

Research Article

Research on Dynamic Task Allocation Algorithm to Improve User Participation in the Witkey Mode

Yujie Wan , **Minglan Fu** , **Lvqiang Chen**, **Debao Chen**, **Jiekun Li**, **Wei Zhou**,
and **Mengxue Liu**

School of Computer Science and Technology, Huaibei Normal University, Huaibei 235000, China

Correspondence should be addressed to Minglan Fu; 11908110448@chnu.edu.cn

Received 15 February 2022; Revised 7 March 2022; Accepted 16 March 2022; Published 23 April 2022

Academic Editor: Qiangyi Li

Copyright © 2022 Yujie Wan 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.

The task allocation process in the Witkey mode is dynamic and open, in which Witkey is a rational person. Due to Witkey's individual rationality, in order to make the result of task allocation stable, task allocation must reach the Nash equilibrium. However, the Nash equilibrium point does not necessarily have the highest total system revenue. In order to make the task allocation result stable and have a high total system revenue, this paper proposes an incentive measure based on integral ranking to improve user participation. When Witkey adopts the best response strategy to select task, the order of participating in the selection will affect the individual income of Witkey to a certain extent. The higher the order, the greater the probability of obtaining higher system income. Based on this idea, a dynamic task allocation algorithm with complex tasks in the Witkey mode is proposed by combining incentive measures with best response strategy and reasonable benefit allocation strategy. The finally simulation results verified the effectiveness of the proposed algorithm, and the impact of incentive measures on the total revenue of the system was also examined.

1. Introduction

The tasks in the Witkey platform are often complex or complicated, which they cannot solve or even find suitable partners to solve, and can only seek online help through the Witkey platform. Guest model is relying on the individual or cooperation team of wisdom and creative to create value to obtain the way of income. For example, the user use the Internet to solve the problems in science, technology, work, life, learning, and other aspects and let knowledge, wisdom, experience, and service turn into economic benefits [1]. This model well reflects that the Internet is a new concept of people-centered community where users achieve value output through work and is a creative Internet knowledge management mode. In the age of data, the free sharing of information has promoted the vigorous development of the Internet, such as the emergence of Wiki websites, search engines, blog websites, etc., which have brought a lot of convenience to our life and work. However, from the perspective of human resource management, whenever the use

of personal knowledge, wisdom, ability, experience, and tasks need to cost a certain economic and time cost that do not need any cost of resource sharing is not practical and does not conform to the law of economics. Consequently, cost-free mode to the end will only hinder the role of the development of the Internet. Generally speaking, individuals protect their core capabilities, which makes shared resources difficult to improve when a certain apex is reached. In addition, the diversification and continuous improvement of online payment means enabling the Internet to price prices for knowledge, wisdom, ability, experience, etc., according to its need. It means that the era of completely free sharing of Internet information has passed and is changing to the era of the value of online resources [2]. This shift is actually a change in the state of intellectual freedom, which can effectively improve the enthusiasm of solution providers to obtain more targeted problem-solving capabilities. More importantly, under the value environment of Internet resources, the employment mode of the Witkey model is more flexible. It is no longer restricted by time, place, and way of

work. The Internet allows workers from all over the world to work on a unified platform and a level playing field. The application of the Witkey model gives them more free working hours, and the competitive and cooperative ways can also bring them more ideas and creativity to the Witkey model. So, making good use of the knowledge of users, wisdom, service and experience of this platform can pay less cost to get the same quality service or even higher quality service to meet the needs of enterprises or individuals in some aspects. It is clear that the Witkey model is good to meet the supply and demand of service and a more efficient and lower cost matching [3].

Rational people or rational agents are characterized by self-interest. Compared with the task assignment of collaborative agents in the general sense, the task assignment of rational users has some unique properties, such as the following: (1) the task assignment of rational users must meet the individual rationality [4, 5]. However, collaborative agents always make decisions that maximize system benefits within their capabilities (communication, computation, etc.). (2) For rational users, in order to maximize their individual benefits, they may deliberately make decisions that harm the benefits of others or system. (3) For rational users, after the completion of the task is completed, whether the benefit distribution is reasonable directly affects whether the cooperation can continue. Unreasonable allocation schemes may affect the enthusiasm of rational users to participate in future tasks and may even lead to their termination of the execution of the task midway. For cooperative agents, how to allocate the benefits is not a problem that must be considered. At the same time, because the income obtained after the task is usually a set value, for rational users, the benefit allocation scheme must meet the budget effectiveness of it [6, 7]. (4) In communication, in order to obtain better personal benefits, rational users may conceal personal information, such as geographical location, cost, or even conceal the specific content of the task. For collaborative agents, there is usually an implicit assumption that they are willing to transmit all their information to other agents or centralized managers as long as communication conditions allow. (5) Under the same conditions, the system income obtained from the rational user task allocation cannot be greater than the system income obtained from the task allocation of the collaborative agents.

Due to the abovementioned differences, some existing complex task allocation algorithms for cooperative agents cannot be directly used to solve the complex task allocation of rational users in the Witkey mode. In this paper, we study the complex task assignment problem model in the Witkey model and a task assignment algorithm based on best response strategy is proposed. Based on this, a task assignment algorithm called Task Allocation Algorithm Considering Witkey Participation (TAACWP) is proposed to improve user participation. The final simulation results verify the effectiveness of this algorithm.

The remainder of this article is organized as follows. Section 2 summarizes the current situation of domestic and foreign research on related problems. Section 3 establishes a problem model for complex task benefit allocation and task

assignment in the Witkey model. Section 4 describes the basic idea of the TAACWP algorithm and analyzes its convergence. Simulation results in Section 5 show that the algorithm proposed in this paper can efficiently distribute complex tasks and proceeds to rational users. Section 6 summarizes the work presented in this article.

2. Related Work

This section analyzes the status of research at home and abroad from two aspects of benefit allocation algorithm and complex task allocation algorithm for rational agents.

Benefit allocation algorithm: When the task set accomplished by all possible sets of users is known, the solution concept of the cooperative game can be used to make benefit allocation if they are independent of each other [8, 9]. However, in practical applications such as Witkey, the task set that a user set can accomplish is generally unknown, and for n users, there is a possible set of 2^n users with high temporal and spatial complexity. Meanwhile, when a task is assigned to a single set of users, the task cannot be reassigned to other user sets, so the sets of tasks that can be accomplished by different user sets are also not generally independent of each other.

In the CRA (Consensus-based Reward Allocation) algorithm [10], a concept called “return” was introduced to distribute benefits. Return is the ratio of the remuneration each user receives to its cost. To achieve fairness, the CRA tries to align returns for all users. The fairness is reflected in the higher the cost of users to get a more share of the revenue. However, this is actually not conducive to the improvement of the total system revenue. Instead, to increase the total revenue of the system, it should be tried to allocate tasks to lower cost users to complete. The IRA (Intermediary Recruitment Algorithm) algorithm can also solve the benefit assignment and task assignment of complex tasks, but the algorithm time complexity is high when the number of services required is large [11].

Complex task assignment algorithm of rational agents: Game theory is an effective method to study the decision-making of rational agents. Learning algorithms based on game theory mainly include the following: Best Response, (BR) [12, 13], Fictitious Play (FP) [14], Computationally Efficient Sampled FP (CESFP) [15], etc. Among them, for the optimal response strategy to obtain good task allocation results, then the benefits must be reasonably distributed, and the best response strategy does not account for the allocation of benefits. In virtual countermeasure algorithms, rational user decisions are based on their own historical information, which is not conducive to improving the quality of task assignment.

Other similar studies on rational multiagent systems have mainly focused on resource allocation [16–20] and the single-agent task assignment [21–26]. The difference between resource assignment and task assignment is that the resources owned by the user in the resource assignment can be transferred, while the services provided by the user in Witkey customer cannot be transferred between the users [24]. The single-agent tasks are tasks that can be done

without the collaboration of multiple agents. While there are also tasks in the Witkey mode that can require only one user, there are also many complex tasks that require multiple users to complete [25].

In addition to the Witkey system, there are task allocation modes where rational users participate in practical applications and platforms such as crowdsourcing, such as the public intelligence perception system [21–23, 26] and space crowdsourcing [24, 25, 27, 28]. Existing crowdsourcing platforms propose corresponding benefits and task assignment algorithms; however, the allocation algorithms are mostly for single-user tasks [29]. Less consideration is given to the cases when more users complete the task. Therefore, in order to trade off the multiuser allocation situation under complex tasks, this paper presents a multiuser-oriented complex task allocation algorithm.

3. Definition of the Problem Model

This section gives the definition of the dynamic task allocation problem model in the Witkey platform: the task allocation in Witkey platform is a dynamic process. The task allocation model at the moment τ includes the following three sets: Witkey set $R(\tau) = \{r_1(\tau), r_2(\tau), \dots, r_{n(\tau)}(\tau)\}$, service set $S = \{s_1, s_2, \dots, s_l\}$, and task set $T(\tau) = \{t_1(\tau), t_2(\tau), \dots, t_{m(\tau)}(\tau)\}$, and $n(\tau)$ and $m(\tau)$, respectively, represent the number of services available to the Witkey platform and the task needed to be completed at the moment τ , whose value changes over time. Assuming that the Witkey platform needs at most l types of services to complete complex tasks, a Witkey platform can provide at most l of services. that is, l represents the maximum number of services that users on the Witkey platform can provide.

For $i \in \{1, 2, \dots, n(\tau)\}$, $j \in \{1, 2, \dots, l\}$, $RS_{i,j}(\tau) = 1$ ($RS_{i,j}(\tau) = 0$) denotes that Witkey can(not) provide services s_j . $TSN_i(\tau)$ indicates the number of skills the i th guest $r_i(\tau)$ has. $ask_{i,j}(\tau)$ represents the lowest price Witkey $r_i(\tau)$ can charge for the skills s_j to complete task. If $RS_{i,j}(\tau) = 0$, $ask_{i,j}(\tau) = 0$. This paper considers complex tasks in the Witkey mode, so for $j \in \{1, 2, \dots, l\}$ and $k \in \{1, 2, \dots, m(\tau)\}$, $ST_{j,k}(\tau) = 1$ ($ST_{j,k}(\tau) = 0$) indicates that the task $t_k(\tau)$ requires (not) services s_j . For each Witkey, $r_i(\tau) \in R(\tau)$ at the moment τ , it corresponds to an integral as $d_j(\tau) \in N$ (N representing the set of natural numbers). The gains corresponding to all tasks at the time τ are indicated as $U(\tau) = \{u_1(\tau), u_2(\tau), \dots, u_{m(\tau)}(\tau)\}$. Each task corresponds to a maximum waiting time expressed as $E(\tau) = \{e_1(\tau), e_2(\tau), \dots, e_{m(\tau)}(\tau)\}$. For $k \in \{1, 2, \dots, m(\tau)\}$, $e_k(\tau) \in N^+$.

The benefit allocation scheme for all tasks is represented as $TS(\tau)$. For $k \in \{1, 2, \dots, m(\tau)\}$ and $j \in \{1, 2, \dots, l\}$, $TS_{k,j}(\tau)$ represents the share of revenue that task $t_k(\tau)$ would give to Witkey who provides service s_j at moment τ . If the task $t_k(\tau)$ can be completed at a moment τ and $r_i(\tau)$ choose to perform the task $t_k(\tau)$ and provide services s_j , then the revenue $TS_{k,j}(\tau)$ share will be obtained by $r_i(\tau)$. $RTS(\tau)$ denote the tasks chosen by Witkey and services provided by Witkey at time τ . $TSN_k(\tau)$ indicates the number of skills required for the task $t_k(\tau)$. $RTS_{i,0}(\tau)$ is the

number of the task selected by $r_i(\tau) \in R(\tau)$ at the time τ . Therefore, $RTS_{i,0}(\tau) \in \{1, 2, \dots, m(\tau)\}$. For $i \in \{1, 2, \dots, n(\tau)\}$, $j \in \{1, 2, \dots, l\}$, and $RTS_{i,j}(\tau) = 1$ ($RTS_{i,j}(\tau) = 0$) means that it $r_i(\tau)$ (not) provides services s_j for the task $t_{RTS_{i,0}(\tau)}$ at all times τ . If $RTS_{i,0}(\tau) = 0$, then for $\forall j \in \{1, 2, \dots, l\}$, $RTS_{i,j}(\tau) = 0$ is true.

The task $t_k(\tau)$ can be completed if $t_k(\tau)$ acquired all the required skills. Given the task selection scheme for all users $RTS(\tau)$, the system gain at the time τ is defined as the sum of all tasks that can be completed, recorded as $SR(\tau)$. The optimal allocation of complex tasks in the Witkey model is the benefit allocation and task allocation scheme that can maximize the system benefits.

The status of Witkey and tasks at moment τ is indicated as follows:

Status of the task: State of Witkey: At the moment τ , the state of Witkey $r_i(\tau) \in R(\tau)$ is expressed as $r_i(\tau) = \langle RS_{i,\cdot}(\tau), RTS_{i,\cdot}(\tau), ask_{i,\cdot}(\tau), d_i(\tau) \rangle$. In particular, $RS_{i,\cdot}(\tau)$ is row i of $RS(\tau)$; $RTS_{i,\cdot}(\tau) = \{RTS_{i,0}(\tau), RTS_{i,1}(\tau), \dots, RTS_{i,l}(\tau)\}$; $ask_{i,\cdot}(\tau)$ is row i of $ask(\tau)$; $d_i(\tau)$ is the integral of Witkey $r_i(\tau)$ at the moment τ .

Status of the task: The status of the task $t_k(\tau) \in T(\tau)$ at the moment τ is represented as $t_k(\tau) = \langle ST_{\cdot,k}(\tau), TSN_k(\tau), u_k(\tau), TS_{k,\cdot}(\tau) \rangle$. Among these, $ST_{\cdot,k}(\tau)$ is the column k of $ST(\tau)$. $TSN_k(\tau) \in Z^+$ represents the number of skills required for the task $t_k(\tau)$. $u_k(\tau)$ represents the gains available after the task $t_k(\tau)$ is completed. $TS_{k,\cdot}(\tau)$ is the row k of $TS(\tau)$, indicating the benefit distribution scheme of $t_k(\tau)$ at the time.

To facilitate analysis, the following assumptions are made for the abovementioned problem model:

Hypothesis 1. Each guest can choose one task at any one time and provide one service, while the number of specific services required for each task is 1.

Hypothesis 2. At any moment, Witkey will only choose the task that brings it the maximum individual benefit.

Hypothesis 3. For the tasks assigned by the Witkey platform, Witkey has the right to refuse and choose another task that can bring it more individual benefits.

Based on the above dynamic task assignment environment, this paper addresses the following dynamic task assignment problems:

$$\begin{aligned} & \max_{RTS(1) \dots RTS(\Gamma)} SSR(\Gamma) \\ & = \max_{RTS(1) \dots RTS(\Gamma)} \sum_{\tau=1}^{\Gamma} \left(\sum_{t_k(\tau) \in T(\tau) \wedge \text{complete}_{e_k}(\tau)} u_k(\tau) \right). \quad (1) \end{aligned}$$

$SSR(\Gamma)$ represents the sum of all the tasks that can be completed from time 1 to time Γ . $RTS(1) \dots RTS(\Gamma)$ satisfies the constraint of ‘‘Hypothesis 1’’. If the value of $\text{Complete}_{e_k}(\tau)$ is true, the value indicating that the task $t_k(\tau)$ can be completed under the task assignment state $RTS(\tau)$ at the moment τ . Otherwise, the task $t_k(\tau)$ cannot be completed.

4. Best Response Strategy and Intermediary Recruitment Algorithm

4.1. Best Response Strategy. $t(i, \tau)$ represents the task selected by Witkey $r_i(\tau) \in R(\tau)$ at the moment τ , and the task selection rule is shown in formula (2):

$$t(i, \tau + 1) \leftarrow \underset{\substack{t_k(\tau) \in T(\tau) \wedge \\ s_j \in S^{\wedge \text{need}}_{k,j}(\tau)}}{\text{argmax}} TS_{k,j}(\tau) \cdot RS_{i,j}(\tau), \quad (2)$$

where $\text{need}_{k,j}(\tau)$ indicates whether service s_j needed by task $t_k(\tau)$ is provided by other Witkey except $r_i(\tau)$ under the task assignment status $RTS(\tau)$ in the moment τ .

Return False, if provided; Others, True. Meanwhile, the benefit allocation strategy for all tasks meets the budget effectiveness conditions shown in formula (3):

$$\sum_{s_j \in S, ST_{j,k}(\tau)=1} TS_{k,j}(\tau) \leq u_k(\tau). \quad (3)$$

At some point, when all Witkey, tasks, and the benefit distribution scheme of tasks remain unchanged, all Witkeys take turns to perform the strategy shown in formula (1) to make the best choice. The optimal task selection process is a kind of weak noncircular game, and each weak noncircular game has limited improvement characteristics [30]. Thus, the procedure will converge to a Nash equilibrium point. Formula (1) ensures the individual rationality of the task assignment and the stability of the task assignment results.

4.2. Intermediary Recruitment Algorithm. Because the incentives to increase users engagement examined in this paper are based on existing task assignment and benefit assignment algorithms, a simple description of the basic idea of “intermediary recruitment algorithm” was given. The proposed algorithm simulates the operation mechanism of the talent recruitment market with an intermediary to solve the task selection problem of self-interest service agent. Workers who provide services are seen as applicants, skills as intermediaries, and tasks as companies to hire employees. Unlike the real recruitment market, a “service intermediary” only manages the workers who can provide specific services and the tasks that need them. “service intermediary” pairs workers and tasks based on information about the number of workers who can provide the service, the costs, the number of tasks that need the service, and the remuneration that can be provided. The proposed algorithm can reasonably distribute the benefits of complex tasks to a certain extent and improve the total system income of task distribution, but because the results of task allocation do not guarantee individual rationality, the task distribution is not stable.

The steps of one task assignment by the intermediary recruitment algorithm are described as follows:

Step 1. All “service intermediary” count basic information about the workers who can provide the service and the tasks

that need the service. Witkey is sorted from small to large at the minimum asking price, and the tasks are sorted from large to small by the revenue share allocated to the corresponding skills. Witkey and task were paired in order. It was success, if the gain share was greater than the minimum asking price, or the pairing failed.

Step 2. All tasks take turns to check whether all the required services are paired successfully. For services without paired success, try to increase the share of revenue allocated to workers who can provide the service until it can be increased.

Step 3. Check if all workers pair with only one task. If so, this task assignment ends; otherwise, go to Step 4.

Step 4. For workers who pair more than one tasks, select a task that brings the maximum individual benefits from the paired successful task and reject other tasks. The rejected intermediary then entered the competitive price adjustment phase until no task provided the worker with greater individual benefit beyond the worker’s currently chosen task, at which point the worker paired only a successful task.

Step 5. This round of task assignment ends. Update the data and go to the next round of task assignment.

4.3. Incentive Measures. It can be seen from formula (1) that in the process of choosing the optimal task according to the optimal response strategy, the earlier he chooses the task, the more likely it is to obtain greater individual benefits. Therefore, for every Witkey, $r_i(\tau) \in R(\tau)$ at the moment τ , it is given an integral, recorded as $d_j(\tau) \in N$. Witkey turns the optimal selection strategy in order of the integral from large to small. If the Witkey $r_i(\tau) \in R(\tau)$ abandons the current maximum individual gain in order to increase the total system gain, the incremental value of the system gain is added to the customer as a new integral.

$$d_i(\tau + 1) = d_i(\tau) + (SR(\tau + 1) - SR(\tau)), \quad (4)$$

where $SR(\tau + 1)$ means that the total system revenue when Witkey $r_i(\tau) \in R(\tau)$ gives up the current task that brings itself maximum individual benefits and chooses to perform a task that can get greater system benefits. At this time, other tasks chosen by Witkeys except Witkey $r_i(\tau) \in R(\tau)$ are satisfied with individual rationality. In other words, only Witkey $r_i(\tau) \in R(\tau)$ sacrificed its own individual income, thus increasing the total revenue of the system. Therefore, in order to motivate Witkey $r_i(\tau) \in R(\tau)$ to participate in the execution of tasks, the increased system revenue is added to Witkey $r_i(\tau) \in R(\tau)$ as an integral. At the same time, the increased system revenue can also increase its participation in performing tasks to a certain extent.

4.4. Algorithmic Description. This section provides a simple description of the task assignment algorithm (Task

```

import:  $n(\tau)$ ,  $m(\tau)$ ,  $\Gamma$ 
Output: the maximum total system revenue  $SSR(\Gamma)$  and its counterpart.  $RTS(1) \dots RTS(\Gamma)$ 
(1) FOR  $\tau \in \{1, 2, \dots, \Gamma\}$ 
(2) Use the “intermediary recruitment algorithm” for a round of task assignment.
(3) Record the system benefits  $SSR(\tau)$  at the moment  $\tau$ .
(4) FOR  $i \in \{1, 2, \dots, n(\tau)\}$ 
(5) Save all Witkey’s current task selection status  $RTS(\tau)$ .
(6) FOR  $k \in \{1, 2, \dots, m(\tau)\}$ 
(7) FOR  $j \in \{1, 2, \dots, L\}$ 
(8) Witkey  $i$  selects tasks and provides skills  $j$ .
(9) All others, except Witkey  $i$ , adopt the optimal reaction strategy selection task in turn in the order of integration from large to small, and the results of guiding two consecutive rounds of the optimal reaction are exactly the same.
(10) Record the new system benefits.  $SSR'(\tau)$ 
(11) END FOR
(12) END FOR
(13) IF  $SSR'(\tau) > SSR(\tau)$ 
(14)  $d_i(\tau) = d_i(\tau) + (SSR'(\tau) - SSR(\tau))$ 
(15) ELSE
(16) Restore the task selection status for all Witkey customers.  $RTS(\tau)$ 
(17) END IF
(18) END FOR
(19) Update data for Witkey and tasks: Remove tasks that are beyond the “maximum waiting time”, add new tasks and new customers, and update Witkey’s points, skills, and other information.
(20) END FOR

```

ALGORITHM 1: TAACWP.

Allocation Algorithm Considering Witkey Participation, TAACWP) as shown in the Algorithm 1:

5. Simulation Results

In Section 5.1, the algorithm TAACWP is compared with the other four algorithms for the average running time and the average system gains under four different datasets. Section 5.2 validates the effectiveness of TAACWP’s task assignment strategy and examines whether incentives have effects on rational people.

Simulation environment:

Memory: 3.0 G B; CPU, Intel (R) Pentium (R) CPU G2030; Main frequency: 3.0 GHz;

Operating system: Windows 7.

For the dataset, to our knowledge, there is currently no standard database for Coalitional skill games. For resource-constrained project scheduling issues, “Project Scheduling Problem Library” is available [31]. It is its standard database, but the resources in it do not have multiple skills. Another similar dataset is the iMOPSE [32, 33], The resources have multiple skills (the resources can be considered as the service agent in the coalitional skills game), but the task in iMOPSE requires only one resource. For some other similar problems, the datasets used are artificially randomly generated [34, 35]. In this paper, dataset 1, dataset 2, and dataset 3 are completely randomly generated. Dataset 4 was generated based on iMOPSE and was specifically generated as follows. The future needs to establish a corresponding database for the coalitional skill game problem model.

Dataset 1. (total of 15 runs, maximum number of iterations $\Gamma = 100$, generated dynamic task assignment data of 1.1–1.20) generates date which satisfies the following rules at the moment τ . (random (a, b) represents a random integer in the set): $n(\tau) = 30$, $l = 15$ and $m(\tau) = 30$. For $r_i(\tau) \in R(\tau)$, $ask_i(\tau) \leftarrow \text{random}(1, 2)$, and $RSN_i(\tau) \leftarrow \text{random}(1, 3)$. For $t_k \in T$, $u_k \leftarrow TSN_k \times \text{random}(1, m/2)$, $TSN_k(\tau) \leftarrow \text{random}(2, 5)$, and $e_k(\tau) \leftarrow \text{random}(5, 10)$.

Dataset 2. (total of 15 runs, each maximum number of iterations $\Gamma = 200$, generated dynamic task assignment data of 2.1–2.20) generates date which satisfies the following rules at the moment τ . (random (a, b) represents a random integer in the set $[a, b]$): $n(\tau) = 30$, $l = 15$ and $m(\tau) = 60$. For $r_i(\tau) \in R(\tau)$, $ask_i(\tau) \leftarrow \text{random}(1, 2)$, and $RSN_i(\tau) \leftarrow \text{random}(1, 3)$. For $t_k \in T$, $u_k \leftarrow TSN_k \times \text{random}(1, m/2)$, $TSN_k(\tau) \leftarrow \text{random}(2, 5)$, and $e_k(\tau) \leftarrow \text{random}(5, 10)$.

Dataset 3. (total of 15 runs, each maximum number of iterations $\Gamma = 200$, generated dynamic task assignment data of 2.1–2.20) generates date which satisfies the following rules at the moment τ . (random (a, b) represents a random integer in the set $[a, b]$): $n(\tau) = 30$, $l = 15$, and $m(\tau) = 60$. For $r_i(\tau) \in R(\tau)$, $ask_i(\tau) \leftarrow \text{random}(1, 2)$, and $RSN_i(\tau) \leftarrow \text{random}(1, 3)$. For $t_k \in T$, $u_k \leftarrow TSN_k \times \text{random}(1, m/2)$, $TSN_k(\tau) \leftarrow \text{random}(2, 4)$, and $e_k(\tau) \leftarrow \text{random}(5, 10)$.

Dataset 4. Witkey and task data in dataset 4 are from the following iMOPSE-based data, generating the following steps:

TABLE 1: Mean system gains of GGA when CP and MP take different values.

MP CP	0.01	0.05	0.1	0.15	0.2	0.4	0.6	0.8
0.1	15214	14845	16544	15802	15816	11349	10389	10668
0.2	14302	15684	16283	16717	14756	11766	10763	10125
0.25	14956	14316	15957	16277	14465	11516	11235	11158
0.3	14449	14224	16762	14597	13470	11578	10739	10055
0.35	14144	15718	14795	14767	13133	11369	11157	10559
0.4	14566	15914	15635	14637	12424	11652	11620	10728
0.5	15852	16046	15328	14414	12829	11450	10663	10539
0.6	14909	14689	15480	14175	12905	11355	10598	10883
0.8	14913	15731	15479	12974	12803	10898	11053	10881

TABLE 2: Average system gain and the average runtime under dataset 1.

Dataset	TAACWP	Time	GGA	Time	CBA	Time	VAA	Time	SAA	Time
1.1	17698	6.01	16762	27.32	17189	0.002	17924	1.67	12000	0.001
1.2	19511	6.27	15963	27.16	16156	0.002	17392	1.70	11303	0.001
1.3	17864	7.11	14236	27.63	17611	0.002	17596	1.54	12539	0.001
1.4	17668	5.61	16456	26.68	17653	0.001	16952	1.58	12630	0.001
1.5	19215	5.25	14287	27.38	17068	0.002	16883	1.65	8903	0.001
1.6	18494	5.53	16325	27.51	16697	0.002	17536	1.56	10167	0.001
1.7	19966	6.02	15164	27.31	17187	0.002	18486	1.55	7475	0.001
1.8	18754	6.62	16386	27.98	17507	0.003	16726	1.76	10130	0.001
1.9	16258	6.02	16263	25.77	16456	0.002	18049	1.66	10963	0.001
1.10	17599	8.71	16458	26.61	19063	0.002	16853	1.62	11529	0.001
1.11	18138	6.50	16422	27.58	16948	0.002	16436	1.61	9485	0.001
1.12	18873	7.15	16271	26.82	18391	0.002	17061	1.55	11065	0.001
1.13	20405	7.10	15637	26.88	18083	0.002	17548	1.47	11033	0.001
1.14	19455	7.21	16537	26.43	17544	0.002	17707	1.59	7970	0.001
1.15	16665	4.53	16475	27.77	16286	0.002	17932	1.56	10514	0.001

Step 6. Draw 300 user information (namely corresponding Witkey): 100_20_22_15, 100_20_23_9_D1, 200_40_91_15, 100_20_45_15, 200_40_45_9, 100_20_47_9, 100_20_65_9, 100_20_65_15, 200_40_45_15, 200_40_90_9, 200_40_130_9_D4, 200_40_133_15.

Step 7. Information for 400 tasks are extracted from the datasets 200_40_91_15 and 200_40_133_15.

Step 8. Make $RSN_i(\tau) \leftarrow \text{random}(1, 2)$ and $TSN_k(\tau) \leftarrow \text{random}(2, 5)$ established by randomly adding or deleting the corresponding skills.

Step 9. For $r_i(\tau) \in R(\tau)$, $ask_i(\tau) \leftarrow \text{random}(1, 2)$. For $t_k \in T$, $u_k \leftarrow TSN_k \times \text{random}(1, m/2)$ and $e_k(\tau) \leftarrow \text{random}(5, 10)$.

For the dataset, the generation process of dynamic data is as follows: At the moment τ , randomly select $\text{random}(1, N(\tau)/10)$ Witkey data meeting the above conditions to replace the Witkey data at moment $(\tau - 1)$, and specifically which to replace is randomly determined. The tasks that can be completed and tasks that cannot be completed are randomly replaced with new task data that meet the above requirements.

5.1. Algorithm Performance Comparison. In this section, the total sum of the system benefits $SST(\Gamma)$ and the average

running time $SST(\Gamma)/\Gamma$ of the TAACWP (where $SST(\Gamma)$ denotes the total time spent during the dynamic task assignment process): general genetic algorithm (GGA), SAA algorithm (Service and Adams' Algorithm, SAA) for Dataset 1, Dataset 2, Dataset 3, and Dataset 4 [36], Combinatorial auction algorithm (Combinatorial Bids based Algorithm, CBA) [37], and the VAA algorithm (Vig & Adams' Algorithm, VAA) [38]. Witkey in the CBA is self-beneficial, while Witkey in the GGA and SAA is not. At moments $\tau \in \{1, 2, \dots, \Gamma\}$, the dynamic task assignments have consistent data generation rules.

Dataset 1.1 was used to obtain the optimal parameters for the GGA. With cross probability (CP) and variant probability (MP) taking different values, GGA runs 100 times, and the average system gain is shown in Table 1 (rows indicate the value of MP, columns indicate the value of CP). Simulation results show that the average system gain is greatest when CP is 0.3 and MP is 0.1. Other parameters of the GGA were set as follows: population size 100 and maximum number of iterations 10000.

Under Dataset 1, 2, 3, and 4, TAACWP, GGA, CBA, VAA, and SAA are shown in Tables 2–5, respectively.

The results in Tables 2–5 show that the sum of system gains from TAACWP is higher than the other four algorithms and the average running time of TAACWP is shorter than GGA. Among them, due to the excessive search space of GGA, the system has the worst profit and the longest running time. Therefore, based on the best response strategy,

TABLE 3: Average system gain and average runtime under dataset 2.

Dataset 2	TAACWP	Time	GGA	Time	CBA	Time	VAA	Time	SAA	Time
2.1	98889	19.13	67145	54.12	95559	0.003	91818	2.65	59957	0.001
2.2	98868	13.17	70007	55.88	91044	0.003	96430	2.27	57204	0.001
2.3	102754	13.87	66835	55.73	92218	0.003	96946	2.62	54673	0.001
2.4	97487	20.49	67438	53.30	93269	0.002	97227	2.43	50268	0.001
2.5	99675	16.77	66559	55.82	98196	0.002	96003	2.24	54019	0.001
2.6	101690	13.73	68807	53.80	93596	0.002	88263	2.73	51064	0.001
2.7	104171	14.54	67066	54.10	93662	0.002	90900	2.49	56126	0.001
2.8	101749	13.58	64140	55.47	92379	0.002	94131	2.31	63947	0.001
2.9	101272	18.88	66121	54.18	92403	0.003	93697	2.28	55298	0.001
2.10	103154	13.51	69793	54.15	89038	0.002	93130	2.26	50347	0.001
2.11	99485	13.76	65344	54.26	95301	0.002	92426	2.66	53932	0.001
2.12	106730	15.26	70054	53.63	93353	0.002	92942	2.29	51910	0.001
2.13	101287	14.93	69952	54.81	89924	0.002	94595	2.99	55466	0.001
2.14	96293	14.41	68418	54.01	90996	0.002	93706	2.38	49241	0.001
2.15	100234	14.01	69904	53.59	94990	0.002	97527	2.35	60336	0.001

TABLE 4: Average system gain and average runtime under dataset 3.

Dataset 3	TAACWP	Time	GGA	Time	CBA	Time	VAA	Time	SAA	Time
3.1	57186	6.29	54630	41.39	52890	0.003	54032	3.42	47879	0.001
3.2	56372	6.38	57560	40.15	49564	0.003	54542	3.56	38207	0.001
3.3	58670	5.82	56900	43.72	52759	0.003	52792	3.31	45899	0.001
3.4	55647	5.79	58860	50.89	54387	0.003	55337	2.93	42465	0.001
3.5	55171	5.93	57957	50.85	52331	0.003	54270	3.89	42959	0.001
3.6	59095	6.27	55158	50.75	55796	0.003	52237	3.72	45613	0.001
3.7	58744	5.69	56413	50.06	54993	0.002	54087	2.97	47507	0.001
3.8	55763	6.21	55548	55.64	52884	0.003	53290	2.94	43793	0.001
3.9	56739	5.75	55401	55.82	54802	0.003	55517	3.84	40048	0.001
3.10	58789	5.53	56556	55.47	55285	0.002	53212	2.88	41612	0.001
3.11	58851	6.49	57071	55.47	52801	0.003	48991	3.29	45823	0.001
3.12	58704	6.18	55115	55.29	52322	0.003	53558	2.99	47562	0.001
3.13	56772	5.90	55749	55.25	54655	0.002	52154	3.15	44535	0.001
3.14	57485	5.45	55966	55.32	53265	0.003	53209	3.20	46139	0.001
3.15	57180	6.03	58899	55.70	54309	0.003	55750	4.10	43571	0.001

TABLE 5: Average system gain and average runtimes under dataset 4.

Dataset 4	TAACWP	Time	GGA	Time	CBA	Time	VAA	Time	SAA	Time
4.1	28356	4.99	24097	26.66	21444	0.001	26436	1.89	10537	0.001
4.2	29577	5.58	21171	27.39	26450	0.002	26002	1.88	6223	0.001
4.3	26597	6.39	25060	27.04	18570	0.001	24468	1.66	7833	0.001
4.4	28915	5.36	22402	27.82	22526	0.002	24569	1.76	5557	0.001
4.5	24871	5.52	23887	27.89	18883	0.001	25044	1.64	8975	0.001
4.6	24741	5.26	22859	28.20	20704	0.001	25961	1.82	9570	0.001
4.7	23595	5.42	22458	27.90	21528	0.002	20988	1.79	8145	0.001
4.8	24539	6.76	21405	27.98	23309	0.002	24874	1.61	8903	0.001
4.9	27705	5.27	24335	27.53	23508	0.001	21206	1.67	12335	0.001
4.10	21916	5.61	21926	27.37	25429	0.001	20719	1.67	5781	0.001
4.11	24597	5.65	25674	26.99	23422	0.002	24038	1.81	7432	0.001
4.12	28734	5.68	22845	27.91	20681	0.001	22990	1.72	9104	0.001
4.13	27083	5.94	21768	28.28	20304	0.001	25470	1.66	11840	0.001
4.14	23969	4.66	22941	27.47	25587	0.002	21247	1.63	7687	0.001
4.15	26703	5.17	22949	27.32	25816	0.001	24959	1.85	11133	0.001

TABLE 6: Average system gains and average runtime for TAACWP and TAACWPI under dataset 1, dataset 2, dataset 3, and dataset 4.

Dataset 1	TAACWP	TAACWPI	Dataset 2	TAACWP	TAACWPI	Dataset 3	TAACWP	TAACWPI	Dataset 4	TAACWP	TAACWPI
1.1	17698	17664	2.1	98889	98529	3.1	57186	57104	4.1	28356	28284
1.2	19511	19457	2.2	98868	98242	3.2	56372	56298	4.2	29577	29495
1.3	17864	17812	2.3	102754	102101	3.3	58670	58641	4.3	26597	26517
1.4	17668	17628	2.4	97487	96943	3.4	55647	55587	4.4	28915	28825
1.5	19215	19128	2.5	99675	99101	3.5	55171	55125	4.5	24871	24775
1.6	18494	18416	2.6	101690	101204	3.6	59095	59041	4.6	24741	24656
1.7	19966	19912	2.7	104171	103684	3.7	58744	58702	4.7	23595	23501
1.8	18754	18699	2.8	101749	101076	3.8	55763	55710	4.8	24539	24487
1.9	16258	16239	2.9	101272	100817	3.9	56739	56654	4.9	27705	27596
1.10	17599	17578	2.10	103154	102704	3.10	58789	58725	4.10	21916	21786
1.11	18138	18084	2.11	99485	99173	3.11	58851	58801	4.11	24597	24516
1.12	18873	18798	2.12	106730	105996	3.12	58704	58680	4.12	28734	28679
1.13	20405	20374	2.13	101287	100818	3.13	56772	56727	4.13	27083	27016
1.14	19455	19410	2.14	96293	95895	3.14	57485	57405	4.14	23969	23930
1.15	16665	16587	2.15	100234	99825	3.15	57180	57146	4.15	26703	26588

the adjustment of task income allocation scheme by using the intermediary allocation algorithm can also improve the system income when the service agent is self-interested.

5.2. Validation of the Effectiveness of the TAACWP Incentive Measures. This section examines the effect of incentives on the total system benefits of the TAACWP algorithm, in which the TAACWP algorithm with no incentives is recorded as TAACWPI. The simulation results are shown in Table 6:

The simulation results from Table 6 show that the system benefits of TAACWP are greater than TAACWPI. The simulation results further show that using incentives can guarantee the higher system gain, even if the user is rational.

6. Conclusion

In this paper, we propose an optimization algorithm that motivate rational Witkey to participate in performing tasks. The users in the Witkey model are rational people, who always choose the tasks that can bring them the most personal benefits and are easy to accomplish. The task requester’s goal is to acquire all the required skills and then complete the task. For rational Witkey, the stability of the task assignment results can be guaranteed only when the benefits are reasonably distributed and the task assignment achieves a Nash equilibrium. However, the self-interest will inevitably affect the improvement of the total system income. By encouraging Witkey to give up the task that brings itself greater individual benefits, it is necessary to choose the task that can maximize the total benefits of the system. Consequently, certain incentives must be taken. In the process of selecting tasks according to the optimal response strategy, the execution order will partly affect the individual benefits of Witkey. According to this feature, this paper proposes the incentive measure for additive integral, proposes the TAACWP algorithm to solve the benefit distribution of complex tasks in Witkey mode, and the final simulation results verify the effectiveness of the algorithm.

We also came to another conclusion: adding incentives seems to yield more benefits, but may not be the best option in all cases because of the computational cost. Future research work will further summarize the dynamic characteristics of Witkey task assignment; consider the Witkey integrity, skill level, and other characteristics; and improve the dynamic task assignment model. Moreover, we also will consider applying the proposed algorithm to various crowdsourcing platform task allocation scenarios and shop floor scheduling. On this basis, the task assignment algorithm meets the characteristics of individual rationality, budget effectiveness, privacy protection, and stability of task assignment results.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

The authors would like to acknowledge that this work was supported by the National Natural Science Foundation of China (Nos. 61976101, 62006091, and 71801108) and the University Natural Science Research Project of Anhui Province (No. KJ2020A0033).

References

- [1] F. Liu, “Speech by liu feng, the founder of witkey theory, at the first global witkey conference,” *News Research Guide*, vol. 3, pp. 12–14, 2011.
- [2] L. Yan, “Research on the cause, nature of witkey and the impact factors of global transaction—the perspective of network culture and creative industry,” *Journal of Intelligence*, vol. 30, no. 1, pp. 190–195, 2011.

- [3] S. Hupp and L. Crowdsourcing, "Why the power of the crowd is driving the future of business," *Library Journal*, vol. 134, no. 1, p. 50, 2009.
- [4] X. Ying, G. David, and J. Winfried, "Open-minded imitation can achieve near-optimal vaccination coverage," *Journal of Mathematical Biology*, vol. 79, no. 4, pp. 1491–1514, 2019.
- [5] H. Jin, L. Su, D. Chen, H. Guo, K. Nahrstedt, and J. Xu, "Thanos: incentive mechanism with quality awareness for mobile crowd sensing," *IEEE Transactions on Mobile Computing*, vol. 18, no. 8, pp. 1951–1964, 2019.
- [6] M. Drechsler, "Performance of input- and output-based payments for the conservation of mobile species," *Ecological Economics*, vol. 134, pp. 49–56, 2017.
- [7] L. Caragiannis and A. A. Voudouris, "The efficiency of resource allocation mechanisms for budget-constrained users," in *Proceedings of the ACM Conference on Economics and Computation*, pp. 681–698, Cornell Univ, Ithaca, NY, June 2018.
- [8] Q. An, Y. Wen, T. Ding, and Y. Li, "Resource sharing and payoff allocation in a three-stage system: integrating network DEA with the Shapley value method," *Omega*, vol. 85, pp. 16–25, 2019.
- [9] Y. Bachrach, E. Elkind, E. Malizia et al., "Bounds on the cost of stabilizing a cooperative game," *Journal of Artificial Intelligence Research*, vol. 63, pp. 987–1023, 2018.
- [10] J. Hou, S. Luo, W. Xu, and L. Wang, "Fairness-based multi-task reward allocation in mobile crowdsourcing system," *IET Communications*, vol. 13, no. 16, pp. 2506–2511, 2019.
- [11] M. L. Fu, H. Wang, and B. F. Fang, "Coalitional skill games for self-interested robots with SVO," *International Journal of Robotics and Automation*, vol. 33, no. 5, pp. 1–11, 2018.
- [12] B. C. Schipper, "Dynamic exploitation of myopic best response," *Dynamic Games and Applications*, vol. 9, no. 4, pp. 1143–1167, 2019.
- [13] B. Swenson, C. Eksin, S. Kar, and A. Ribeiro, "Distributed inertial best-response dynamics," *IEEE Transactions on Automatic Control*, vol. 63, no. 12, pp. 4294–4300, 2018.
- [14] N. Kamra, U. Gupta, K. Wang, F. Fang, Y. Liu, and M. Tambe, "Deep fictitious play for games with continuous action spaces," in *Proceedings of the 18th International Conference on Autonomous Agents and Multi-Agent Systems*, pp. 2042–2044, Montreal, Canada, May 2019.
- [15] B. Swenson, S. Kar, and J. Xavier, "A computationally efficient implementation of fictitious play for large-scale games," in *Proceedings of the 2015 23rd European Signal Processing Conference (EUSIPCO)*, pp. 1–16, IEEE, Nice, France, August 2015.
- [16] D. Li, Q. Yang, W. Yu, D. An, Y. Zhang, and W. Zhao, "Towards differential privacy-based online double auction for smart grid," *IEEE Transactions on Information Forensics and Security*, vol. 15, pp. 971–986, 2020.
- [17] M. N. Faqiry and S. Das, "Double auction with hidden user information: application to energy transaction in Microgrid," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 49, no. 11, pp. 2326–2339, 2019.
- [18] J. N. Qin, X. Fu, S. M. Peng, Y. Xu, J. Huang, and S. Huang, "Asymmetric bargaining model for water resource allocation over transboundary rivers," *International Journal of Environmental Research and Public Health*, vol. 16, no. 10, pp. 1–23, 2019.
- [19] L. Feng, Q. Yang, K. Kim, and K. S. Kwak, "Two-timescale resource allocation for wireless powered D2D communications with self-interested nodes," *IEEE Access*, vol. 7, pp. 10857–10869, 2019.
- [20] Y. Liang, W. Wei, and C. Wang, "A generalized Nash equilibrium approach for autonomous energy management of residential energy hubs," *IEEE Transactions on Industrial Informatics*, vol. 15, no. 11, pp. 5892–5905, 2019.
- [21] N. Cao, S. Brahma, B. Geng, and P. K. Varshney, "Optimal auction design with quantized bids for target tracking via crowdsensing," *IEEE Transactions on Computational Social Systems*, vol. 6, no. 5, pp. 847–857, 2019.
- [22] H. Cai, Y. Zhu, Z. Feng, H. Zhu, J. Yu, and J. Cao, "Truthful incentive mechanisms for mobile crowd sensing with dynamic smartphones," *Computer Networks*, vol. 141, pp. 1–16, 2018.
- [23] P. Zhou, W. Chen, S. Ji, H. Jiang, L. Yu, and D. Wu, "Privacy-preserving online task allocation in edge-computing-enabled massive crowdsensing," *IEEE Internet of Things Journal*, vol. 6, no. 5, pp. 7773–7787, 2019.
- [24] P. Xiong, D. Zhu, L. Zhang, W. Ren, and T. Zhu, "Optimizing rewards allocation for privacy-preserving spatial crowdsourcing," *Computer Communications*, vol. 146, pp. 85–94, 2019.
- [25] D. Yuan, Q. Li, G. Li, Q. Wang, and K. Ren, "PriRadar: a privacy-preserving framework for spatial crowdsourcing," *IEEE Transactions on Information Forensics and Security*, vol. 15, pp. 299–314, 2020.
- [26] C. Martin, D. Ivana, G. Iosifidis, and M. Bouroche, "Adaptive reward allocation for participatory sensing," *Wireless Communications and Mobile Computing*, vol. 2018, Article ID 6353425, 15 pages, 2018.
- [27] Y. Li, M. D. Jia, W. Y. Yang, and Y. Zhao, "Optimal Task Assignment Algorithm Based on Tree-Decouple in Spatial Crowdsourcing," *Journal of Software*, vol. 29, no. 3, pp. 824–838, 2018.
- [28] T. S. Song, Y. X. Tong, L. B. Wang, and K. Xu, "Online task assignment for three types of objects under spatial crowdsourcing environment," *Journal of Software*, vol. 28, no. 3, pp. 611–630, 2017.
- [29] E. Nunes, M. Manner, H. Mitiche, and M. Gini, "A taxonomy for task allocation problems with temporal and ordering constraints," *Robotics and Autonomous Systems*, vol. 90, no. 90, pp. 55–70, 2017.
- [30] K. R. Apt and S. Simon, "A classification of weakly acyclic games," *Theory and Decision*, vol. 78, no. 4, pp. 501–524, 2015.
- [31] R. Kolisch and A. Sprecher, "PspLib—a project scheduling problem library," *European Journal of Operational Research*, vol. 96, no. 1, pp. 205–216, 1997.
- [32] P. B. Myszkowski, M. E. Skowroń, P. Olech, and K. Oślizło, "Hybrid ant colony optimization in solving multi-skill resource-constrained project scheduling problem," *Soft Computing*, vol. 19, pp. 3599–3619, 2015.
- [33] B. P. Myszkowski, E. M. Skowroń, and K. Sikora, "A new benchmark data set for multi-skill resource-constrained project scheduling problem," in *Proceedings of the 2015 Federated Conference on Computer Science and Information Systems (FedCSIS)*, pp. 129–138, IEEE, New York, NY, USA, 2015.
- [34] T. Hegazy, K. Shabeeba, and E. Elbeltagi, "Algorithm for scheduling with multiskilled constrained resources," *Journal of Construction Engineering and Management*, vol. 126, no. 6, pp. 414–421, 2005.
- [35] A. Santosm and P. A. Tereso, "On the multi-mode, multi-skill resource constrained project scheduling problem—a software application," *Soft Computing in Industrial Applications*, vol. 1, no. 96, pp. 239–248, 2011.

- [36] T. C. Service and J. A. Adams, "Coalition formation for task allocation: theory and algorithms," *Autonomous Agents and Multi-Agent Systems*, vol. 22, pp. 225–248, 2011.
- [37] L. Lin and Z. Q. Zheng, "Combinatorial bids based multi-robot task allocation method," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pp. 1145–1150, IEEE, Barcelona, Spain, April 2005.
- [38] L. Vig and J. A. Adams, "Multi-Robot coalition formation," *IEEE Transactions on Robotics*, vol. 22, no. 4, pp. 637–649, 2006.