ISSN 2616-5775 Vol. 6, Issue 5: 14-20, DOI: 10.25236/AJCIS.2023.060503

Collective Assignment of Complex Crowdsourcing Tasks Based on the KM Algorithm

Jinwei Zhang[1,*]

[1]School of Information Engineering, Nanjing University of Finance and Economics, Nanjing, China
[*]Corresponding author

Abstract: *With the increasing complexity of crowdsourcing scenarios, the assignment of complex tasks brings new challenges to the further application of crowdsourcing. Previous research work only focused on how to find a worker team that meets the task requirements, without comprehensively considering factors such as task and worker skill characteristics, time, task budget, and worker compensation. Thus, the success rate of task assignment is low. When assigning large-scale complex tasks, relatively complex tasks will not be able to find workers who meet the requirements, resulting in allocation failures. Thus, this paper studies a complex task-oriented collective assignment model to solve the problem that a large number of complex tasks cannot be assigned in crowdsourcing platforms while many workers have no tasks to do. In the model, the complex task assignment problem is mapped as a weighted bipartite graph matching problem, then the KM algorithm is used to solve the optimal assignment scheme. Finally, this paper conducts comparative experiments on real datasets, and the results show that the proposed model has better performance in terms of task success rate and task payment total cost.*

Keywords: *Crowdsourcing, Complex task, Collective task allocation, KM algorithm*

1. Introduction

Crowdsourcing is a new type of task assignment model, which uses the wisdom of the masses to distribute and collaborate to complete tasks [1]. In traditional crowdsourcing platforms such as Amazon Mechanical Turk, workers can independently complete assigned simple tasks in a short period of time, such as image annotation [2], text labeling [3], etc. With the increasing complexity of crowdsourcing application scenarios, tasks are also becoming more and more complex. For example, in software projects released on the Upwork platform, tasks require more and more professional skills of workers, and usually no single worker can complete them alone [4]. Therefore, complex tasks pose great challenges to the further application of crowdsourcing.

Much of the existing research work on complex task allocation mainly focuses on methods based on task decomposition [5,6]. It requires the task requester to decompose the task into a collection of subtasks, however many task requesters may not necessarily have such ability. Consequently, some researchers have explored ways to directly assign complex tasks to teams of workers who meet the task's skill requirements. For example, in the literature [7], Wang et al. designed a team formation mechanism based on distributed negotiation, and the workers hired by the task requester can decide to join the team they prefer. However, most of such research work has the following disadvantages: 1) From the perspective of tasks, some relatively simple and complex tasks may preferentially use workers with high skill coverage, resulting in the remaining workers on the platform being relatively unskilled workers, so relatively complex tasks will not be able to find satisfying workers, resulting in allocation failures. 2) From the perspective of workers, most of the current research work starts from the perspective of controlling the quality of tasks completed by workers, ignoring the suitability of workers and tasks, that is, not assigning tasks to the most suitable workers to complete.

In general, most of the current researches do not consider the needs of tasks and the skill characteristics of workers globally, resulting in a low success rate of task assignment. In reality, we found that there are many unassigned complex tasks and workers who have no tasks to do in the crowdsourcing platform. One of the reasons is that the crowdsourcing platform will assign tasks one by one according to the order of the release time of the tasks, which leads to the failure of subsequent tasks [8]. Workers that meet the requirements are found, but these remaining workers can cooperate with the assigned workers to complete more tasks. Since more and more specialized workers are required for complex tasks, it is especially necessary to use workers rationally when a large number of tasks are assigned at the same time. Existing work shows that multi-skill-oriented complex task assignment problems are NP-hard problems [9]. Therefore, how to allocate a large number of complex tasks is a great challenge.

Therefore, this paper proposes a new crowdsourcing collective task assignment model. Our allocation mechanism is not to allocate tasks one by one, but to try to allocate all tasks to workers globally, that is, to consider the needs of unassigned tasks in the crowdsourcing platform and the skill characteristics of workers within a period of time, and allocate tasks to the most suitable group of workers. The advantage of this allocation model is that it allocates as many tasks as possible, so it is more suitable for large-scale concurrent crowdsourcing tasks. In order to improve the successful assignment rate of tasks, this paper considers the needs of tasks globally and organizes workers reasonably for collective task assignment, and abstracts the collective assignment problem into a weighted bipartite graph matching model by treating tasks and workers as different edge sets, and then this paper proposes to maximize the comprehensive benefits of the crowdsourcing worker group as the optimization goal, so that the collective allocation problem can be transformed into a global optimization problem. In order to solve this problem, this paper proposes the globally optimal allocation scheme based on the Kuhn-Munkres (KM) algorithm. Finally, this paper conducts comparative experiments on real datasets, and the results show that the proposed model has better performance in terms of task success rate and worker reward.

2. Problem model

This paper conducts collective task allocation by considering the time cost of tasks, the opportunity cost of workers, the utility of task allocation and other factors in an overall manner, with the aim of successfully assigning as many complex tasks as possible. The descriptions and definitions of the elements included in the above questions are given below.

Crowdsourcing tasks are expressed as $T = \{t_1, \dots, t_m\}$, each task t_i has four attributes, expressed as $t_i =< S_{t_i}, b_{t_i}, a_{t_i}, e_{t_i} >$, where $S_{t_i} = \{s_1, \dots, s_k\}$ represents the k different skill sets required by the task t_i; b_{t_i} represents the budget for a_{t_i} completing the task; t_i represents the time when the task t_i is released; e_{t_i} represents the deadline for the task. Crowd workers are denoted as $W = \{w_1, \dots, w_n\}$, and workers w_j are usually defined in the form of a quadruple $w_j =< S_{w_j}, r_{w_j}, p_{w_j}, d_{w_j} >$ as follows. Among them: $S_{w_j} = \{s_1, \dots, s_j\}$ it represents the skill set possessed by the worker; r_{w_j} represents the return expected by the worker to complete the task; the time when the worker w_j completed the last task is p_{w_j}, and the current time of the worker is d_{w_j}.

Definition 1 (Waiting cost of task) The task waiting time means the time from when l_{t_i} the crowdsourcing task t_i is released on the platform to when the task is assigned to the worker, we call it the task waiting time. The AT_{t_i} denotes time cost, the longer the task wait time, the higher the assignment priority. Knowing that the cost per unit time is la_{t_i}, the final AT_{t_i}calculation is:

$$AT_{t_i} = \left(l_{t_i} - a_{t_i}\right) \cdot la_{t_i} \tag{1}$$

Definition 2 (Opportunity cost of worker) The time from the time when the worker w_j completes the last task to the time when the worker is currently waiting for the assignment is called the waiting time of the worker. To simplify the calculation, this paper defines $w_j$the cost per_{w_j} per unit time of the worker as. Therefore, this article uses the term co_{w_j} to express the cost of the waiting time of workers, which is called the opportunity cost of workers.

$$co_{w_j} = (d_{w_j} - p_{w_j}) \cdot per_{w_j} \tag{2}$$

In this way, the higher the worker's opportunity cost, the higher the priority of his assignment. Usually, complex tasks require the cooperation of multiple workers. This paper assigns tasks in the form of worker groups, expressed as $group_{t_i} = \{w_j \big| < t_i, w_1 >, \dots, < t_i, w_j >\}$, where it represents $mp =< t_i, w_j >$ the combination of crowdsourcing tasks t_i and task workers. The basic conditions for successful task assignment need to be satisfied by the worker $(\bigcup_{\sum \forall w_j \in group_{t_i}} S_{w_j}) \supseteq S_{t_i}$ group and $\sum_{\forall w_j \in group_{t_i}} r_{w_j} \leq b_{t_i}$ the workers need to be online at the same time, and the set of all worker groups is denoted as $Group = \{group_{t_1}, \dots, group_{t_i}\}$.

Definition 3 (Assignment Utility) The utility value of each task assigned to the worker group is different, so the utility of each worker in the worker group to complete the task is also different. In this paper, the distribution utility is used c_{ij} to represent the utility generated by assigning crowdsourcing tasks t_i to crowdsourcing workers w_j, that is, the utility of matching pairs $mp =< t_i, w_j >$, which is

ISSN 2616-5775 Vol. 6, Issue 5: 14-20, DOI: 10.25236/AJCIS.2023.060503

defined as the product of the task budget and b_{t_i} the skill coverage of workers, as:

$$c_{ij} = b_{t_i} \cdot \frac{|S_{t_i} \cap S_{w_j}|}{\sqrt{|S_{t_i}| \cdot |S_{w_j}|}} \tag{3}$$

This c_{ij} is determined by the task's budget and worker's skill coverage. When workers have the same skill coverage, the larger the budget of the task, the greater the allocation utility, and the higher the priority of the worker assigned to this task.

Definition 4 (Comprehensive benefit of crowdsourcing worker group): The task allocation method of the worker group involves four indicators: allocation utility, task's waiting cost for allocation, worker's opportunity cost, and worker expected reward. In order to reasonably organize workers to complete more tasks, we define the comprehensive benefits of worker groups. This can reflect the matching degree of the worker group to the task, which is expressed as follows:

$$CS_{group_{t_i}} = \frac{\alpha \sum c_{ij} + \beta \sum co_{w_j} + \gamma AT_{t_i}}{\sum r_{w_j}} \qquad w_j \in group_{t_i} \tag{4}$$

The greater $group_{t_i}$ the expected return $\sum r_{w_j}$, the smaller the overall benefit of the worker group, and the smaller the priority assigned to it. When the worker group allocation utility $\sum c_{ij}$, task waiting time cost $\sum co_{w_j}$ and worker opportunity cost AT_{t_i} are greater, the priority of allocation is higher.

Collective allocation model: Therefore, the collective allocation problem can be transformed into a global optimization problem. That is, given the crowdsourcing task set *T*, the crowdsourcing worker set W and the comprehensive benefit calculation function of the crowdsourcing worker group, from any possible distribution scheme M, seek an allocation scheme that maximizes the comprehensive benefit of the crowdsourcing worker group $CS_{group_{t_i}}$, which we use M^* to represent, formally expressed as:

$$M^* = argmax_M \sum\nolimits_{\forall group_{t_i} \in M} CS_{group_{t_i}} \tag{5}$$

$$Subject\ to: \quad d_{w_j} < e_{t_i} \tag{6}$$

$$x_{w_j} \in [0,1] \tag{7}$$

$$(\cup_{\sum \forall w_j \in group_{t_i}} S_{w_j}) \supseteq S_{t_i} \tag{8}$$

$$\sum\nolimits_{\forall w_j \in group_{t_i}} r_{w_j} \le b_{t_i} \tag{9}$$

Final distribution plan M^* is composed of worker groups, and $group_{t_i}$ the worker group corresponding $group_{t_i}$ to each task needs to meet four basic constraints: (6) Time constraints: crowdsourcing tasks t_i and the worker's current waiting time d_{w_j} must be assigned before the deadline of the task, otherwise the assignment cannot be completed e_{t_i}; (7) Assignment constraints: a worker can only be assigned to complete one task at the same time $x_{w_j} = 0$ or $x_{w_j} = 1$ task; (8) Skill constraint: $group_{t_i}$ the skill set in the worker group needs to cover the skill demand of the task; (9) Cost constraint: $group_{t_i}$ the expected return of the worker in the worker group needs less than the task budget.

Theorem 1: It is NP-hard to select a worker group from the worker group W and meet the above four constraints to achieve the maximum comprehensive benefit of the worker group.

Proof: The 0-1 knapsack problem is a classic NP-hard problem: Given a set, the benefit value and weight of each element are known, and on the premise that the total weight does not exceed the budget, select a subset from the set to achieve Maximum benefit. Therefore, this article is looking for a subset of worker groups in the worker set, which can achieve the maximum comprehensive benefit of the worker group under the cost constraint of (9), and satisfy the conditions of (6), (7), and (8). Thus, theorem 1 is proved.

3. Method

3.1. Candidate task worker groups

In this section, we propose an algorithm for candidate task worker groups based on assignment utility. Firstly, a task assignment period is defined as the task assignment period from the task minimum assignment time to the maximum task deadline in the θ current crowdsourcing platform to obtain the set of tasks and workers; then obtain the task-worker matching pairs with skill coverage for each task and worker, and sort the assignment utility; Finally, use the greedy idea to obtain the matching pairs of task workers for all tasks. The specific steps are shown in the table below:

Algorithm 1: Candidate worker Group algorithm based on allocation utility
Input: All tasks and θ workers W_θ online during the allocation period T_θ.
$Group_\theta \leftarrow \emptyset$
While $T_\theta \neq \emptyset \; and \; W_\theta \neq \emptyset$ do:
For $t_i \in T_\theta$ do:
$wt_{t_i} \leftarrow \emptyset,\ W'_\theta \leftarrow W_\theta,\ group_{t_i} \leftarrow \emptyset$
For $W'_\theta \neq \emptyset \; and \; w_j \in W_\theta$ do:
If $S_{t_i} \cap S_{w_j} \neq \emptyset$ and $d_{w_j} < e_{t_i} \; and \; r_{w_j} \leq b_{t_i}$
$wt_{t_i} \leftarrow w_j$
End
$temp_{work} \leftarrow \emptyset, temp\_group \leftarrow \emptyset$
while $wt_{t_i} \neq \emptyset$ do:
For $w_j \in W_\theta$ do:
If $S_{t_i} \cap S_{w_j} \supseteq S_{t_i}$
$temp_{group} \cup w_j,\ group_{t_i} \cup \ temp\_group,\ W_\theta = W_\theta - w_j$, break
Else
$temp\_group \cup w_j$, $W_\theta = W_\theta - w_j$;
End
End
$Group_\theta \cup group_{t_i}$
End
Output: All $Group_\theta$

From the above algorithm, we can know that the complexity of the algorithm is reduced to O ($|T_\theta| \cdot |W_\theta|^2$) . The task can be completed by multiple worker groups, and a worker group can complete multiple tasks, but a worker can only assign one task in an allocation cycle. Therefore, this paper proposes an algorithm based on KM to maximize the comprehensive benefits of worker groups to complete the optimal matching scheme.

Algorithm 2: The maximizing the comprehensive benefit of workers group based on KM method
1) **Input:** tasks T_θ and $Group_\theta$ workers composed of W_θ
2) For $group_{t_i} \in Group_\theta$ do:
3) $CS_{group_{t_i}} = \frac{\alpha \sum c_{ij} + \beta \sum co_{w_j} + \gamma AT_{t_i}}{\sum r_{w_j}}$
4) Build a bipartite graph $G = \left(T, Group_\theta, E, CS\right)$:
5)   Initialize the feasible label $A[i] \leftarrow CS_{group_{t_i}}$, B $[i] \leftarrow 0$, satisfy $A[i] + B[j] \geq CS_{group_{t_i}}$
6)   Find an augmenting path to get a match $M^* \leftarrow M_0$until no augmenting path is found
7)   Judging constraints and recording used workers
8)   Update feasible markup:$d = \min\{A[i] + B[j] - CS_{group_{t_i}}\}$
9)   Repeat steps 5, 6, 7 until a complete match is found M^*
10) **Output:** result $M \leftarrow M^*$

3.2. Based allocation method of maximizing comprehensive benefits of workers group

From the content of the previous section, we can get the set of task worker groups and build a bipartite graph $G = (T, Group, E, CS)$: T represents the task set within $Group$; θ represents the worker group; E represents the matching pair of task worker groups; CS represents the comprehensive benefits of the worker group tasks, that is the weight of the edge. Then we calculate the comprehensive benefits of the worker group $CS_{group_{t_m}}$. In order to make the bipartite graph have a complete matching, we add edges to the tasks that the worker group cannot complete, and assign their weights to 0. Finally, we use the standard KM algorithm to obtain the optimal matching scheme M, and its matching conditions need to meet constraints such as time, allocation, skills, and cost. The specific algorithm flow is shown in the

ISSN 2616-5775 Vol. 6, Issue 5: 14-20, DOI: 10.25236/AJCIS.2023.060503

table below:

Its algorithm complexity is O ($|T_\theta|\cdot|W_\theta|$). In addition, there may be many situations in the matching scheme that can obtain the maximum comprehensive benefit of the worker group through the above method, and we will set different conditions in the experiment to choose.

4. Experiment and Analysis

The experiments in this article are implemented using Python language, and the environment is python3.7, Intel (R) Core (TM) i7-7500U CPU@2.30GHz and 8G memory. In addition, the experimental data is crawled from the www.freelancer.com website, and its task characteristics are complex tasks with 1-5 skills (such as java, python, etc.), which require multiple workers with 1-5 skills to work together. Meet the needs of multi-skill complex tasks.

Table 1: The impact of changes in verification parameters on task assignment.

Assign utility parameters	**Task waiting for allocation time cost parameter**	**Worker Opportunity Cost Parameters**	**Number of successfully assigned tasks**
α	β	γ	
0.0	0.0	1 .0	48
	0.5	0.5	37
	0.8	0.2	34
	1 .0	0.0	40
0.4	0.0	0.6	813
	0.2	0.4	1062
	0.4	0.2	995
	0.6	0.0	883
0.8	0.0	0.2	1628
	0.1	0.1	1725
	0.15	0.05	1562
	0.2	0.0	1603
1 .0	0.0	0.0	1769

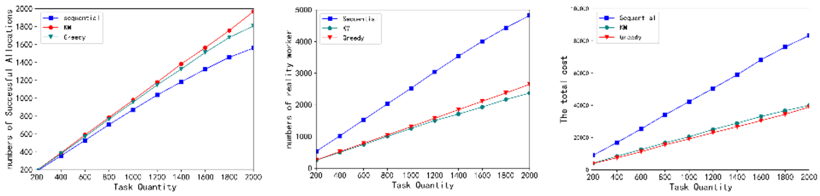
2000 tasks and 5000 workers are randomly selected from the dataset within one allocation period θ for the experiment. The parameters involved in the model in this paper come from formula (4), which are distribution utility parameters α, task waiting time cost parameters β, and worker opportunity cost parameters respectively γ. First, we verify the impact of parameter changes on collective task allocation, and the results are shown in Table 1: when α= 0, no matter how $\gamma$the parameter sum β changes, the number of task allocation successes is very low.

From the above analysis, we can see that the parameters of the collective allocation model are mainly affected α. The experiment in this paper will select the distribution utility parameter, task waiting time cost parameter, worker opportunity cost parameter which are, $\beta = 0.1$, γ= 0.1 respectively $\alpha = 0.8$, and from the number of tasks, the size of the number of tasks, the complexity of the task, the budget of the task, the worker's expected reward, Different sample data six aspects to test. The test results are measured by three indicators: the number of successfully assigned tasks, the actual number of workers assigned, and the cost of paying workers for tasks. In addition, the collective allocation method based on *KM* in this experiment will be compared with sequential allocation and greedy allocation methods.

4.1. Increasing tasks quantity

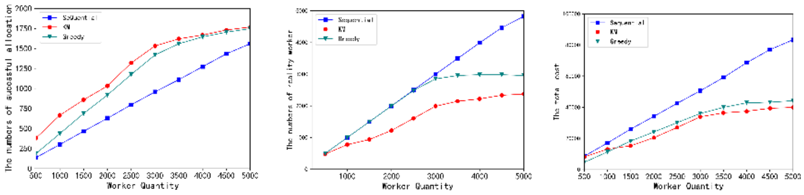
In this experiment, we first test the influence of changes in the number of tasks and the results are shown in Figure 1: in Figure 1 (a) our method can allocate more task, and Figure 1 (b), (c) requires fewer workers and pays less than other methods. One of the reasons is that the method in this paper considers the distribution of workers and task characteristics globally, and assigns tasks to relatively more suitable workers to complete in the form of worker groups, rather than selecting the optimal worker to complete. Therefore, through such a reasonable organization, more workers are released to complete other tasks, and the task completion rate is improved while reducing the overall payment cost.

ISSN 2616-5775 Vol. 6, Issue 5: 14-20, DOI: 10.25236/AJCIS.2023.060503



(a) *(b)* *(c)*

Figure 1: Increasing the number of tasks, the number of successful task assignments, the required workers, and the changes in the three indicators paid.



(a) *(b)* *(c)*

Figure 2: Increasing the number of workers, the number of successful task assignments, the required workers, and the changes in the three indicators paid.

4.2. Increasing worker quantity

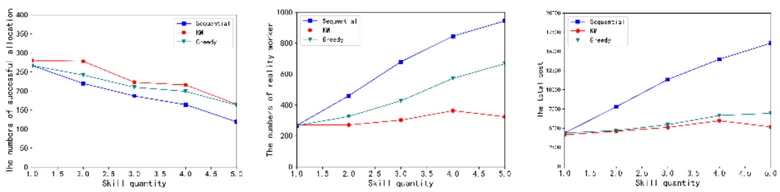
Testing the influence of changes in the number of workers on collective allocation, From Figure 2 (a) and (c), we have verified this conclusion. As the number of workers increases, the number of successful tasks increases, and the payment costs decrease; and Figure 2 (b), we can see that the actual number of workers required less than other methods. Therefore, our method not only rationally organizes workers and successfully assigns more tasks, utilizes fewer workers but also reduces the total reward of workers.

4.3. Increasing task complexity

The experiments in this section test the effect of task complexity on collective assignment. The complexity of the task is expressed by the number of skills required for the task, and the greater the number of skills required, the more complex the task. The results are shown in Figure 3: Figure 3(a) shows that under the same constraints, the number of collective assignment successes in our method is higher than that obtained by other methods; Figure 3 (b), (c) show that our method actually requires fewer workers and requires less cost to complete the task. As the budget increases, all methods show a trend of increasing first and then stabilizing. The reason is that the budget of the task has exceeded the compensation of some worker groups, and such worker groups can be assigned to the task.

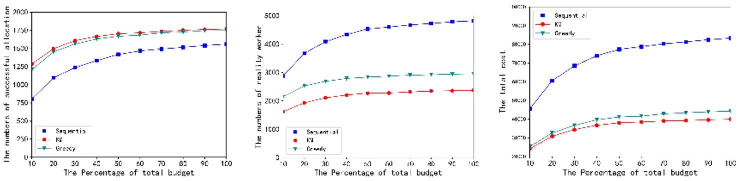
4.4. Increasing task budget

It is observed from the dataset that the task budget is much higher than the worker's expected compensation. Therefore, the experiment in this section uses the percentage of the original budget to test the impact of increasing the budget on collective allocation, and the results are shown in Figure 4: With the increase of the budget, the number of successfully assigned tasks, the number of workers required for tasks, and the cost of workers required for tasks are all equal. It shows a trend of increasing first and then leveling off; while Figure 4 (a), (b) and (c) show that under the same constraints, our method is superior to other methods. The reason is that the budget of the task has exceeded the return of the worker group. When the budget is increased, only a small number of worker groups that are below the budget can be assigned to the task.



(a) *(b)* *(c)*

Figure 3: Increasing the complexity of the task, the change of the three indicators of the number of successful task assignments, required workers, and paid expenses.



(a) *(b)* *(c)*

Figure 4: Increasing the budget of the task, the change of the three indicators of the number of successful task assignments, required workers, and paid expenses.

5. Conclusions

There are many unassigned complex tasks and workers in the crowdsourcing system, as well as the low success rate of a large number of complex tasks, so this paper proposes a new collective allocation model to solve the shortcomings, the allocation model mainly has two Advantages: First, it allocates as many tasks as possible, which is more suitable for the situation where there are many unassigned tasks in the platform; second, it assigns tasks to more suitable workers through overall consideration of tasks and workers, which promotes the application of crowdsourcing develop. Finally, this paper conducts extensive experiments on real crowdsourcing datasets. Experimental results show that compared with previous methods, our proposed model and collective assignment method based on KM algorithm can always improve the success rate of task assignment and reduce the total cost of task payment.

6. Acknowledgments

This work was supported by the Postgraduate Research & Practice Innovation Program of Jiangsu Province of China under Grant KYCX21_1532 (ZJWXW21001). The article is an extension of our early conference paper [10].

References

[1] Zhang Z., Kui J., Xie X., Zhou Y. Crowdsourcing quality control strategy and evaluation algorithm. Chinese Journal of Computers, 36(08).1636-1649 (2013).
[2] Li S., Wei M., Huang S. Deep generative crowdsourcing learning using annotator correlation. Journal of Software, 33(4). 1274-1286 (2022).
[3] Drapeau R., Chilton L., Bragg J., Weld D. Microtalk: Using argumentation to improve crowdsourcing accuracy. In. Proceedings of the 4th AAAI Conference on Human Computation and Crowdsourcing, vol. 4, pp. 32–41(2016).
[4] Chittilappilly A. I., Chen L., Amer-Yahia S. A survey of general-purpose crowdsourcing techniques. IEEE Transactions on Knowledge and Data Engineering 28(9), 2246–2266 (2016).
[5] Liu D., Hu H., Wu D. Weighted Network Modeling and Module Partitioning Among Crowdsourcing Design Tasks for Social Product Development. Industrial Engineering Journal, 24(5). 95-100 (2021).
[6] Jiang J., An B., Jiang Y., Lin D., Bu Z., Cao J., Hao Z. Understanding crowdsourcing systems from a multiagent perspective and approach. ACM Transactions on Autonomous and Adaptive Systems 13(2), 1–32 (2018).
[7] Wang W., Jiang J., An B., Jiang Y., Chen B. Toward efficient team formation for crowdsourcing in noncooperative social networks. IEEE Transactions on Cybernetics 47(12), 4208–4222 (2017).
[8] Jiang J., An B., Jiang Y., Zhang C., Bu Z., Cao J. Group-oriented task allocation for crowdsourcing in social networks. IEEE Transactions on Systems, Man, and Cybernetics. Systems 51(7), 4417–4432 (2021).
[9] Cheng P., Lian X., Chen L., Han J., Zhao J. Task assignment on multi-skill oriented spatial crowdsourcing. IEEE Transactions on Knowledge and Data Engineering 28(8), 2201–2215 (2016).
[10] Zhang J., Wei J. Research on Crowdsourcing-oriented Global Complex Task Assignment Based on Artificial Intelligence. 2022 2nd International Symposium on Artificial Intelligence and its Application on Media, pp.70-74(2022).