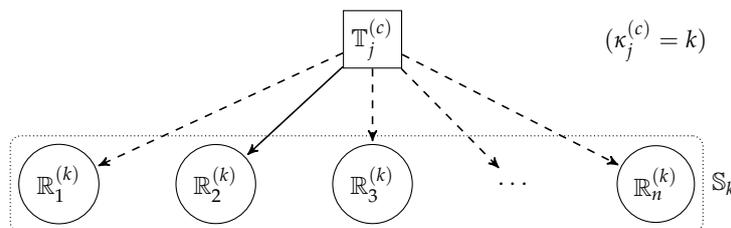


Figure 6. Allocating $T_j^{(c)}$ to one of the candidate MRs in S_k , the capability type is matched.

$$\mathcal{I}_k(v) \triangleq \{i \in \mathcal{I}_k \mid k = \kappa_j^{(c)}, \bar{V}_i^{(k)} \geq v\} \quad \forall k \in \mathcal{K} \tag{23}$$

However, since the number of candidate MRs is large and the processing statuses of these MRs are changing over time, the conventional search algorithms are no longer applicable because they will spend a long time checking all of the possible time intervals in every candidate MR. Therefore, we adopt an ANN that is based on the multi-layer perceptron architecture to speed up the searching procedure by estimating the objective values of each candidate MR.

Specifically, take $T_j^{(c)}$ as an example, after filtering out candidate MRs according to (23), the objective values projected on MT can be estimated by the completion status prediction model as shown in Figure 7.

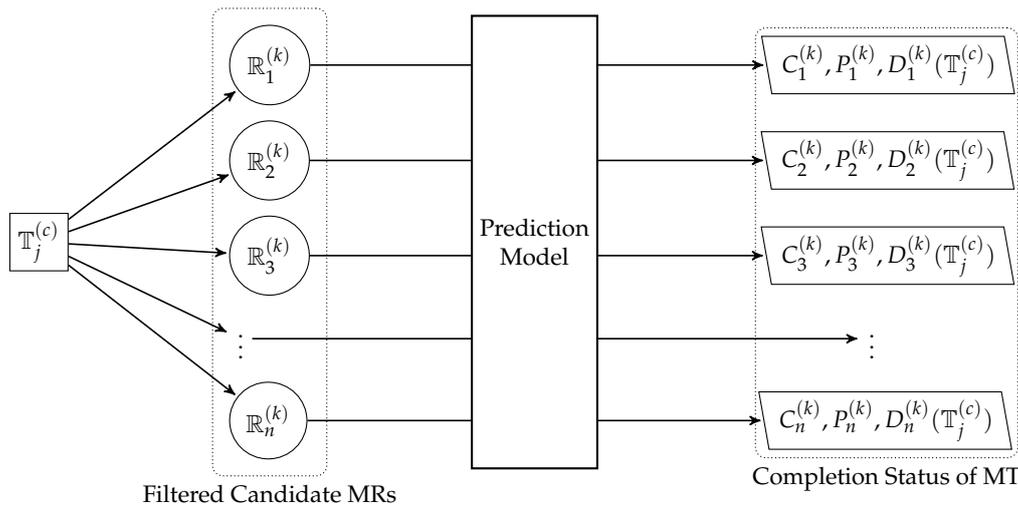


Figure 7. Procedure of Manufacturing Task (MT) completion status prediction on candidate Manufacturing Resources (MRs.)

Since both $C_i^{(k)}$ and $P_i^{(k)}$ are intrinsic values of $R_i^{(k)}$, the key of this prediction model is to predict the completion time $d_j^{(k)}$ of $T_j^{(k)}$. It can be known from the activity-on-vertex structure of one MP (Figure 3), that is, one MT will affect the make-span of its MP only if it is on the critical path, otherwise it has a flexible interval for completion time. Using the level-order traversal algorithm, the upper and lower bounds of the ideal completion time of $T_j^{(c)}$ that may not affect the make-span of its MP can be obtained. Then, the upper and lower bounds can divide the time axis into three adjacent sections, and their probabilities are Equations (24)–(26).

$$\varphi_{i,2}^{(c,j)} = Pr \{ \underline{d}_j^{(c)} \leq d_j^{(c)} | x_{i,j}^{(c)} = 1 \} \tag{24}$$

$$\varphi_{i,1}^{(c,j)} = Pr \{ \underline{d}_j^{(c)} < d_j^{(c)} \leq \bar{d}_j^{(c)} | x_{i,j}^{(c)} = 1 \} \tag{25}$$

$$\varphi_{i,0}^{(c,j)} = Pr \{ \bar{d}_j^{(c)} < d_j^{(c)} | x_{i,j}^{(c)} = 1 \} \tag{26}$$

where $\varphi_{i,n}^{(c,j)}$ means the probability of completion time $d_j^{(c)}$ lies in time interval “Completion status n ”. Since the completion status of $T_j^{(c)}$ can only belong to one of these three statuses, we have $\varphi_{i,0}^{(c,j)} + \varphi_{i,1}^{(c,j)} + \varphi_{i,2}^{(c,j)} = 1$. Then the allocation preference for MR candidates of $T_j^{(c)}$ can be defined as Equation (27), which motivates the allocation of MR to process $T_j^{(c)}$ as early as possible.

$$D_i^{(k)}(T_j^{(c)}) = \sum_{n=0}^2 10^n \cdot \varphi_{i,n}^{(c,j)} \tag{27}$$

As shown in Figure 8, the ANN predicts the completion status of MT.

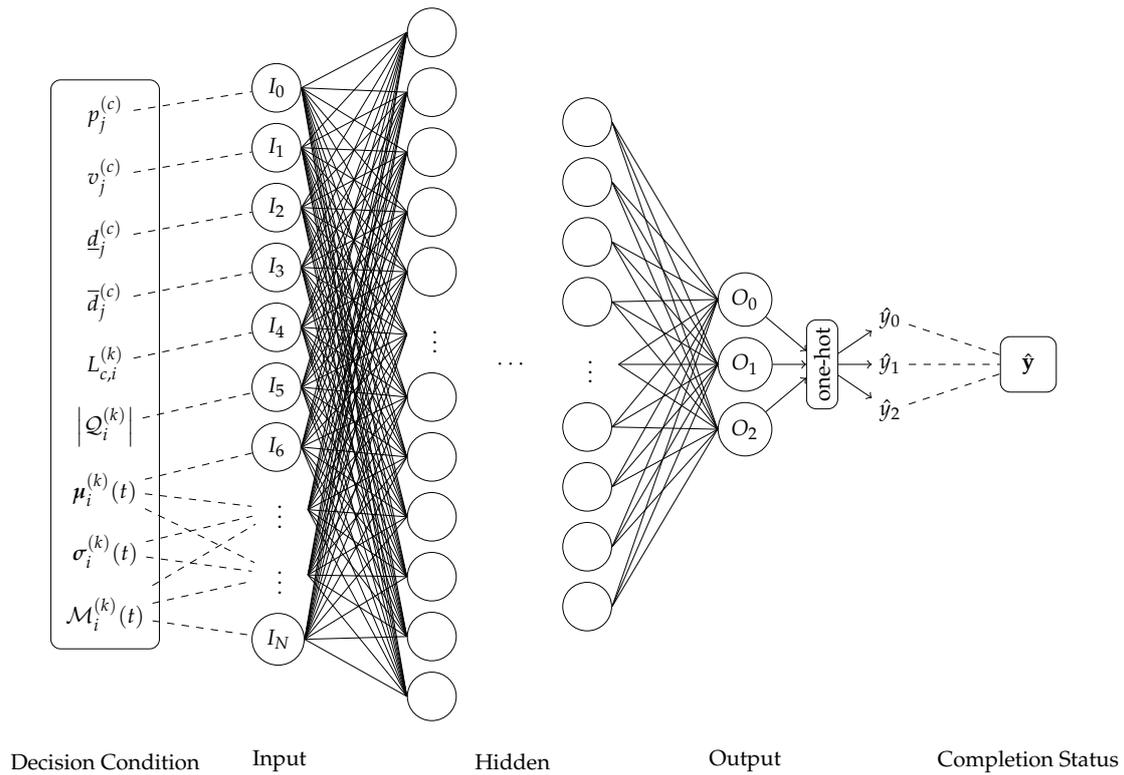


Figure 8. Artificial Neural Network model for the MT completion status prediction.

Where O_0, O_1, O_2 represent $\varphi_{i,0}^{(c,j)}, \varphi_{i,1}^{(c,j)}, \varphi_{i,2}^{(c,j)}$, respectively. The expression of actual completion status of $\mathbb{T}_j^{(c)}$ processed on $\mathbb{R}_i^{(k)}$ is defined as Equation (28).

$$\mathbf{y}(j, c, i, k) = [y_0 \quad y_1 \quad y_2]^T(j, c, i, k) \tag{28}$$

The inputs of this ANN are written in Input (29), which means all the possible features from the decision condition.

$$\mathbf{I}(j, c, i, k) \triangleq [I_0 \quad I_1 \quad \dots \quad I_N]^T(j, c, i, k) \tag{29}$$

The decision condition for $\mathbb{T}_j^{(c)}$ allocated to $\mathbb{R}_i^{(k)}$ consists of intrinsic attributes of $\mathbb{T}_j^{(c)}$ ($p_j^{(c)}, v_j^{(c)}, d_j^{(c)}, \bar{d}_j^{(c)}, L_{c,i}^{(k)}$) and dynamic features of $\mathbb{R}_i^{(k)}$ defined in Equation (30).

$$\begin{matrix} \begin{bmatrix} g_{j_0}^{(c_0)} & p_{j_0}^{(c_0)} & l_{j_0}^{(c_0)} & v_{j_0}^{(c_0)} & d_{j_0}^{(c_0)} & \bar{d}_{j_0}^{(c_0)} \end{bmatrix}^T & \dots & \mathbb{T}_{j_0}^{(c_0)} \\ \begin{bmatrix} g_{j_1}^{(c_1)} & p_{j_1}^{(c_1)} & l_{j_1}^{(c_1)} & v_{j_1}^{(c_1)} & d_{j_1}^{(c_1)} & \bar{d}_{j_1}^{(c_1)} \end{bmatrix}^T & \dots & \mathbb{T}_{j_1}^{(c_1)} \\ \begin{bmatrix} g_{j_2}^{(c_2)} & p_{j_2}^{(c_2)} & l_{j_2}^{(c_2)} & v_{j_2}^{(c_2)} & d_{j_2}^{(c_2)} & \bar{d}_{j_2}^{(c_2)} \end{bmatrix}^T & \dots & \mathbb{T}_{j_2}^{(c_2)} \\ \vdots & & \\ \begin{bmatrix} g_{j_n}^{(c_n)} & p_{j_n}^{(c_n)} & l_{j_n}^{(c_n)} & v_{j_n}^{(c_n)} & d_{j_n}^{(c_n)} & \bar{d}_{j_n}^{(c_n)} \end{bmatrix}^T & \dots & \mathbb{T}_{j_n}^{(c_n)} \end{matrix} \quad , n = |\mathcal{Q}_i^{(k)}| \tag{30}$$

$$\begin{matrix} \begin{bmatrix} \mu_{i,g}^{(k)} & \mu_{i,p}^{(k)} & \mu_{i,l}^{(k)} & \mu_{i,v}^{(k)} & \mu_{i,d}^{(k)} & \mu_{i,\bar{d}}^{(k)} \end{bmatrix}^T & \dots & \boldsymbol{\mu}_i^{(k)}(t) \\ \begin{bmatrix} \sigma_{i,g}^{(k)} & \sigma_{i,p}^{(k)} & \sigma_{i,l}^{(k)} & \sigma_{i,v}^{(k)} & \sigma_{i,d}^{(k)} & \sigma_{i,\bar{d}}^{(k)} \end{bmatrix}^T & \dots & \boldsymbol{\sigma}_i^{(k)}(t) \end{matrix}$$

where $\boldsymbol{\mu}_i^{(k)}(t)$ and $\boldsymbol{\sigma}_i^{(k)}(t)$ are the mean value and standard deviation value of the processing pending queue $\mathcal{Q}_i^{(k)}$ at time t , respectively.

The outputs of the ANN are formulated in Output (31), which means the possibilities of completion status as Equations (24)–(26).

$$\mathbf{O}(j, c, i, k) \triangleq [O_0 \ O_1 \ O_2]^T(j, c, i, k) \tag{31}$$

Since the real completion status obtained by sampling is a discrete value as Equation (32), we convert the outputs into the style of one-hot encoding, which is in the same style of Equation (28).

$$\hat{\mathbf{y}}(j, c, i, k) \triangleq \begin{cases} \begin{bmatrix} 1 & 0 & 0 \end{bmatrix}^T & \text{Completion status 2, } (d_j^{(c)} \leq \underline{d}_j^{(c)}) \\ \begin{bmatrix} 0 & 1 & 0 \end{bmatrix}^T & \text{Completion status 1, } (\underline{d}_j^{(c)} < d_j^{(c)} \leq \bar{d}_j^{(c)}) \\ \begin{bmatrix} 0 & 0 & 1 \end{bmatrix}^T & \text{Completion status 0, } (\bar{d}_j^{(c)} \leq d_j^{(c)}) \end{cases} \tag{32}$$

4. Numerical Results

4.1. Experimental Environment Setting

We conduct comparative experiments to verify the effectiveness of the proposed ANN-based approach for real-time scheduling, and the dataset is adapted from MPSPLib(www.mpsplib.com/download.php), which consists of multiple project instances defined in Table 2.

Table 2. Project Instance in MPSPLib (take file j301_1.sm as an example).

MT Label (j)	Processing Time ($p_j^{(c)}$)	MS Type ($\kappa_j^{(c)}$)	MR Capacity ($v_j^{(c)}$)	Successors ($\mathcal{S}_j^{(c)}$)
1	0	n/a	0	2, 3, 4
2	8	0	4	6, 11, 15
⋮	⋮	⋮	⋮	⋮
31	2	2	2	32
32	0	n/a	0	n/a

Processing Time ($p_j^{(c)}$) and MR Capacity ($v_j^{(c)}$) are dimensionless quantities.

According to the MT scale of projects in the aforementioned MPSPLib, the experimental dataset is grouped as follows:

Exp-30 Each MP in this group contains 30 MTs;

Exp-60 Each MP in this group contains 60 MTs;

Exp-90 Each MP in this group contains 90 MTs;

Exp-120 Each MP in this group contains 120 MTs;

Exp-mix MT scale of each MP in this group can be any value in {30, 60, 90, 120}.

Each project in the MPSPLib is enabled with random features as Equations (33)–(37) to imitate the process of publishing MPs to the CMfg platform from the demanders and to enlarge the diversity of MS from the providers.

$$\begin{cases} L_{c,i}^{(k)} & \sim U(\text{dist_min}, \text{dist_max}) & (33) \\ |\mathcal{I}_k| & \sim U(\text{n_min}, \text{n_max}) & (34) \\ C_i^{(k)} & \sim U(\text{c_min}, \text{c_max}) & (35) \\ \bar{V}_i^{(k)} & \sim U(\text{v_min}, \text{v_max}) & (36) \\ P_i^{(k)} & \sim U(\text{p_min}, \text{p_max}) & (37) \end{cases}$$

where $U(a, b)$ represents a random distribution over the interval $[a, b]$, and the involved parameters are listed in Table 3.

Table 3. Parameters for MP and MS in the experimental environment.

Name	Value	Description
dist_max	10	Maximum distance
dist_min	0	Minimum distance
num_type	4	Number of capability type of MS
n_max	5	Maximum number of MRs in each MS
n_min	1	Minimum number of MRs in each MS
c_max	50	Maximum unit cost of MR
c_min	1	Minimum unit cost of MR
v_max	40	Maximum capacity value of MR
v_min	10	Minimum capacity value of MR
p_max	200	Maximum service quality value of MR
p_min	5	Minimum service quality value of MR

4.2. Preparation for Real-Time ANN-Based Scheduling Approach

The network inside the ANN-based approach is constructed in PyTorch, and its main parameters are listed in Table 4.

Table 4. Parameters for the training of the Artificial Neural Network (ANN).

Name	Value	Description
learning_rate	0.001	Learning rate of ANN
batch_size	64	Batch size of dataset
hidden_layer	(300,100,80,40)	Size of hidden layers
num_exp_iter	20	Number of iterations in the training process
num_epochs	1000	Number of epochs for gradient descent for each training iteration

Before starting to train the ANN model, we need to sample the data that is consistent with the style of Input (29) and Output (31). The quality of an ANN model can be judged by its cross-entropy loss function in Equation (38) during training, as depicted in Figure 9.

$$\theta_{loss} \triangleq - \sum_{n=0}^2 \hat{y}_n \log y_n \tag{38}$$

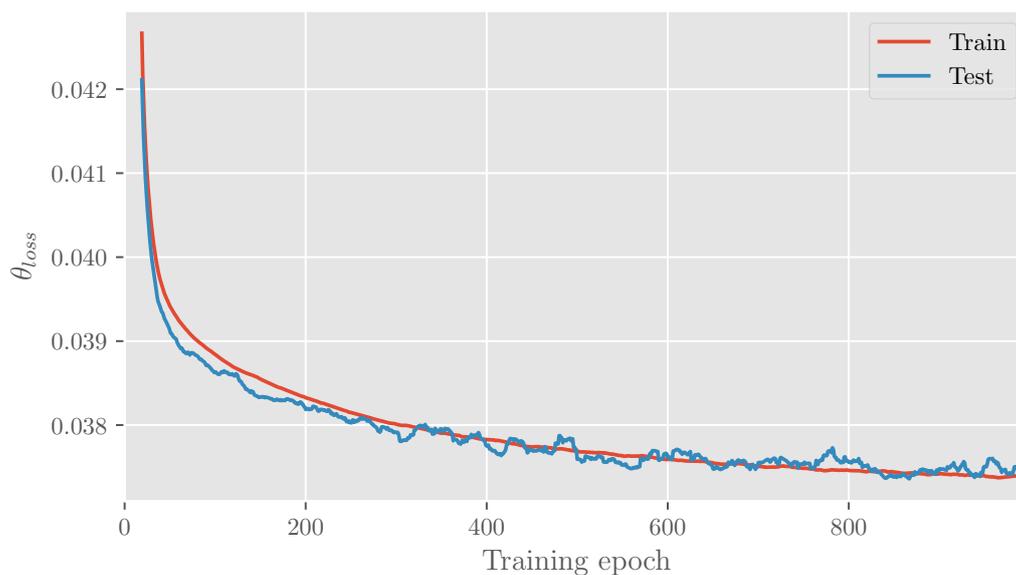


Figure 9. Cross-entropy loss of train/test dataset over 1000 training epochs.

As shown in Figure 9, after 1000 epochs, the training loss and testing loss both reached a small range and gradually converged. As the number of training epoch increases, the difference between training loss and testing loss becomes smaller and smaller, which indicates that the proposed ANN-based approach has the ability for generalization. Figure 10 plots the prediction accuracy as Equation (39) to evaluate the performance of the ANN model.

$$\theta_{accu} \triangleq \frac{\sum_{j,c,i,k \in \mathcal{J}_c, \mathcal{C}, \mathcal{I}_k, \mathcal{K}} \mathbb{1}_{\hat{y}(j,c,i,k)=y(j,c,i,k)}}}{\sum_{c \in \mathcal{C}, k \in \mathcal{K}} \mathbb{1}_{\kappa_j^{(c)}=k} \cdot |\mathcal{J}_c| \cdot |\mathcal{I}_k|} \quad (39)$$

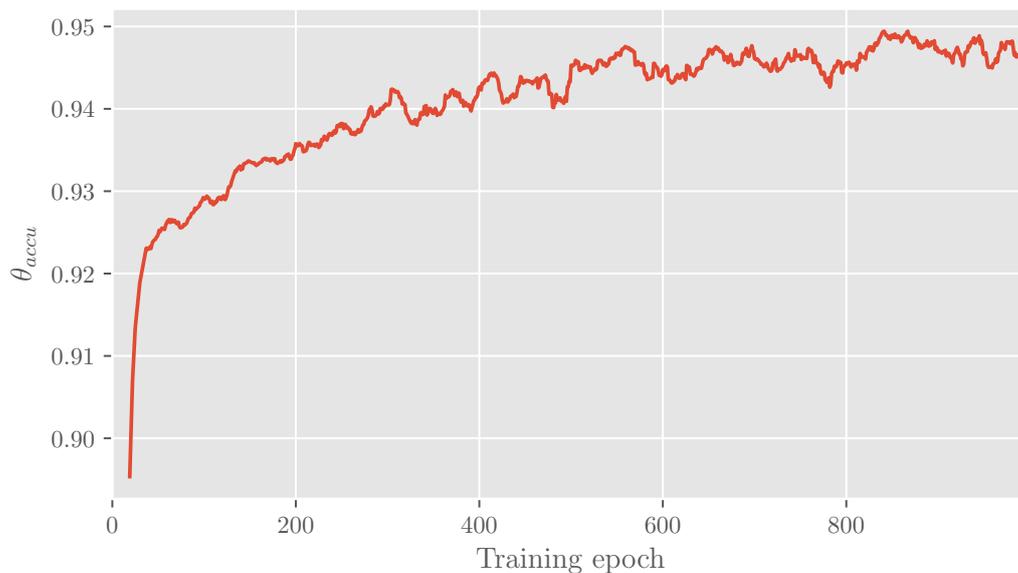


Figure 10. Prediction accuracy of the ANN model over 1000 training epochs.

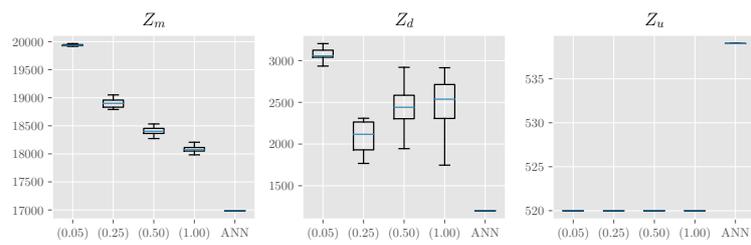
After 1000 iterations, the prediction accuracy on the test set gradually converges to a high region about 94.7%, and other evaluation metrics are listed in Table 5. These metric values demonstrate the effectiveness of the ANN-based completion status prediction model.

Table 5. Conventional evaluation metrics for the ANN.

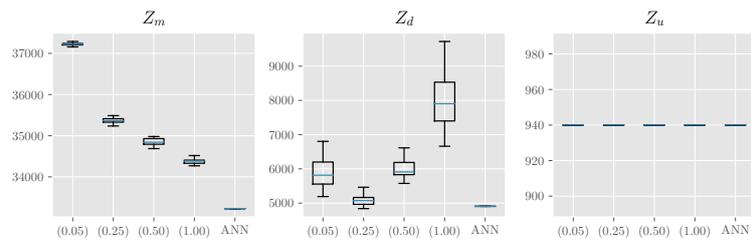
Label	Precision	Recall	F1-Score
Completion status 0	0.848	0.943	0.893
Completion status 1	0.848	0.700	0.767
Completion status 2	0.968	0.972	0.970
Macro average	0.888	0.872	0.877
Weighted average	0.944	0.944	0.943

4.3. Performance with Discussions

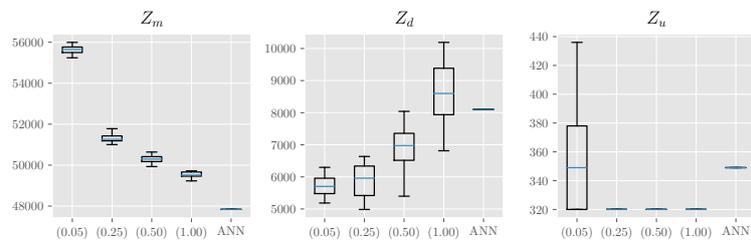
We use the modified NSGA-II (Nondominated Sorting Genetic Algorithm version II) [35] as the referred method for the comparative experiment. Specifically, the corresponding modification is based on the approximation that the whole problem is divided into the time dimension initially, then NSGA-II(α) is called to solve the divided sub-problems one by one. The problem segmentation factor $\alpha \in (0, 1]$ indicates the degree of the division, that $\alpha = 1$ means no division and $\alpha \rightarrow 0$ will mimic the effect of real-time scheduling. Figure 11a–e show the performance comparison of these methods in terms of total cost, service satisfaction, and make-span.



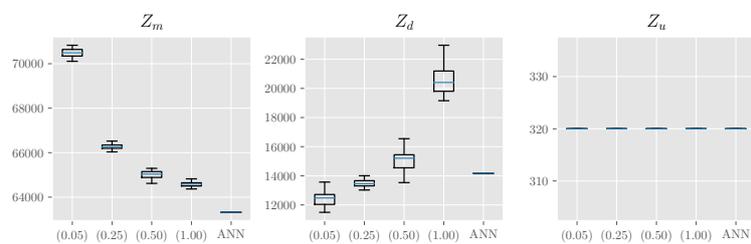
(a) Dataset Group Exp-30



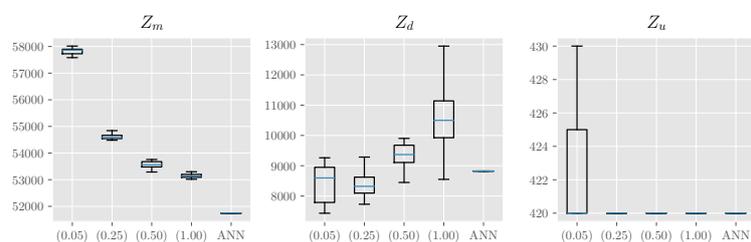
(b) Dataset Group Exp-60



(c) Dataset Group Exp-90



(d) Dataset Group Exp-120



(e) Dataset Group Exp-mix

Figure 11. Performance comparison on various MT Scales, (α) represents NSGA-II(α) in the x-labels, $\alpha \in \{0.05, 0.25, 0.50, 1.00\}$.

Since the dataset groups for these experiments mentioned in Section 4.1 are dimensionless quantities, we use comparison ratio in Equation (40) to evaluate the performance of the scheduling methods.

$$\rho_{obj} = \begin{cases} \frac{\overline{Z_{obj}}(ANN)}{\max_{\alpha} \left\{ \overline{Z_{obj}}(NSGA-II(\alpha)) \right\}} & \text{if } obj = u \\ \frac{\min_{\alpha} \left\{ \overline{Z_{obj}}(NSGA-II(\alpha)) \right\}}{\overline{Z_{obj}}(ANN)} & \text{otherwise} \end{cases}, \quad obj \in \{m, u, d\} \quad (40)$$

where $\overline{Z_{obj}}(\text{method})$ means the average of the objective value gained by the corresponding method, and the summarized experimental results are listed in Table 6.

Table 6. Summarized objective values result from different scheduling methods.

Dataset Group	Method	Average Cost (\overline{Z}_m)	Average Satisfaction (\overline{Z}_u)	Average Make-span (\overline{Z}_d)
Exp-30	NSGA-II(0.05)	19936.50	520.00	3067.65
	NSGA-II(0.25)	18,899.10	520.00	2085.15
	NSGA-II(0.50)	18,404.20	520.00	2467.45
	NSGA-II(1.00)	18,078.50	520.00	2475.50
	ANN	16,987.00	539.00	1201.00
		($\rho_m = 1.064$)	($\rho_u = 1.037$)	($\rho_d = 2.055$)
Exp-60	NSGA-II(0.05)	37,221.25	941.00	5881.85
	NSGA-II(0.25)	35,365.10	941.00	5057.35
	NSGA-II(0.50)	34,854.95	940.00	5942.30
	NSGA-II(1.00)	34,372.50	940.00	8026.00
	ANN	33,231.00	940.00	4049.00
		($\rho_m = 1.034$)	($\rho_u = 0.999$)	($\rho_d = 1.249$)
Exp-90	NSGA-II(0.05)	57,029.15	521.15	6237.05
	NSGA-II(0.25)	52,598.70	493.00	6168.35
	NSGA-II(0.50)	51,594.55	491.65	7100.70
	NSGA-II(1.00)	50,818.75	491.45	8702.70
	ANN	50,742.50	570.00	8563.50
		($\rho_m = 1.002$)	($\rho_u = 1.094$)	($\rho_d = 0.984$)
Exp-120	NSGA-II(0.05)	70,489.20	322.90	12,391.10
	NSGA-II(0.25)	65,013.05	320.00	15,160.20
	NSGA-II(0.50)	66,264.75	320.00	13,435.25
	NSGA-II(1.00)	64,586.50	320.00	20,547.70
	ANN	63,318.00	320.00	14,171.00
		($\rho_m = 1.020$)	($\rho_u = 0.991$)	($\rho_d = 0.874$)
Exp-mix	NSGA-II(0.05)	57,822.30	422.75	8406.95
	NSGA-II(0.25)	54,625.10	421.00	8344.35
	NSGA-II(0.50)	53,559.60	420.50	9380.85
	NSGA-II(1.00)	53,157.45	421.00	10,617.90
	ANN	51,743.00	420.00	8823.00
		($\rho_m = 1.027$)	($\rho_u = 0.993$)	($\rho_d = 0.946$)

In each group of the dataset, the ANN-based method has an absolute advantage in the cost (Z_d) compared to the NSGA-II(α) method series. Although the advantages of ANN in make-span (Z_d) and service satisfaction (Z_u) are not significant, this method has also reached higher levels in both criteria. For the NSGA-II(α) method series, as the value of division factor α increases, the number of sub-problems decreases, which leads to the decrease of cost, but the make-span increases. As for the service satisfaction, all of the methods perform similarly, which are mainly due to the limited number of candidate MRs in each MS. It can be inferred that the ANN-based approach has a strong generalization ability since it has a great performance in different dataset groups.

In addition to the objective values, another important indicator is the responsiveness that measures the decision time of each allocation procedure in the scheduling. Figure 12 summarizes the average responsiveness for each dataset group, the decision time for using the ANN to determine a schedule is only about 4.4% of the NSGA-II(α) on average.

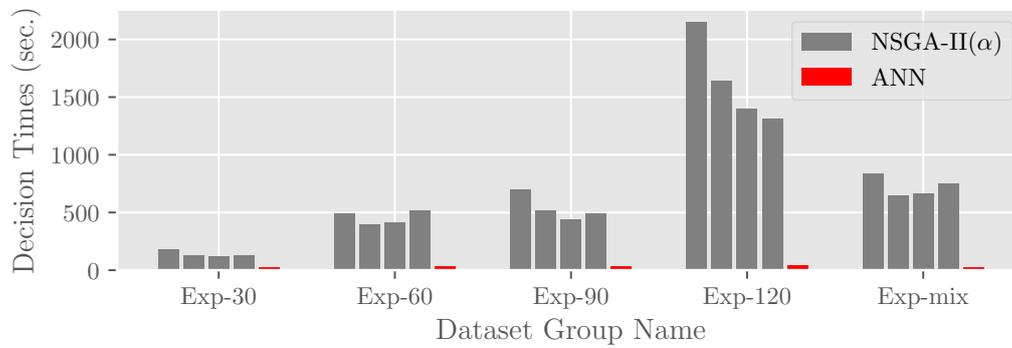


Figure 12. Average Responsiveness of NSGA-II(α) series and ANN.

As for the performance of the ANN-based approach on a single MR allocation for each MT, Table 7 summarizes the average decision time.

Table 7. Average decision time of the ANN-based approach.

Dataset Group	MT Amount	Average Schedule Time (s)	Average Decision Time for MT (ms)
Exp-30	600	25.147	41.911
Exp-60	1200	32.909	27.424
Exp-90	1800	32.113	17.840
Exp-120	2400	38.990	16.246
Exp-mix	1558	23.491	15.078

The average decision time for MR allocation is under 50 ms, and the average decision time for determining a schedule is under 40s, which indicates that the ANN-based approach is suitable for the real-time scheduling because, compared to the NSGA-II, the proposed ANN-based approach only takes 4.4% of the decision time to determine a sound schedule in such a discrete manufacturing environment, and the decision time is negligible if compared to the duration of resource configuration, manufacturing execution, transportation, and so on, which are usually measured in hours or days.

5. Application Design of ANN-Based Real-Time Scheduling in the Cloud Manufacturing Environment

Mold casting is a common industry that applies the CMfg. Figure 13 shows its demand and supply distribution in the Zhejiang province.

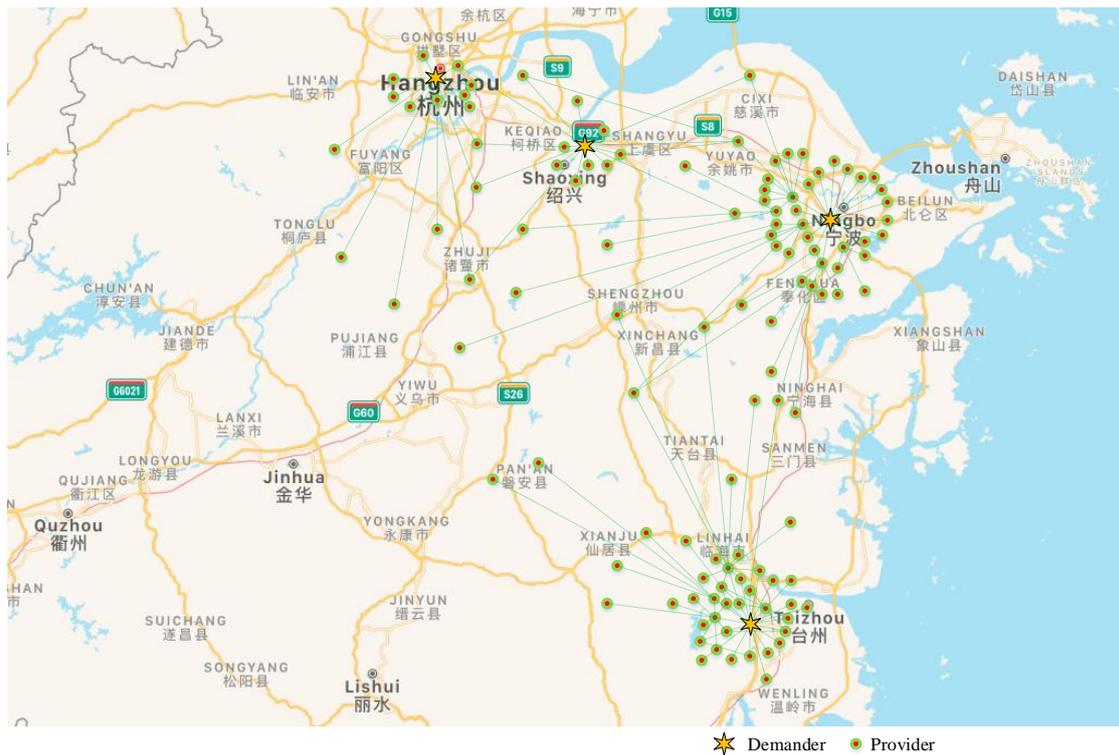


Figure 13. Provider and demander distribution in the CMfg environment.

It can be seen that the demanders are surrounded by numerous providers, and the resource optimization and coordination are in desperate need. With the increase in diversity and complexity of requirements from demanders, the real-time scheduling becomes much more difficult. Figure 14 shows the procedure for the demander to use MR via the CMfg platform.

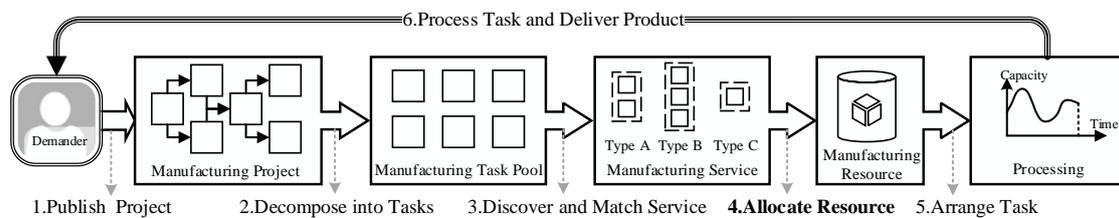


Figure 14. Procedure for demander to use MR via the CMfg platform.

The key step is the allocation of MR, which can be realized by the ANN. Figure 15 shows the functional design of the application of the proposed ANN-based approach in the CMfg environment.

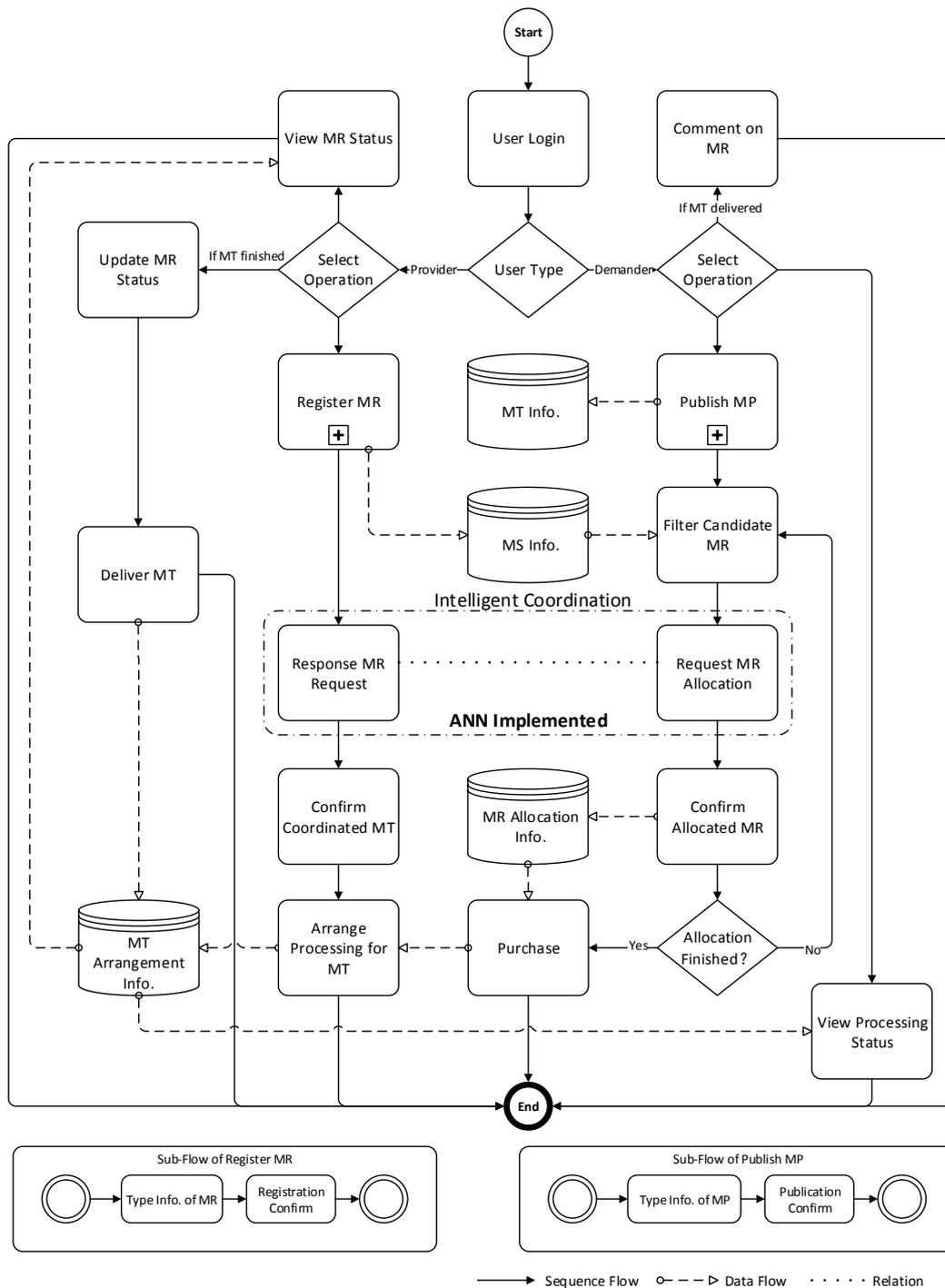


Figure 15. Logical flow chart of application design of ANN-based scheduling in the CMfg environment.

The “Intelligent coordination” module that implements the ANN will handle the real-time scheduling problem, and putting the design into practical application is an ongoing work.

6. Conclusions

In this paper, we studied the real-time scheduling problem in the CMfg environment. After presenting the premises and antecedents, several research contents and the results can be pointed out:

- We constructed a mathematical model for real-time scheduling in the CMfg environment with optimization objectives as minimizing the cost, minimizing the make-span, and maximizing the service satisfaction constrained by task precedence, resource occupation, and logistics duration;
- An ANN-based approach for real-time scheduling has been designed, it uses the MT attributes and process pending queue of MR as inputs to predict the completion status of MT if it will be allocated to any of the candidate MRs;
- We conducted the comparison experiments and modified the NSGA-II as the referred scheduling method, the results show that:
 1. The proposed ANN-based approach performs better than the NSGA-II in terms of objective values;
 2. The response time of the ANN is only about 4.4% of the NSGA-II on average;
 3. Using ANN, the average decision time for MR allocation is under 50 ms, which indicates that the proposed ANN-based approach is suitable for the real-time scheduling.
- We designed the application of ANN-based real-time scheduling to show how to implement this method to help coordinate providers of MR in the CMfg environment.

In addition, the proposed method has good performances on different MT scale datasets, which indicates its generalization ability. Since the proposed ANN-based approach solves the common problem for the platform based manufacturing environment, it can also be used in related web based manufacturing environments.

It should be noticed that the research in this article is mainly based on the library instances rather than the data from the real-world. Since there is no exactly the same real-time scheduling problem as described in this paper, the proposed ANN-based approach is not compared with other articles which also use ANN model to prove we have a better result. How to improve the ANN of the proposed real-time scheduling approach to perform better and apply this approach to the real world is our on-going work.

Author Contributions: Conceptualization, S.C. and S.F.; formal analysis, S.C.; funding acquisition, S.F.; methodology, S.C.; project administration, S.F. and R.T.; validation, S.C.; writing—original draft, S.C.; writing—review and editing, S.F. and R.T. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by 1. National Natural Science Foundation of China (grant number 71571161); 2. National High-tech Research and Development Program (grant number 2015AA042101); 3. Science Fund for Creative Research Groups of National Natural Science Foundation of China (grant number 51821093).

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Zhou, K.; Liu, T.; Zhou, L. Industry 4.0: Towards Future Industrial Opportunities and Challenges. In Proceedings of the 12th International Conference on Fuzzy Systems and Knowledge Discovery (FSKD), Zhangjiajie, China, 15–17 August 2015; pp. 2147–2152. doi:10.1109/FSKD.2015.7382284. [[CrossRef](#)]
2. Li, P.; Jiang, P.; Liu, J. Mini-MES: A Microservices-Based Apps System for Data Interconnecting and Production Controlling in Decentralized Manufacturing. *Appl. Sci.* **2019**, *9*, 3675. doi:10.3390/app9183675. [[CrossRef](#)]
3. Qu, Y.J.; Ming, X.G.; Liu, Z.W.; Zhang, X.Y.; Hou, Z.T. Smart Manufacturing Systems: State of the Art and Future Trends. *Int. J. Adv. Manuf. Technol.* **2019**, *103*, 3751–3768. doi:10.1007/s00170-019-03754-7. [[CrossRef](#)]
4. He, W.; Xu, L. A State-of-the-Art Survey of Cloud Manufacturing. *Int. J. Comput. Integr. Manuf.* **2015**, *28*, 239–250. [[CrossRef](#)]
5. Adamson, G.; Wang, L.; Holm, M.; Moore, P. Cloud Manufacturing—A Critical Review of Recent Development and Future Trends. *Int. J. Comput. Integr. Manuf.* **2017**, *30*, 347–380. [[CrossRef](#)]
6. Prinsloo, J.; Sinha, S.; von Solms, B. A Review of Industry 4.0 Manufacturing Process Security Risks. *Appl. Sci.* **2019**, *9*, 5105. doi:10.3390/app9235105. [[CrossRef](#)]

7. Nazarenko, A.A.; Sarraipa, J.; Camarinha-Matos, L.M.; Garcia, O.; Jardim-Goncalves, R. Semantic Data Management for a Virtual Factory Collaborative Environment. *Appl. Sci.* **2019**, *9*, 4936. doi:10.3390/app9224936. [[CrossRef](#)]
8. Liu, Y.; Xu, X. Industry 4.0 and Cloud Manufacturing: A Comparative Analysis. *J. Manuf. Sci. Eng.* **2017**, *139*, 034701. [[CrossRef](#)]
9. Tran, N.H.; Park, H.S.; Nguyen, Q.V.; Hoang, T.D. Development of a Smart Cyber-Physical Manufacturing System in the Industry 4.0 Context. *Appl. Sci.* **2019**, *9*, 3325. doi:10.3390/app9163325. [[CrossRef](#)]
10. Ren, L.; Zhang, L.; Wang, L.; Tao, F.; Chai, X. Cloud Manufacturing: Key Characteristics and Applications. *Int. J. Comput. Integr. Manuf.* **2017**, *30*, 501–515. doi:10.1080/0951192X.2014.902105. [[CrossRef](#)]
11. Tran, L.V.; Huynh, B.H.; Akhtar, H. Ant Colony Optimization Algorithm for Maintenance, Repair and Overhaul Scheduling Optimization in the Context of Industrie 4.0. *Appl. Sci.* **2019**, *9*, 4815. doi:10.3390/app9224815. [[CrossRef](#)]
12. Behnamian, J. Heterogeneous Networked Cooperative Scheduling With Anarchic Particle Swarm Optimization. *IEEE Trans. Eng. Manag.* **2017**, *64*, 166–178. [[CrossRef](#)]
13. Moghaddam, M.; Nof, S.Y. Collaborative Service-Component Integration in Cloud Manufacturing. *Int. J. Prod. Res.* **2018**, *56*, 677–691. [[CrossRef](#)]
14. Díaz-Reza, J.R.; Mendoza-Fong, J.R.; Blanco-Fernández, J.; Marmolejo-Saucedo, J.A.; García-Alcaraz, J.L. The Role of Advanced Manufacturing Technologies in Production Process Performance: A Causal Model. *Appl. Sci.* **2019**, *9*, 3741. doi:10.3390/app9183741. [[CrossRef](#)]
15. Xu, X. From Cloud Computing to Cloud Manufacturing. *Robot. Comput.-Integr. Manuf.* **2012**, *28*, 75–86. [[CrossRef](#)]
16. Wu, D.; Greer, M.J.; Rosen, D.W.; Schaefer, D. Cloud Manufacturing: Strategic Vision and State-of-the-Art. *J. Manuf. Syst.* **2013**, *32*, 564–579. [[CrossRef](#)]
17. Yang, G.; Chung, B.D.; Lee, S.J. Limited Search Space-Based Algorithm for Dual Resource Constrained Scheduling Problem with Multilevel Product Structure. *Appl. Sci.* **2019**, *9*, 4005. doi:10.3390/app9194005. [[CrossRef](#)]
18. Choi, Y.B.; Yun, H.Y.; yeop Kim, J.; Jin, S.H.; Kim, K.S. Robust Optimization Approach Using Scenario Concepts for Artillery Firing Scheduling Under Uncertainty. *Appl. Sci.* **2019**, *9*, 2811. doi:10.3390/app9142811. [[CrossRef](#)]
19. Boctor, F.F. Resource-Constrained Project Scheduling by Simulated Annealing. *Int. J. Prod. Res.* **1996**, *34*, 2335–2351. [[CrossRef](#)]
20. Nonobe, K.; Ibaraki, T. Formulation and Tabu Search Algorithm for the Resource Constrained Project Scheduling Problem. In *Essays and Surveys in Metaheuristics*; Ribeiro, C.C., Hansen, P., Eds.; Operations Research/Computer Science Interfaces Series; Springer: Boston, MA, USA, 2002; pp. 557–588.
21. Bouleimen, K.; Lecocq, H. A New Efficient Simulated Annealing Algorithm for the Resource-Constrained Project Scheduling Problem and Its Multiple Mode Version. *Eur. J. Oper. Res.* **2003**, *149*, 268–281. [[CrossRef](#)]
22. Mahdi Mobini, M.D.; Rabbani, M.; Amalnik, M.S.; Razmi, J.; Rahimi-Vahed, A.R. Using an Enhanced Scatter Search Algorithm for a Resource-Constrained Project Scheduling Problem. *Soft Comput.* **2009**, *13*, 597–610. [[CrossRef](#)]
23. Zamani, R. A Competitive Magnet-Based Genetic Algorithm for Solving the Resource-Constrained Project Scheduling Problem. *Eur. J. Oper. Res.* **2013**, *229*, 552–559. [[CrossRef](#)]
24. Chen, Z.; Demeulemeester, E.; Bai, S.; Guo, Y. Efficient Priority Rules for the Stochastic Resource-Constrained Project Scheduling Problem. *Eur. J. Oper. Res.* **2018**, *270*, 957–967. [[CrossRef](#)]
25. Vieira, G.E.; Herrmann, J.W.; Lin, E. Rescheduling Manufacturing Systems: A Framework of Strategies, Policies, and Methods. *J. Sched.* **2003**, *6*, 39–62. [[CrossRef](#)]
26. Lamas, P.; Demeulemeester, E. A Purely Proactive Scheduling Procedure for the Resource-Constrained Project Scheduling Problem with Stochastic Activity Durations. *J. Sched.* **2016**, *19*, 409–428. [[CrossRef](#)]
27. Davari, M.; Demeulemeester, E. The Proactive and Reactive Resource-Constrained Project Scheduling Problem. *J. Sched.* **2019**, *22*, 211–237. [[CrossRef](#)]
28. Herroelen, W.; Leus, R. Robust and Reactive Project Scheduling: A Review and Classification of Procedures. *Int. J. Prod. Res.* **2004**, *42*, 1599–1620. [[CrossRef](#)]
29. Akkan, C. Improving Schedule Stability in Single-Machine Rescheduling for New Operation Insertion. *Comput. Oper. Res.* **2015**, *64*, 198–209. [[CrossRef](#)]

30. Zhou, L.; Zhang, L.; Sarker, B.R.; Laili, Y.; Ren, L. An Event-Triggered Dynamic Scheduling Method for Randomly Arriving Tasks in Cloud Manufacturing. *Int. J. Comput. Integr. Manuf.* **2018**, *31*, 318–333. [[CrossRef](#)]
31. Davari, M.; Demeulemeester, E. Important Classes of Reactions for the Proactive and Reactive Resource-Constrained Project Scheduling Problem. *Ann. Oper. Res.* **2019**, *274*, 187–210. [[CrossRef](#)]
32. Arnaiz-González, Á.; Fernández-Valdivielso, A.; Bustillo, A.; López de Lacalle, L.N. Using Artificial Neural Networks for the Prediction of Dimensional Error on Inclined Surfaces Manufactured by Ball-End Milling. *Int. J. Adv. Manuf. Technol.* **2016**, *83*, 847–859. doi:10.1007/s00170-015-7543-y. [[CrossRef](#)]
33. Yuce, B.; Rezgui, Y.; Mourshed, M. ANN–GA Smart Appliance Scheduling for Optimised Energy Management in the Domestic Sector. *Energy Build.* **2016**, *111*, 311–325. doi:10.1016/j.enbuild.2015.11.017. [[CrossRef](#)]
34. Wang, L.; Guo, C.; Li, Y.; Du, B.; Guo, S. An Outsourcing Service Selection Method Using ANN and SFLA Algorithms for Cement Equipment Manufacturing Enterprises in Cloud Manufacturing. *J. Ambient Intell. Hum. Comput.* **2019**, *10*, 1065–1079. doi:10.1007/s12652-017-0612-3. [[CrossRef](#)]
35. Deb, K.; Pratap, A.; Agarwal, S.; Meyarivan, T. A Fast and Elitist Multiobjective Genetic Algorithm: NSGA-II. *IEEE Trans. Evol. Comput.* **2002**, *6*, 182–197. [[CrossRef](#)]



© 2020 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).