

Research Article

A Method for Efficient Task Assignment Based on the Satisfaction Degree of Knowledge

Jiafu Su ^{1,2}, Jie Wang ³, Sheng Liu,⁴ Na Zhang,⁴ and Chi Li ⁴

¹Research Center for Economy of Upper Reaches of the Yangtze River, Chongqing Technology and Business University, Chongqing, China

²National Research Base of Intelligent Manufacturing Service, Chongqing Technology and Business University, Chongqing, China

³Shanghai Aerospace Equipments Manufacturer Co., Ltd, Shanghai, China

⁴College of Mechanical Engineering, Chongqing University, Chongqing, China

Correspondence should be addressed to Chi Li; lichimsor@gmail.com

Received 3 July 2019; Revised 11 June 2020; Accepted 13 July 2020; Published 3 September 2020

Academic Editor: Alex Alexandridis

Copyright © 2020 Jiafu Su et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

For the product R&D process, it is a challenge to effectively and reasonably assign tasks and estimate their execution time. This paper develops a method system for efficient task assignment in product R&D. The method system consists of three components: similar tasks identification, tasks' execution time calculation, and task assignment model. The similar tasks identification component entails the retrieval of a similar task model to identify similar tasks. From the knowledge-based view, the tasks' execution time calculation component uses the BP neural network to predict tasks' execution time according to the previous similar tasks and the Task-Knowledge-Person (TKP) network. When constructing the BP neural network, the satisfaction degree of knowledge and the execution time are set as the input and output, respectively. Considering the uncertain factors associated with the whole R&D process, the task assignment model component serves as a robust optimization model to assign tasks. Then, an improved genetic algorithm is developed to solve the task assignment model. Finally, the results of numerical experiment are reported to validate the effectiveness of the proposed methods.

1. Introduction

Nowadays, manufacturing industries are facing challenges arising from continuous innovations, complex competition, and collaboration environment [1, 2]. “Advanced Manufacturing National Strategic Plan,” “Industry 4.0 Manufacturing,” and “Manufacturing 2025” all assert that the product Research and Development (R&D) is the most important competitive advantage of a country's manufacturing industry. Due to the complexity of R&D, product R&D needs the collaborations among teams and employees with different knowledge fields and organizational backgrounds [3, 4]. For the effective collaborations, task assignment is the primary issue [5]. According to Alidaee et al. [6], task assignment refers to establishing correspondences between the set of tasks and the set of organizational units. Task assignment is one of the most important works in product R&D, which has direct

effects on the operating efficiency of product R&D. A product R&D project always involves hundreds of tasks and the dependences among tasks. The efficient task assignment can reduce the design lead time and costs, while maintaining the quality of product [1]. Thus, the efficient task assignment is a key in the process of product R&D to achieve the operational competitive advantages.

Task assignment is a decision problem of working process when the workloads and responsibilities about tasks are assigned among different organizational units, which will perform their own parts of a common project [7]. This paper aims to explore a type of task assignment problem under uncertainty. The problem is described as follows. There are n tasks $T = (T_1, T_2, T_3, \dots, T_n)$ that need to be assigned to m R&D teams $P = (P_1, P_2, P_3, \dots, P_m)$ during the process of product R&D. The objective is to minimize the final completion time of all tasks. The uncertainty means that tasks' theoretical

execution time is unknown or inaccurate, and the fluctuation of execution time is also uncertain in the future. For the above problem, this paper studies how to develop an efficient task assignment method, so as to minimize the final completion time of the task assignment plan when facing the uncertainty.

For the decision of task assignment, the decision-makers should not only consider the theoretical execution time of tasks, but also incorporate the anticipation of uncertainty during the execution of task assignment plan. Dealing with the uncertainty can make task assignment plan with more robustness and stability [8]. Furthermore, in the knowledge economy age, knowledge is an important resource of product R&D [9, 10]. Thus, the knowledge and ability of teams are the key factors for task execution. Each team has certain knowledge and capabilities, and each task requires a specific knowledge and capability that is possessed by at least one of the teams. This implies that the match degree between task's knowledge requirement and teams' knowledge plays a vital role for the efficient task assignment. In view of this, this paper proposed a method system for efficient task assignment considering the satisfaction degree between task and knowledge. The method system includes three components, which are the similar tasks identification, tasks' execution time calculation, and task assignment model. Firstly, the effective set of similar tasks is constructed based on the satisfaction degree of knowledge for tasks. Then, a BP neural network is built to predict the theoretical execution time for tasks based on a Knowledge-Task-Person (T-K-P) network. When it comes to assigning tasks, a robust task assignment model is proposed, which is applicable to situations with limited information about fluctuations of the execution time. The main contributions of this paper can be summarized as follows. Firstly, a knowledge-based method is proposed to predict the tasks' execution time, which can overcome the shortcoming of the ambiguity and subjectivity of the manager's decision. Secondly, a robust optimization model and a corresponding improved genetic algorithm are developed to settle the task assignment problem under uncertainty, which can protect decision-makers from the parameter ambiguity and stochastic uncertainty. Finally, a comprehensive method system for task assignment that incorporates knowledge and uncertainty is developed to make more effective and robust task assignment plans.

The remainder of this paper is organized as follows. Section 2 reviews related works about task assignment problem. Section 3 presents the efficient task assignment method system. In Section 4, an improved genetic algorithm is developed to solve the task assignment model. Section 5 describes the applications of the proposed methods and reports the numerical results of a real task assignment problem. The conclusion and future work are presented in Section 6.

2. Related Works

This paper focuses on the problem of task assignment considering knowledge satisfaction degree under uncertainty; thus, the works about task assignment, knowledge-based systems, uncertainty, and robust optimization methods are most relevant.

The problem of task assignment is to assign tasks to a set of organizational units based on the premise that a project schedule is given. Guo et al. [11] presented a Best-Harmonically-Fit task assignment algorithm on multiple periodic resources to assign tasks by establishing the worst-case execution time. Aiming at the tree-structured task allocation problem (TSTAP), Liu et al. [12] formalized TSTAP with group multirole assignment (GMRA), and then obtained the more effective and efficient CPLEX solutions, which provided a viable and effective decision support to solve the complex TSTA problems. Zhu et al. [13] clarified the group role assignment problem (GRAP) and proposed an efficient algorithm to solve it based on the Kuhn–Munkres algorithm, which offers efficient enough solutions for the GRAP. Furthermore, Zhu et al. [14] formalized staff and task assignment problems on the basis of the environment by adapting the process to a GRA and proposing a novel algorithm based on the Kuhn–Munkres algorithm. Yi et al. [15] proposed a reliability-guaranteed task assignment and scheduling approach for heterogeneous multiprocessors, which included a timing constraint. Mendonça et al. [16] proposed a simple yet efficient distributed control algorithm to implement dynamic task allocation in a robotic swarm. However, they did not account for tasks' execution time. Moreover, in the above works', the task assignment models all assume that the task' execution time is available. Obviously, it cannot reflect all the real situations of task assignment problem, especially for the complex product R&D project under uncertainty. Considering that the duration of a task carried out by different employees varies, Cheng and Chu [17] used the fuzzy system theory to represent the duration of each task by fuzzy numbers. Zhang and Su [18] used fuzzy variables to describe processing time by introducing a fuzzy triangular number, and they pointed that different teams vary in their knowledge and abilities, so the execution time is different for each team when they carry out each task. For the complex industrial manufacturing process, Li et al. [19] solved the task assignment optimization problem by establishing a dynamic process model and developing an improved quantum genetic algorithm with a heuristic principle.

To better manage the schedule of product R&D project, the execution time of a task always needs to be estimated and obtained in advance. Traditionally, the execution time of task is determined by the project managers, who mainly make decisions by their subjective experience. To avoid the limitations of subjective experience in decision of task execution time, this work proposes a method to accurately calculate tasks' execution time using knowledge-based systems. Nowadays, knowledge has been gradually considered as a significant resource of task execution to deal with more complex product R&D [20, 21]. Against this background, knowledge-based systems (KBSs) are increasingly used to help managers in their decision-making [22–24]. Roy et al. [25] presented a framework for optimizing task assignment in knowledge-intensive crowdsourcing and proposed an efficient adaptive algorithm to settle the task assignment problem. Lyu et al. [26] proposed a task assignment method based on fuzzy set theory and knowledge similarity, where

each work unit's knowledge is extracted from knowledge-based systems. Considering the issue that knowledge is not equal in task assignment of product development, Youling et al. [27] proposed a task assignment model based on bilateral matching of task and designer's knowledge. In the above works, their task assignment decision all considered the match between task and knowledge, and provided useful approaches for task assignment from the view of knowledge base. Each stage in the product R&D requires certain knowledge, which can be effectively captured, modeled, and reused by knowledge-based systems to improve the quality of product R&D [28, 29]. Furthermore, the consideration of knowledge in task assignment problem is essential to enhance the efficiency and reliability of task execution, which can be used to estimate task's execution time. Inspired by the above works, this paper plans to acquire tasks' execution time regarding knowledge as a key factor.

Uncertainties of task assignment arise naturally in many scenarios. For example, companies in industries of electronic products, software, and engineering services that do task-based work. Facing the increasingly complex product R&D project, it is common for differences between the plan and practice, and uncertainty is an important characteristic which exists throughout the R&D project [30]. Thus, the decision-makers must consider uncertainties when assigning tasks. Taking uncertainties into account can be done in two ways: chance-constrained optimization and robust optimization [7], which have been widely applied in the task assignment problems. Padulo and Liou [31] proposed a chance-constrained optimization method using min-max objectives and constraints. Zhen [7] proposed a stochastic programming model that copes with the arbitrary probability distributions of tasks' random workload requirements, as well as a robust optimization model that is applicable to situations in which there is limited information available about probability distributions. Quintana-Amate et al. [28] introduced a green and robust optimization scheme based on the hose model by formulating the problem as a mixed-integer linear programming (MILP) problem. Robust optimization models can provide insights into the optimal solution and probabilistic guarantees of constraint violation [32, 33]. As a pioneer work, Ben-Tal and Nemirovski [34] developed the robust solutions (min-max/max-min objective) by integrating uncertainty sets and proposed an efficient algorithm to solve the convex optimization problems under data uncertainty. Equally important, Bertsimas and Sim [35] proposed a classic approach for robust linear optimization to reduce the price of robustness. Moreover, their work can offer full control on the degree of conservatism for constraints as data change. Following Bertsimas and Sim's work, Büsing and D'Andreagiovanni [36] proposed a multiband uncertainty set with histogram cardinality-constraints in robust optimization, and their work formulated the problem of separating a robustness cut as a min-cost flow problem which can be solved efficiently. Assava-peek et al. [37] presented a three-stage optimization algorithm to solve two-stage deviation robust decision-making problems under uncertainty. Carello and Lanzarone [38] used the robust optimization method to build a

cardinality-constrained robust nurse-to-patient assignment model, which did not need to generate scenarios. Chen et al. [39] formulated a robust optimization model to solve the multiple unmanned aerial vehicles task assignment problem with parameter uncertainty. Golpra and Tirkolaei [8] introduced the robust optimization concept into the maintenance and repair tasks scheduling problem and developed a novel robust Bi-Objective Mixed-Integer Linear Programming model to deal with the uncertainty of the tasks duration.

3. The Efficient Task Assignment Method System

The proposed method system for efficient task assignment (ETA) is shown in Figure 1. The ETA system contains three components, which are similar tasks identification, tasks' execution time calculation, and task assignment model. Firstly, the ETA system starts with the identification of similar tasks according to the knowledge base. Once the similar tasks are identified, the satisfaction degree of knowledge can be determined by the historical data of previous tasks. On the basis of satisfaction degree of knowledge, tasks' execution time can be calculated using the BP neural network. Then, a robust task assignment model considering uncertainty and a genetic algorithm are proposed to obtain the optimal task assignment scheme for the product R&D. The following sections will discuss the three components of the ETA system in detail:

3.1. Similar Tasks Identification. Traditionally, the execution time of tasks is determined by project managers who rely on their subjective experience, which will inevitably bring subjective limitations. The method proposed in this paper accounts for the similarity between a previous task and the current task, and it estimates the current task's execution time using the information from similar historical tasks in knowledge base [40]. The knowledge base contains the execution time, the main needed knowledge, and the satisfaction degree of knowledge for each task. The similar tasks in the knowledge base can be identified by the Retrieving Similar Task (RST) model. Specifically, retrieving similar tasks is an important part of the process of estimating tasks' execution time. Firstly, this work gets a set of previous tasks for the current task from the knowledge base and then calculates the similarity degree between each previous task and the current task and finally puts the task into the set of similar tasks when its similarity degree is greater than the threshold value. The specific steps of retrieving similar tasks are shown as follows:

Step 1: Use relevant keywords to retrieve previous tasks that are similar to the current task according to the knowledge base, and then establish a set of previous tasks.

Step 2: For each task T'_i in the previous task set, calculate its similarity to the current task T_i using the following equation:

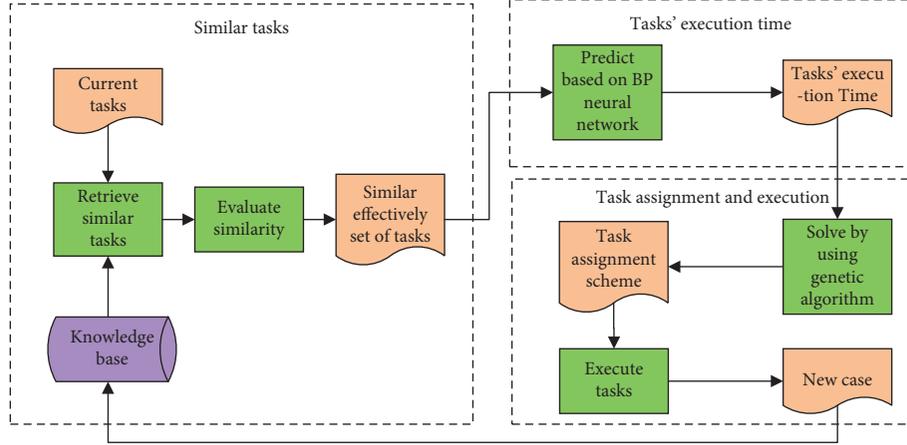


FIGURE 1: The efficient task assignment method system for product R&D.

$$\begin{aligned} \text{sim}(T_i, T'_i) = & \phi \times \text{knowledgesim}(T_i, T'_i) + \varphi \\ & \times \text{workloadsim}(T_i, T'_i), \end{aligned} \quad (1)$$

where knowledgesim and workloadsim are the functions to calculate the similarity in knowledge and workload, respectively. ϕ and φ denote the weights of knowledgesim and workloadsim respectively, and $\phi + \varphi = 1$.

$$\begin{aligned} \text{knowledgesim} &= \sqrt{\prod_{j=1}^n p_j}, \\ \text{workloadsim} &= \begin{cases} e^{-|wl-1|}, & 0.5 < wl < 1.5, \\ 0, & \text{otherwise,} \end{cases} \end{aligned} \quad (2)$$

where p_j denotes the similarity value on the j -th knowledge between the previous task and current task; wl denotes the relative ratio value of the current task's workload compared with the previous task's workload. The fuzzy evaluation sets of p_j and wl are listed in Table 1.

Step 3: Set the threshold value η . When $\text{sim}(T_i, T'_i) \geq \eta$, we put the task into the set of similar tasks of U_i ; otherwise, the task is excluded from the similar tasks set.

3.2. Tasks' Execution Time Calculation. To calculate tasks' execution time, Li et al. [41] proposed a knowledge-task network to describe the relationships between tasks and their associated knowledge. For most tasks, their knowledge requirements can be determined from the task description that is written in the project management manual. Tasks' execution time is associated with the knowledge structure of the R&D team. Each team has different knowledge, skills, and abilities for the R&D task. In order to calculate tasks' execution time by certain team more accurately, a method is proposed to incorporate the similarity between the knowledge structure of the R&D team and the knowledge needed by the task using the BP neural network.

3.2.1. Constructing T-K-P Network. The T-K-P network (see Figure 2) connects task, knowledge, and team. Each task has its required knowledge, and each team has its acquired knowledge. Suppose that all team are defined by the set of $P = \{P_1, P_2, \dots, P_m\}$, all tasks are defined by the set of $T = \{T_1, T_2, \dots, T_n\}$, and the knowledge are defined by the set of $K = \{K_1, K_2, \dots, K_h\}$. Based on the above network, we assume S_{kai} represents the satisfaction degree of knowledge K_α of team P_k for task T_i .

Then, we need to quantify the satisfaction degree of knowledge, and the value of S_{kai} is rated by the R&D team members using fuzzy evaluation. The fuzzy evaluation sets are listed in Table 2.

3.2.2. Calculating Tasks' Execution Times. The tasks' execution times are determined by the satisfaction degree of knowledge. Although the relationships between the satisfaction degree of knowledge and tasks' execution time are uncertain, the neural network method can be used to adjust their relationships to adapt to external changes. When it comes to a changing environment, the neural network is robust. So, the BP neural network method is used to predict tasks' execution times in this work. The BP neural network (see Figure 3) contains an input layer, a hidden layer, and an output layer, and we selected a single hidden layer in this work. The input is the satisfaction degree of knowledge, and the output is the task's execution time.

The determination process of tasks' execution times using BP neural network can be divided by the following specific procedures:

- (a) *Constructing the BP Neural Network.* For the task assignment between task T_i and team P_k , the input of the BP neural network is knowledge satisfaction degree S_{kai} , and the output is task's execution time t_{ik} . In this work, for n_i input data and one output data, we set the structure of the neural network as $[n_i - (2n_i + 1) - 1]$ according to Hecht Nielsen rule. Moreover, considering the prediction error of different transfer functions, the *Logsig* function and the

TABLE 1: Fuzzy evaluation sets of p_j and w_l .

Similarity of knowledge	Quantized value	Relative workload	Quantized value
Best	0.9	More	>1.5
Good	0.7	A lot	1.25
Equal	0.5	Equal	1
Bad	0.3	Little	0.75
Worst	0.1	Less	<0.5

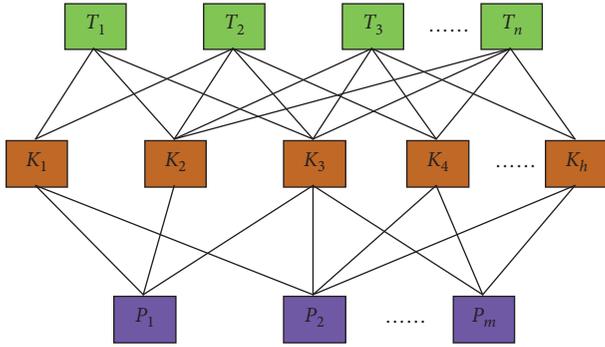


FIGURE 2: The T-K-P network.

TABLE 2: The fuzzy evaluation sets of satisfaction degree of knowledge.

Satisfaction of knowledge	Quantized value
Best	0.9
Good	0.7
Equal	0.5
Bad	0.3
Worst	0.1

Purelin function are used as the activation functions for the hidden layer and output layer, respectively.

- (b) *Training the BP Neural Network and Predicting Tasks' Execution Time.* First, we select a number set of data for training and select the *Trainscg* function as the training function. Because the BP neural network falls into the local minimum when trained in batches, we added a momentum factor that feeds back the error signal, making the weights of neurons reoscillate. Finally, we choose the function of *Sim* to predict tasks' execution time.

3.3. Task Assignment Model

3.3.1. *Notations.* For ease of reading, Table 3 lists the notations which are used in the task assignment model.

3.3.2. *The Task Assignment Model.* In this work, we assume that each task is assigned to one team, and each team can be assigned with no tasks or several tasks. Moreover, each team can do no more than one task at once, and the tasks are carried out continuously. For the task assignment model, it has one objective, which is minimizing the final completion

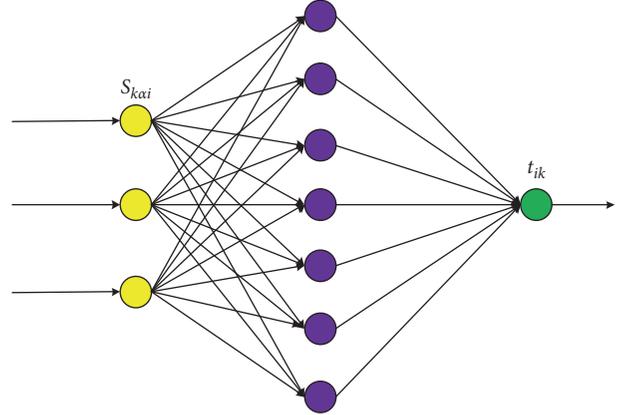


FIGURE 3: The topology diagram of BP neural network.

time. The mathematical model of the task assignment problem in this paper is described below.

Objective function:

$$\min(\text{MAX}_i f_i). \quad (3)$$

Subject to the following constraints:

$$r_{ij} = 0, \quad \text{or } 1, \quad \forall i, j, \quad (4)$$

$$x_{ik} = 0, \quad \text{or } 1, \quad \forall i, k, \quad (5)$$

$$\sum_{k=1}^m x_{ik} = 1, \quad \forall i, \quad (6)$$

$$f_i = s_i + \sum_{k=1}^m (x_{ik} \times t_{ik}), \quad \forall i, \quad (7)$$

$$s_j \geq \max(r_{ij} f_i), \quad \forall i, j, \quad (8)$$

$$d_{ik\beta} = 0, \quad \text{or } 1, \quad \forall i, k, \beta, \quad (9)$$

$$\sum_{i=1}^m d_{ik\beta} \leq 1, \quad \forall k, \beta, \quad (10)$$

$$E_{k\beta} \geq E_{k(\beta-1)} + t_{ik}, d_{ik\beta} = 1, \quad \forall k, \beta, i, \quad (11)$$

TABLE 3: The notations.

T_i, T_j	The i th and j th task, respectively
P_k	The k th team
K_α	The α th knowledge
f_i	The completion time of task T_i
$\text{MAX}_i f_i$	The final completion time of all tasks
r_{ij}	A binary variable denoting the order task T_i and T_j
x_{ik}	A binary variable denoting whether task T_i is assigned to team P_k
s_i, s_j	The start time of task T_i and T_j , respectively
t_{ik}	The practical execution time of T_i carried out by P_k
t'_{ik}	The theoretical execution time of T_i carried out by P_k
$d_{ik\beta}$	A binary variable denoting whether T_i is the β -th task that P_k can perform
$E_{k\beta}$	The completion time of the β -th task by P_k
ε	The random fluctuation variable
ξ	The disturbance variable
κ	The probability of violating a constraint

$$E_{k\beta} = f_i, d_{ik\beta} = 1, \quad \forall k, \beta, i, \quad (12)$$

$$E_{k0} = 0, \quad \forall k. \quad (13)$$

The objective (3) is to minimize the final completion time, where f_i denotes the completion time of T_i . Constraint (4) ensures that the tasks have a certain order in the R&D process. The order is indicated by r_{ij} , and r_{ij} is a binary variable. If task T_j can begin after the completion of task T_i , $r_{ij} = 1$; otherwise, $r_{ij} = 0$. Constraints (5) and (6) ensure that a task T_i should be implemented by one team P_k during the R&D process. x_{ik} is a binary variable. If T_i is assigned to P_k , $x_{ik} = 1$; otherwise, $x_{ik} = 0$. Constraint (7) indicates that the finish time of T_i equals the start time s_i plus the practical execution time t_{ik} . Constraint (8) denotes that task T_j can begin after T_i is finished, and s_j denotes the start time of T_j . Constraints (9) and (10) ensure that a team P_k can only perform one task at any moment. $d_{ik\beta}$ is a binary variable. If T_i is the β -th task that P_k can perform, $d_{ik\beta} = 1$; otherwise, $d_{ik\beta} = 0$. Constraint (11) ensures that the β th task can be carried out by team P_k only after finishing the $(\beta - 1)$ th task. Constraint (12) denotes that the completion time of the task equals to the time when the team have finished the task. Constraint (13) ensures that all teams are available at the initial time.

In the process of task assignment for product R&D, there are some interference factors affecting tasks' execution time such as urgent order inserting, task changes, and inadequate capability of team. These uncertainty factors can interfere with the execution of a task and cause deviations in the tasks' execution time between the plan and the practice. Among these uncertain factors, there are factors that change regularly. Moreover, by investigating these factors' changing regularity in task assignment, it can enhance the immunity to uncertain interference. Specifically, considering the execution time of task is an uncertain value and referencing the literature of Li and Ierapetritou [42], the relationship between the practical execution time t_{ik} and theoretical execution time t'_{ik} can be described by the random fluctuation variable ε and the disturbance variable ξ , which can be denoted as $t_{ik} = (1 + \varepsilon\xi)t'_{ik}$. In this work, it is assumed that the disturbance variable ξ is consistent with a uniform distribution on $[-1, 1]$, and its distribution function can be denoted as

$$F(\xi) = \frac{\xi - (-1)}{1 - (-1)} = \frac{\xi + 1}{2}, \quad \xi = 2F(\xi) - 1. \quad (14)$$

Then, suppose κ denotes the probability of violating a constraint, there is

$$F(\xi) = 1 - \kappa, \quad \xi = 1 - 2\kappa. \quad (15)$$

Finally, the practical execution time t_{ik} can be obtained as

$$t_{ik} = [1 + \varepsilon(1 - 2\kappa)]t'_{ik}. \quad (16)$$

4. The Improved Genetic Algorithm for Task Assignment Model

In this work, an improved genetic algorithm is proposed to settle the task assignment model. Genetic algorithm (GA) simulates the evolution rules for the survival of the fittest in the biological world. It can achieve global optimization through an efficient parallel search in a complex solution space, and the coding is simple and easy to operate [43–45]. For these reasons, we use GA to solve the task assignment problem. However, weak local search ability and early convergence are the two main defects of GA. Therefore, GA often needs to be improved for better search ability. Considering the fluctuations in the execution time during task assignment, this paper improves GA by introducing the binary set encoding and decoding mechanism of greedy algorithm, along with the improved fitness function to avoid local optimization and solve the model rapidly. The specific steps of the improved genetic algorithm for task assignment model in this work are shown in the following.

4.1. Chromosome Coding. As the input of the GA, one solution of the task assignment problem is one chromosome, which consists of a number of genes. In this chromosome, a gene is denoted by one tuple (T_i, P_k) , which indicates that task T_i is assigned to team P_k . Thus, the two-dimensional array $S[n][2] = [(T_{x1}, P_{y1}), (T_{x2}, P_{y2}), \dots, (T_{xn}, P_{yn})]$ is used to represent a chromosome.

4.2. Initial Population. The proposed method randomly generates the initial population based on the depth of the task. The population produced by this method can satisfy the constraint relationship among tasks and improve the efficiency of the algorithm. $T = (T_1, T_2, T_3, \dots, T_n)$ represents the set of tasks; R represents the structure matrix, which reflects the constraint relationship among tasks; $P = (P_1, P_2, P_3, \dots, P_m)$ represents the set of R&D teams; and S represents the assignment scheme. The steps of generating the initial population are shown as follows:

- (i) Step 1: Calculate each task's depth in set T according to the structure matrix $R = [r_{ij}]_{n \times n}$ and get the tasks' depth set $D = \{D_1, D_2, D_3, \dots, D_n\}$.
- (ii) Step 2: Sequence the tasks according to their depth to achieve the task sequence denoted by $\{T_{x1}, T_{x2}, \dots, T_{xn}\}$.
- (iii) Step 3: Assign a team to each task by randomly selecting from $P = (P_1, P_2, P_3, \dots, P_m)$ to ultimately get the task assignment scheme $S' [n] [2] = [(T'_{x1}, P'_{y1}), (T'_{x2}, P'_{y2}), \dots, (T'_{xn}, P'_{yn})]$.
- (iv) Step 4: Repeat steps 2–3 n times to obtain the initial population.

4.3. The Decoding Mechanism. Decoding is a process that transforms chromosomes into solutions. We regard the first dimension of the two-dimensional array as an ordered sequence and decode them in accordance with the sequence. The specific implementation processes are described as follows: there are two prerequisites that should be met for P_k to begin the execution of T_j . First, the tasks that precede T_j , that is task T_i , have been completed. Second, P_k is available. The plug-greedy algorithm is used to obtain the best possible start time with active scheduling [46]. Specifically, the best start time is the earliest start time under two premises, they are, the pretask T_i of T_j is done, and the team P_k is available. The best possible start time s_j can be denoted as

$$s_j = \max(f_i \times r_{ij}, E_{k(\beta-1)}). \quad (17)$$

4.4. The Fitness Function. Suppose that T_j is assigned to P_k , and it is operated in accordance with the best possible start time. Taking the random fluctuations in the execution time into account, the actual execution time can be denoted as $t_{jk} = [1 + \varepsilon_{jk}(1 - 2\kappa_{jk})]t'_{ik}$. Thus, the finish time f_j can be obtained by the following equation:

$$f_j = \max(f_i \times r_{ij}, E_{k(\beta-1)}) + [1 + \varepsilon_{jk}(1 - 2\kappa_{jk})]t'_{ik}. \quad (18)$$

In this work, the minimum of the final completion time of all tasks is regard as the objective of task assignment model; thus, the fitness function of chromosomes X can be defined as

$$\text{fitness}(X) = \frac{1}{\max\{f_j\}}. \quad (19)$$

4.5. The Genetic Manipulations. In order to achieve the evolution of a population after generating the initial population, this work needs to implement the genetic manipulations, which can be divided into three steps, which are selection, crossover, and mutation.

- (a) **Selection.** The roulette strategy is used as the selection strategy. The roulette strategy determines the selection probability for each individual according to its fitness; the shorter the completion time, the higher the individual fitness, and the larger the probability of being selected. In this way, it is conducive to the inheritance of excellent genetic individuals to the next generation, and it also contributes to the rapid convergence of the genetic algorithm.
- (b) **Crossover.** The single-point crossover mode is used in this work. The crossover aims to generate a new task assignment scheme from the known schemes. For task assignment schemes S_i and S_j , the crossover steps are as follows:
 - (i) First, we generate a random integer p ($0 \leq p \leq n$) and divide both schemes S_i and S_j into two halves, where p is the cross-point.
 - (ii) Second, we crossover the R&D team of the latter part of S_i and S_j , and then generate new individuals S'_i and S'_j , which keeps the task sequence unchanged.
- (c) **Mutation.** Mutation aims to maintain the diversity of the population to prevent premature convergence of the genetic algorithm. The steps of mutation are as listed follows:
 - (i) First, we select an individual to mutate using the predefined mutation probability.
 - (ii) Second, we generate a random integer p' ($0 \leq p' \leq n$) that serves as the mutation position. At this position, the R&D team is reassigned by the roulette approach.

The pseudocode and framework of the improved GA is given as follows.

5. Numerical Experiments

In this section, we present a real case on product R&D task assignment decision for a partial segment of diesel fuel injection system, one of the R&D projects which are led by CA Technology Co., Ltd. It is adopted to illustrate the proposed methods in this paper. The project consists of 11 tasks $T = (T_1, T_2, T_3, \dots, T_{11})$, and there are 5 R&D teams $P = (P_1, P_2, P_3, P_4, P_5)$ available for the tasks. The task sequence chart is shown in Figure 4.

5.1. Experiments on Retrieving Similar Task. Suppose that $\phi = 0.5$, $\phi = 0.5$, and $\eta = 0.8$. For task T_1 , we found 20 tasks that are similar with T_1 from the knowledge base, and calculated their similarities to T_1 using the proposed

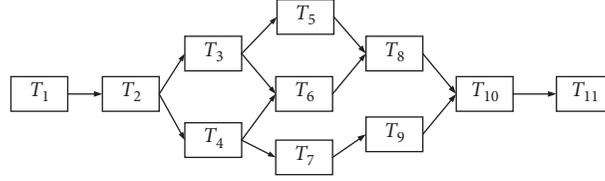


FIGURE 4: The task sequence chart.

```

Initialize parameters which include population size PopsiZe, selection rate  $P_s$ , crossover rate  $P_c$ , mutation rate  $P_m$ , and the maximum number of iterations  $Maxgen$ .
Generate the initial population randomly.
Calculate each task's depth  $D = (D_1, D_2, D_3, \dots, D_n)$ ,
  for  $j$  from 1 to  $n$  do
    Sequence tasks according to tasks' depth, and achieve the task sequence  $\{T_{x1}, T_{x2}, \dots, T_{xn}\}$ 
    Assign the team to each task, and achieve the chromosome  $Pop\{j\}$ 
  end for
while current generation  $Cg < Maxgen$  do
  for  $j$  from 1 to  $n$  do
    Decode it by greedy algorithm and calculate its completion time  $f_j$ 
    Set the fitness function as  $fitness(X) = 1 / \max_j \{f_j\}$ 
  end for
  Rank the individuals' fitness value, get the max (fitness) value Val
  Save the result as the best solution of current generation  $yy (Cg) = Val$ 
  for  $j$  from 1 to  $n$  do
    Select individual by roulette wheel selection
    Crossover individual by one-cut point crossover
    Mutation individual
    for  $j$  from 1 to  $n$  do
      if  $r$  and  $(0, 1) < Pm$  then
        Generate mutation position  $p' (0 \leq p' \leq n)$ 
        Reset individual at position  $p'$ , generate new offspring
         $j \leftarrow j + 1$ 
      end if
    end for
  end while
Output the best solution
  
```

ALGORITHM 1: The pseudo code and framework of the improved GA.

TABLE 4: The effective similar task set for T_1 .

Task	S_{i1k}	S_{i2k}	S_{i3k}	S_{i4k}	Execution time
$T_{1,1}$	0.5	0.7	0.6	0.6	60
$T_{1,2}$	0.7	0.8	0.5	0.5	51
$T_{1,3}$	0.9	0.9	0.9	1	15
$T_{1,4}$	0.8	0.8	0.7	0.7	40
$T_{1,5}$	1	0.8	0.8	0.9	30
$T_{1,6}$	0.7	1	0.6	0.8	40
$T_{1,7}$	0.9	0.7	1	0.6	31
$T_{1,8}$	0.6	0.6	0.5	0.7	55
$T_{1,9}$	0.5	0.3	0.5	0.3	80
$T_{1,10}$	0.5	0.9	0.5	0.8	45
$T_{1,11}$	0.8	0.8	0.8	0.7	35

method. Furthermore, there are 11 tasks that meet the requirement, and we put these 11 tasks into the effective similar task set $U_1 = \{T_{1,z} | z = 1, 2, \dots, 11\}$. The 11 similar tasks' satisfaction degree of knowledge and execution time are shown in Table 4.

Similarly, we can obtain similar effective task sets for $T_2 \sim T_{11}$ and put these sets into the entire effective similar task set $J = (U_i | i = 1, 2, \dots, 11)$.

5.2. Experiments on Calculating the Execution Time. The BP neural network was run in MATLAB R2014a. Here, we set the momentum factor as 0.9, the learning rate as 0.05, the error target as 10^{-5} , and the maximum number of training as 200. For task T_1 , we selected 8 similar tasks' data ($T_{1,11} - T_{1,8}$) as the train set, and 3 similar tasks' data ($T_{1,9} - T_{1,11}$) as the test set to train and test the BP neural network. The train and test result of the 11 similar tasks are listed in Table 5.

Table 5 reveals that the improved BP neural network method is suitable to solve the problem of execution time prediction, whose deviation is within 1%. Figures 5–7 show the iteration chart, training chart, and regression chart, respectively, which show a good convergence performance.

With the example for P_1 and T_1 , there are 4 kinds of knowledge required by T_1 , and then estimate the execution

TABLE 5: The train and test result of the 11 similar tasks.

	Sample	Actual value	Predictive value	Deviation
Train set	$T_{1,1}$	60	60.0004	-0.0004
	$T_{1,2}$	51	51.0022	-0.0022
	$T_{1,3}$	15	15.0042	-0.0042
	$T_{1,4}$	40	39.9975	0.0025
	$T_{1,5}$	30	29.9970	0.0030
	$T_{1,6}$	40	39.9967	0.0033
	$T_{1,7}$	31	30.9983	0.0017
	$T_{1,8}$	55	55.0027	-0.0027
Test set	$T_{1,9}$	80	79.9992	0.0008
	$T_{1,10}$	45	44.9956	0.0044
	$T_{1,11}$	35	35.0051	-0.0051

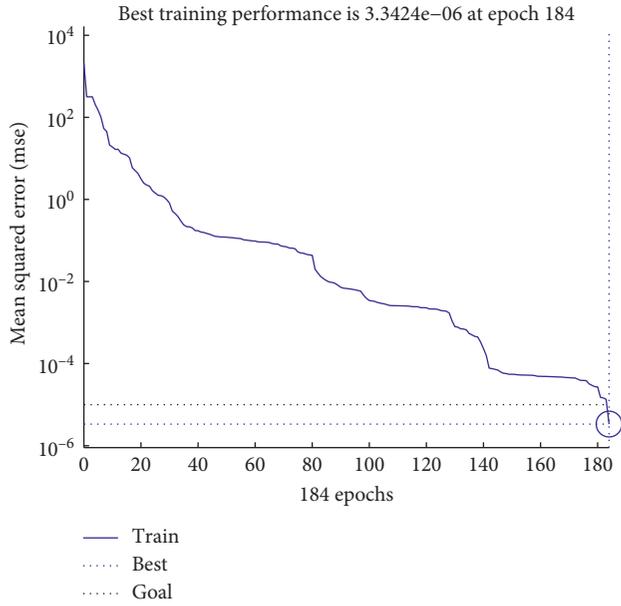


FIGURE 5: The iteration chart.

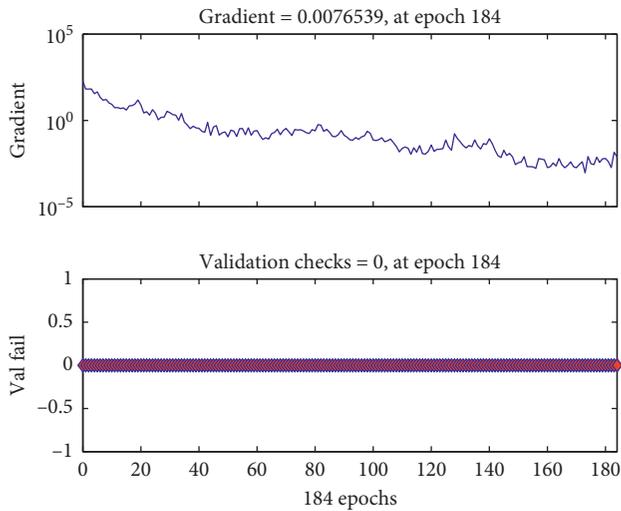


FIGURE 6: The training chart.

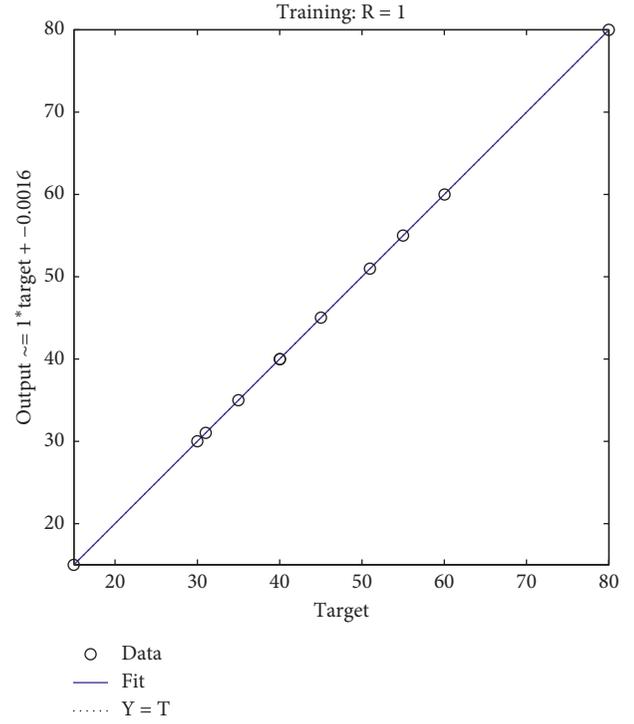


FIGURE 7: The regression chart.

TABLE 6: The execution time of P_1 for T_1

S_{111}	S_{121}	S_{131}	S_{141}	t_{11}^s
0.8	0.9	0.9	0.9	15.7082

TABLE 7: The influence of required knowledge on execution time.

S_{111}	S_{121}	S_{131}	S_{141}	t_{11}^s
0.8	0.9	0.9	0.9	15.7082
0.8	0.9	0.9	0.6	32.3823
0.8	0.9	0.6	0.9	40.1253
0.8	0.6	0.9	0.9	23.9871
0.5	0.9	0.9	0.9	29.1104

time of P_1 for T_1 (see Table 5) using the trained BP neural network. In Table 6, it shows that the execution time of T_1 by P_1 is about 16. Moreover, we get that the execution time of T_1 is relatively shorter compared with the 11 similar tasks, because the knowledge satisfaction degree of P_1 is relatively higher.

Furthermore, in order to investigate the influence of the required knowledge of T_1 on the execution time, the control variable method is used for analogous analysis. The influence of the satisfaction degree of knowledge on the execution time is shown in Table 7.

From Table 6, we can find that the 3rd required knowledge has the largest influence on the execution time of T_1 . Therefore, to improve the performance of T_1 , managers can improve the satisfaction degree of the 3rd knowledge by

TABLE 8: The execution time for all tasks.

t_{ik}^s	T_1	T_2	T_3	T_4	T_5	T_6	T_7	T_8	T_9	T_{10}	T_{11}
P_1	15.71	23.24	40.21	45.11	46.99	16.38	29.84	30.19	24.79	38.15	55.28
P_2	24.65	15.33	79.69	60.29	35.82	40.07	25.16	55.68	30.85	25.09	46.85
P_3	50.15	39.74	60.41	48.94	25.17	18.77	15.03	50.16	26.92	45.13	60.19
P_4	37.89	18.06	30.88	39.18	31.76	25.15	38.36	39.04	24.05	70.66	26.07
P_5	23.12	67.64	56.77	24.15	40.33	49.80	32.76	16.58	30.31	29.69	37.77

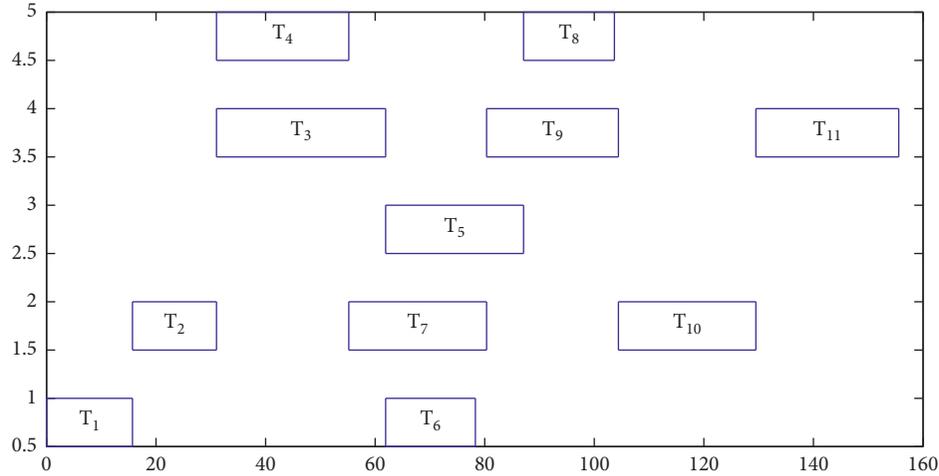


FIGURE 8: Gantt chart for task assignment.

TABLE 9: The task assignment scheme.

Task	Start time	Finish time	Execution time	Team
T_1	0	15.71	15.71	P_1
T_2	15.71	31.04	15.33	P_2
T_3	31.04	61.92	30.88	P_4
T_4	31.04	55.19	24.15	P_5
T_5	61.92	87.09	25.17	P_3
T_6	61.92	78.3	16.38	P_1
T_7	55.19	80.35	25.16	P_2
T_8	87.09	103.67	16.58	P_5
T_9	80.35	104.4	24.05	P_4
T_{10}	104.4	129.49	25.09	P_2
T_{11}	129.49	155.56	26.07	P_4

training team members with the 3rd knowledge or introducing people who are rich in the 3rd knowledge.

Similarly, we can obtain the execution time of each team for each task use the above method. The prediction results for all 11 tasks are listed in Table 8.

5.3. Experiments on Optimizing Task Assignment. The improved genetic algorithm is applied to solve the task assignment problem. According to the size and complexity of the problem, the parameters are set as Popsiz = 100, Maxgen = 50, where Popsiz is the population size and Maxgen is the number of generations. We set the selection rate at 0.8, the crossover rate at 0.8, and the mutation rate at 0.1. Finally, we obtain the Gantt chart (see Figure 8) and the corresponding task assignment schemes (see Table 9). In Table 9, we can get that the optimal

result of the task assignment problem is 155.56 according to theoretical execution time illustrated, and it is an efficient task assignment scheme which can give decision-maker satisfaction.

6. Conclusions

Aiming at the task assignment problem under uncertainty, this paper develops a method system for task assignment from the knowledge-based perspective, which contains three main components: similar tasks identification, tasks' execution time calculation, and task assignment model. First, this work proposes a model to calculate tasks' execution time based on the T-K-P network and BP neural network, which accounts for the relationship between knowledge satisfaction degree and tasks' execution time. This method helps decision-makers to predict the execution time of each team more

objectively and effectively. Second, this paper proposes a task assignment model considering uncertainty, and an improved genetic algorithm to deal with the fluctuations in the execution time of tasks. When assigning tasks, we consider the degree of fluctuation and the probability of violating a constraint of task's execution time, and the results reflect the proposed method can effectively solve the task assignment problem and obtain the satisfied task assignment scheme. The proposed method can provide a helpful support to help decision-makers assign product R&D tasks efficiently. However, there are some limitations for this study. For example, we did not consider that the explicit knowledge and implicit knowledge may have more subtle influences on the task's execution time. This problem may be more applicable and interesting for real task assignment issue. It will also be our research direction in the future. Furthermore, in the future work, we will investigate more real instances of product R&D task assignment to further deepen the efficient task assignment method system and broaden its application range.

Data Availability

The data used to support the findings of this study are included within the article.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

The authors are grateful to the support of the Team of Trade Circulation Funding Project (CJSYTD201701), Youth Foundation of Ministry of Education of China (19YJC630141), and the Chongqing Humanities and Social Sciences Research Project (20SKGH110).

References

- [1] J. Jian, Y. Zhang, L. Jiang, and J. Su, "Coordination of supply chains with competing manufacturers considering fairness concerns," *Complexity*, vol. 2020, Article ID 4372603, 15 pages, 2020.
- [2] J. Su, C. Li, Q. Zeng, J. Yang, and J. Zhang, "A green closed-loop supply chain coordination mechanism based on third-party recycling," *Sustainability*, vol. 11, no. 19, p. 5335, 2019.
- [3] H. Zhu and M. Zhou, "Role-based collaboration and its kernel mechanisms," *IEEE Transactions on Systems Man & Cybernetics Part C*, vol. 36, no. 4, pp. 578–589, 2006.
- [4] J. Su, Y. Yang, and X. Zhang, "Knowledge transfer efficiency measurement with application for open innovation R&D networks," *International Journal of Technology Management*, vol. 81, no. 1, pp. 118–142, 2019.
- [5] X. Zhang, B. Gong, Y. Cao, Y. Ding, and J. Su, "Investigating participants' attributes for participant estimation in knowledge-intensive crowdsourcing: a fuzzy DEMATEL based approach," *Electronic Commerce Research*, 2020.
- [6] B. Alidaee, H. Gao, and H. Wang, "A note on task assignment of several problems," *Computers & Industrial Engineering*, vol. 59, no. 4, pp. 1015–1018, 2010.
- [7] L. Zhen, "Task assignment under uncertainty: stochastic programming and robust optimisation approaches," *International Journal of Production Research*, vol. 53, no. 5, pp. 1487–1502, 2015.
- [8] H. Golpra and E. B. Tirkolaei, "Stable maintenance task scheduling: a Bi-objective robust optimization model," *Computers & Industrial Engineering*, vol. 137, Article ID 106007, 2019.
- [9] J. Su, Q. Bai, S. Sindakis et al., "Vulnerability of multinational corporation knowledge network facing resource loss," *Management Decision*, 2020, In press.
- [10] S. Xu and E. Cavusgil, "Knowledge breadth and depth development through successful R&D alliance portfolio configuration: an empirical investigation in the pharmaceutical industry," *Journal of Business Research*, vol. 101, no. 8, pp. 402–410, 2019.
- [11] C. Guo, X. Hua, H. Wu, D. Lautner, and S. Ren, "Best-harmonically-fit periodic task assignment algorithm on multiple periodic resources," *IEEE Transactions on Parallel and Distributed Systems*, vol. 27, no. 5, pp. 1303–1315, 2016.
- [12] D. Liu, B. Huang, and H. Zhu, "Solving the tree-structured task allocation problem via group multirole assignment," *IEEE Transactions on Automation Science and Engineering*, vol. 17, no. 1, pp. 41–55, 2020.
- [13] H. Zhu, M. Zhou, and R. Alkins, "Group role assignment via a kuhn-munkres algorithm-based solution," *IEEE Transactions on Systems, Man, and Cybernetics—Part A: Systems and Humans*, vol. 42, no. 3, pp. 739–750, 2012.
- [14] X. Zhu, Y. Wang, W. Hu, et al., "Multi-task assignment in information Technology companies based on the E-CARGO model," in *Proceedings of the IEEE International Conference on Systems, Man, and Cybernetics*, Kowloon, China, October 2015.
- [15] J. Yi, Q. Zhuge, J. S. Hu, M. Qin, and E. H.-M. Sha, "Reliability-guaranteed task assignment and scheduling for heterogeneous multiprocessors considering timing constraint," *Journal of Signal Processing Systems*, vol. 81, no. 3, pp. 359–375, 2015.
- [16] R. M. Gu, N. Nedjah, and L. M. Mourelle, "Efficient distributed algorithm of dynamic task assignment for swarm robotics," *Neurocomputing*, vol. 1, pp. 500–510, 2015.
- [17] H. Cheng and X. Chu, "Task assignment with multiskilled employees and multiple modes for product development projects," *The International Journal of Advanced Manufacturing Technology*, vol. 61, no. 1–4, pp. 391–403, 2012.
- [18] X. Zhang and J. Su, "An integrated QFD and 2-tuple linguistic method for solution selection in crowdsourcing contests for innovative tasks," *Journal of Intelligent & Fuzzy Systems*, vol. 35, no. 6, pp. 6329–6342, 2018.
- [19] N. Li, M. Y. Li, and G. H. GongKong, "An optimization method for task assignment for industrial manufacturing organizations," *Applied Intelligence*, vol. 47, no. 4, pp. 1144–1156, 2017.
- [20] J. F. S, Y. Yu, and Y. Tao, "Measuring knowledge diffusion efficiency in R&D networks," *Knowledge Management Research & Practice*, vol. 16, no. 2, pp. 208–219, 2018.
- [21] X. Zhang and J. Su, "A combined fuzzy DEMATEL and TOPSIS approach for estimating participants in knowledge-intensive crowdsourcing," *Computers & Industrial Engineering*, vol. 137, Article ID 106085, 2019.
- [22] H.-F. Ho and S.-T. Li, "Using mutually validated memories of experts for case-based knowledge systems," *Knowledge-Based Systems*, vol. 86, pp. 102–115, 2015.

- [23] F. Serghei, P. Sorin, and T. Richard, "Strategies for managing the structural and dynamic consequences of project complexity," *Complexity*, vol. 2018, Article ID 3190251, 17 pages, 2018.
- [24] Y. Su and T. Li, "Simulation analysis of knowledge transfer in a knowledge alliance based on a circular surface radiator model," *Complexity*, vol. 2020, Article ID 4301489, 27 pages, 2020.
- [25] S. B. Roy, I. Lykourentzou, S. Thirumuruganathan et al., "Task assignment optimization in knowledge-intensive crowdsourcing," *VLDB Journal*, vol. 24, no. 4, pp. 467–491, 2015.
- [26] Y. J. Lyu, G. Zhao, and Y. Yu, "Task assignment method of complex product design based on fuzzy set theory and knowledge similarity," *Computer Integrated Manufacturing Systems*, vol. 21, no. 4, pp. 904–913, 2015.
- [27] C. Youling, Z. Lidan, N. Yufei et al., "Task assignment method of product development based on knowledge similarity," *Journal of Computer Applications*, vol. 39, no. 2, pp. 323–329, 2019.
- [28] S. Quintana-Amate, P. Bermell-Garcia, and A. Tiwari, "Transforming expertise into Knowledge-Based Engineering tools: a survey of knowledge sourcing in the context of engineering design," *Knowledge-Based Systems*, vol. 84, no. 8, pp. 89–97, 2015.
- [29] N. Li, H. Kong, Y. Ma et al., "Human performance modeling for manufacturing based on an improved KNN algorithm," *International Journal of Advanced Manufacturing Technology*, vol. 84, no. 1–4, pp. 473–483, 2016.
- [30] W. Xu, C. Chen, S. Ding, and P. M. Pardalos, "A bi-objective dynamic collaborative task assignment under uncertainty using modified MOEA/D with heuristic initialization," *Expert Systems With Applications*, vol. 140, Article ID 112844, 2020.
- [31] M. Padulo and M. S. Liou, "A min max framework for robust design optimization," in *Proceedings of the 10th World Congress on Structural and Multidisciplinary Optimization*, Orlando, FL, USA, May 2013.
- [32] A. Ben-Tal and A. Nemirovski, "Selected topics in robust convex optimization," *Mathematical Programming*, vol. 112, no. 1, pp. 125–158, 2007.
- [33] D. Bertsimas, D. B. Brown, and C. Caramanis, "Theory and applications of robust optimization," *SIAM Review*, vol. 53, no. 3, pp. 464–501, 2011.
- [34] A. Ben-Tal and A. Nemirovski, "Robust solutions of Linear Programming problems contaminated with uncertain data," *Mathematical Programming*, vol. 88, no. 3, pp. 411–424, 2000.
- [35] D. Bertsimas and M. Sim, "The price of robustness," *Operations Research*, vol. 52, no. 1, pp. 35–53, 2004.
- [36] C. Büsing and F. D'Andreagiovanni, "New results about multi-band uncertainty in robust optimization," in *Proceedings of the International Symposium on Experimental Algorithms*, Springer, pp. 63–74, Bordeaux, France, June 2012.
- [37] T. Assavapokee, M. J. Realf, and J. C. Ammons, "Min-max regret robust optimization approach on interval data uncertainty," *Journal of Optimization Theory and Applications*, vol. 137, no. 2, pp. 297–316, 2008.
- [38] G. Carello and E. Lanzarone, "A cardinality-constrained robust model for the assignment problem in Home Care services," *European Journal of Operational Research*, vol. 236, no. 2, pp. 748–762, 2014.
- [39] Y. Chen, D. Yang, and J. Yu, "Multi-UAV task assignment with parameter and time-sensitive uncertainties using modified two-Part Wolf pack search algorithm," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 54, no. 6, pp. 2853–2872, 2018.
- [40] J. Su, M. Wei, and A. Liu, "A robust PredictiveReactive allocating approach, considering random design change in complex product design processes," *International Journal of Computational Intelligence Systems*, vol. 11, no. 1, pp. 1210–1228, 2018.
- [41] Y. Li, Y. Lu, D. Li et al., "Metanetwork analysis for project task assignment," *Journal of Construction Engineering & Management*, vol. 141, no. 12, Article ID 04015044, 2015.
- [42] Z. Li and M. Ierapetritou, "Process scheduling under uncertainty: review and challenges," *Computers & Chemical Engineering*, vol. 32, no. 5, pp. 715–727, 2008.
- [43] Y. Su and W. Sun, "Analyzing a closed-loop supply chain considering environmental pollution using the NSGA-II," *IEEE Transactions on Fuzzy Systems*, vol. 27, no. 5, pp. 1066–1074, 2019.
- [44] J. Long, S. Zhang, and C. Li, "Evolving deep echo state networks for intelligent fault diagnosis," *IEEE Transactions on Industrial Informatics*, vol. 16, no. 7, pp. 4928–4937, 2020.
- [45] J. Jian, Y. Guo, L. Jiang, Y. An, and J. Su, "A multi-objective optimization model for green supply chain considering environmental benefits," *Sustainability*, vol. 11, no. 21, p. 5911, 2019.
- [46] Y. Peng, J. Kong, X. Chen et al., "Job shop scheduling algorithm for intelligent manufacturing," *Journal of Nanjing University of Science & Technology*, vol. 41, no. 3, pp. 322–329, 2017.