

Received July 31, 2020, accepted August 13, 2020, date of publication September 2, 2020, date of current version September 17, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.3021286

# Supply-Demand Matching in Non-Cooperative Social Networks

SHAO-JIE ZHANG<sup>1</sup>, XU-DONG LU<sup>1,2</sup>, SHI-PENG WANG<sup>1,2</sup>, AND WEI GUO<sup>1,2</sup>

<sup>1</sup>School of Software, Shandong University, Jinan 250101, China

<sup>2</sup>Joint SDU-NTU Centre for Artificial Intelligence Research (C-FAIR), Shandong University, Jinan 250101, China

Corresponding author: Xu-Dong Lu (dongxul@sdu.edu.cn)

This work was supported in part by the National Key Research and Development Project of China under Grant 2019YFB1705904, and in part by the Science and Technology Development Plan Project of Shandong Province under Grant 2019JZZY020505, Grant 2019JZZY010109, and Grant 2018YFJH0506.

**ABSTRACT** The complex supply-demand matching problem is a kind of social service computing problem, which can be applied to the coordinated production of products or the supply of service. In this scenario, the demander needs a number of suppliers to provide services or products to complete a given task. The key to solving this problem is to build a supply network that covers the requester's requirement. Traditional collaboration issues in social network mainly focused on the "team formation problem", that is to build a team that covers all the skills required for the task. However, due to the complex characteristics of supply-demand matching problems in the application of social services, the team formation method is limited and inefficient, and there is no special solution for the complex supply-demand matching problems in social network. This paper proposes a general framework to solve the complex matching problem of supply and demand. On the premise of non-cooperative constraints, social networks are used to build supply networks with low communication loss, and the unnecessary cost is reduced through cooperation.

**INDEX TERMS** Coordination, non-cooperative, social networks, strategy design, supply-demand matching.

## I. INTRODUCTION

This paper studies the complex matching problem of supply and demand in Non-cooperative social networks (SNs). The demander wants to build a supply network that covers the task demand in both type and quantity. Unlike the previous "team formation problem" [1]–[11], we had to redesign the solution because of the following characteristics of the supply-demand matching problem: ① The same service or product requirement in a task can be segmented to multiple providers to provide collaboratively; ② Different from contributing skills, suppliers of services or products may face capacity caps; ③ A single supplier can supply multiple products and services to multiple demanders simultaneously. Besides, due to the selfishness of members in practical application [12], [13], the scheme must be feasible under the constraints of non-cooperation. At the same time, considering that individuals with closer social relations are more likely to have closer geographical and linguistic connections and a higher level of trust [14], [15], SNs will be used to select

suppliers with closer social relations with the employer, to reduce the cost of communication [16]–[20].

Although there are many ways to use SNs to form professional collaborative team [4], [21], [22], there are still some defects and some practices that are not applicable to this supply and demand problem: ① these methods ignore the differences in social relationship quality caused by the differences in SNs structure. For example, the quality of two SNs composed of a complete graph and its minimum spanning tree cannot be generalized; ② privacy policy leads to the imbalance of network information, which hinders a better employment plan (reduce the total cost of each task) in the whole SNs; ③ different from the professional cooperative team, many participants in the same supply network have no cooperative relationship, and they supply products to the demander independently. Therefore, an effective supply-demand matching method should evaluate the social relationship between the supplier and the demander, expand the supply network by using some information of member neighbors, and pay attention to global optimization.

In this context, we design a complete set of supply-demand matching method, including ① a set of distributed

The associate editor coordinating the review of this manuscript and approving it for publication was Leyi Wei.

negotiation-based supply networks formation algorithm, which allows the supplier and the demander to decide whether to cooperate or not and agree on a quotation in line with the interests of both parties, so as to initially build a supply network covering the task demand; ② preference algorithm, which combines the relationship between nodes in SNs, describes the impact of trust and communication issues on task cost; ③ cooperate algorithm, which is used to further reduce unnecessary cost loss and achieve the goal of global optimization after the initial construction of the supply network.

The theoretical analysis shows that the proposed method can solve the complex supply and demand problem on the premise of satisfying members' selfishness and considering communication cost, and optimize the cost of individuals and groups at the same time. Finally, a series of experiments are set up to verify the improvement of this method compared with the traditional Contract Net (CN) method and adjust the parameters to simulate different task scenarios to observe the impact of condition changes on the performance of the algorithm. The experimental results show that: ① compared with the traditional CN method, the method proposed in this paper can effectively reduce the cost; ② the proposed method performs better in the scenario of low supply cost and small demand for a single product.

## II. RELATED WORK

### A. TEAM FORMATION PROBLEM

As mentioned above, there are differences between "supply-demand matching problem" and "team formation problem", but current researches on the latter can provide references for the former. The goal of the "team formation problem" is to form a team of experts covering all the skills required for the task. Considering the evolution, emergent behavior, operational independence, and management independence of the systems, Lim and Ncube [22] developed a method of team building based on the stakeholders' recommendation, and proposes using social networks and crowdsourcing to identify and prioritize the stakeholders of systems projects. Taking into account the impact of social relations on team cooperation, Wang *et al.* [21] used the social neighborhood information of members to expand the connectivity graph to build a team and designed a mechanism based on distributed negotiation to improve social welfare. Different from the methods of searching experts in the whole SN, Sun *et al.* [8] proposed a team formation model that outsource tasks to social networks, and selected the list of centrality experts as the seed, so as to reduce the communication cost of the team and narrow the search space.

Instead of targeting individuals, Chamberlain [2] first proposed the concept of Groupsourcing. In Groupsourcing, tasks are assigned to a group of people with different expertise who were connected through social networks. Compared with other methods, Groupsourcing offers a high-precision, data-driven, and low-cost method. In Groupsourcing, every

time a complex task is published, a new team would not be formed from scratch to satisfy the skill requirements of the task. Based on Chamberlain's research, Jiang *et al.* [3] formally defined the context-aware task allocation problem in group-oriented crowdsourcing, proposed a heuristic context-aware task allocation approach, and proposed a modeling method for natural worker groups in crowdsourcing, including groups with and without leadership. Besides, instead of solving the problem from the requestor's perspective, Lykourentzou *et al.* [6] explored a "team dating" strategy, which is a self-organized group team formation method. In this method, employees try and evaluate different candidate partners. Rokicki *et al.* [7] explores a cohesive strategy that includes self-organization. In this strategy, workers (initially in the form of a one-man team) can decide which team they want to join or who could join their team.

### B. TEAM REVENUE OPTIMIZATION

Common methods to improve team profitability include reducing personnel and communication costs, improving team members' quality, and Optimizing the rationality of task allocation.

Complex tasks can be decomposed into smaller subtasks, which can be executed either sequentially or in parallel by workers. In order to build a high-quality team by rationalizing task requirements, Jiang and Matsubara [23] demonstrated the superiority of vertical task decomposition over horizontal task decomposition in improving the quality of the task's solution, and clearly explained the optimal vertical task decomposition strategies under two revenue sharing schemes, which maximized the quality of task solutions.

Tran-thanh *et al.* [9] studied the issue of how to hire higher-quality experts on a limited budget, they redesigned the classic multi-armed Bandit (MAB) model to solve this problem. An algorithm called **bounded  $\epsilon$ -first** was proposed, it uses the first  $\epsilon B$  of its total budget  $B$  to derive estimates of the workers' quality characteristics (exploration), while the remaining  $(1 - \epsilon)B$  is used to maximize the total utility based on those estimates (exploitation). Tran-Thanh also developed another BudgetFix algorithm [24], which determines the number of interdependent micro-tasks and the price to pay for each task given budget constraints. Moreover, BudgetFix provides quality guarantees on the accuracy of the output of each phase of a given workflow.

Wolf *et al.* [25] were the first to realize that SNs plays an important role in teamwork, and social connections among social individuals might represent collaboration relationships (e.g., collaborate on common tasks previously). The advantage of using these SNs is that the social individuals who have worked together previously are estimated to work effectively as a team without much coordination overhead [16], [17]. Therefore, scholars consider establishing a cooperative team in SNs [18]–[20], so that the members of the team can form a connected graph, and work together effectively. Considering the selfishness of the members of SNs [12], [13], Wang *et al.* [21] further explored a team-

building approach adapted to non-cooperative constraints. They model each individual as a selfish entity, using a negotiating mechanism to cut costs and improve social welfare.

### III. PROBLEM DESCRIPTION

First, Social network  $SN = \langle A, E \rangle$  is an unweighted undirected graph, where  $A = \{a_1, a_2, \dots, a_m\}$  is the set of all member nodes in the graph,  $\forall (a_i, a_j) \in E$  indicates that there is a social relationship between nodes  $a_i$  and  $a_j$ .  $\forall a_i \in A$  is defined by 4-tuple  $\langle G(a_i), M(a_i), C(a_i), N(a_i) \rangle$ . Where  $G(a_i) = \{g_1, g_2, \dots\}$  represents the type of product that  $a_i$  can supply;  $M(a_i) = \{\max(a_i, g_1), \dots, \max(a_i, g_{|G(a_i)|})\}$  indicates the maximum upper limit of  $g_j \in G(a_i)$  supplied by  $a_i$ ;  $C(a_i) = \{c(a_i, g_1), \dots, c(a_i, g_{|G(a_i)|})\}$  is the cost of  $a_i$  supply unit  $g_j \in G(a_i)$ ;  $N(a_i)$  is the direct social neighbors of  $a_i$ , that is,  $N(a_i) = \{a_j | (a_i, a_j) \in E\}$ .

Task  $t$  is defined by 4-tuple  $\langle I_t, G(t), R(t), E(t) \rangle$ . Where  $I_t$  is the  $t$  task requester;  $G(t) = \{g_1, g_2, \dots\}$  indicates the type of all product supply needed to complete task  $t$ ;  $R(t) = \{r(t, g_1), \dots, r(t, g_{|G(t)|})\}$  indicates the demand of task  $t$  for various product supply;  $E(t) = \sum_{j=1, \dots, |G(t)|} e(t, g_j)$ , indicates the total value (not profit) that each product supply in task  $t$  can bring to  $I_t$ .

The supply network  $N_t$  for task  $t$  is defined by  $\langle t, \Omega_t, O(t), Us(t) \rangle$ . Where  $\Omega_t$  represents the set of employed members of  $N_t$ ;  $O(t) = \{(t, a_i, g_j, q(a_i, g_j, t), p(a_i, g_j, t), \eta(I_t, a_i)), \dots, (t, a_p, g_q, q(a_p, g_q, t), p(a_p, g_q, t), \eta(I_t, a_p))\}$  refers to the set of order details contained in  $N_t$ , where  $q(a_i, g_j, t)$  refers to the quantity of  $g_j$  supplied by  $a_i$  for task  $t$  in the order.  $p(a_i, g_j, t) = q(a_i, g_j, t) * c(a_i, g_j)$  indicates  $a_i$ 's payment for supplying  $g_j$  for task  $t$ , i.e.  $a_i$ 's salary.  $\eta(I_t, a_i)$  is the communication cost coefficient between  $I_t$  and  $a_i$ ;  $Us(t) = \{us(t, g_j, q_{us}(t, g_j)), \dots, us(t, g_q, q_{us}(t, g_q))\}$  is the unmet requirement of task  $t$ , where  $q_{us}(t, g_j) = r(t, g_j) - \sum_{(t, ai, gj, \dots) \in O(t)} q(a_i, g_j, t)$  is the unmet requirement of  $g_j$  in  $t$ .

In particular, there are some concepts that will be used many times in this paper. Although they can be expressed by the above-mentioned defined symbols, the symbol definitions still gave for the convenience of use:  $\mu(t, a_i, g_j) = q(a_i, g_j, t) / r(t, g_j)$ ,  $\mu(t, a_i, g_j) \in (0, 1]$ , which represents the ratio of the quantity of  $g_j$  supplied by  $a_i$  to  $r(t, g_j)$  in the contract of task  $t$ ;  $\lambda(t, g_j)$  is the ratio of current remaining unallocated demand in  $t$ 's sub demand  $g_j$  to  $r(t, g_j)$ . That is to say,  $\lambda(t, g_j) = 1 - \sum_{(t, ai, gj, \dots) \in O(t)} q(a_i, g_j, t) / r(t, g_j)$ ,  $\lambda(t, g_j) \in [0, 1]$ .  $th(t, g_j)$  is the threshold value to measure whether it is in its interest to employ a node to supply  $g_j$  for task  $t$ .  $th(t, g_j) = E(t) / r(t, g_j)$ .

The problem is that given task  $t$  and  $SN = \langle A, E \rangle$ , task initiator  $I_t$  wants to build  $N_t = \langle \Omega_t, O(t), \emptyset \rangle$ , so that task  $t$  is completely covered by  $N_t$  in the type and quantity of requirements;  $N_t$  should not contain redundant members, and each member should provide at least one product supply; reduce the cost as much as possible, in order to increase the revenue  $Pro(t) = E(t) - \sum_{(t, ai, gj, \dots) \in O(t)} q(a_i, g_j, t) * (1 + \eta(I_t, a_i)) * c(a_i, g_j)$ .

TABLE 1. Definition of notation.

Notation	Definition
$a_i$	node $i$
$N(a_i)$	direct social neighbors of $a_i$
$G(a_i)$	the type of product that $a_i$ can supply
$g_j$	demand $j$
$M(a_i)$	the maximum upper limit of various kinds of products supplied by $a_i$
$\max(a_i, g_j)$	maximum upper limit of $g_j$ supplied by $a_i$
$C(a_i)$	the cost of $a_i$ supply unit products
$c(a_i, g_j)$	the cost of $a_i$ supply unit $g_j$
$t$	task $t$
$I_t$	initiator of task $t$
$G(t)$	the type of all product supply needed to complete task $t$
$Pro(t)$	the profit that task $t$ can bring to $I_t$
$R(t)$	the demand of task $t$ for various product supply
$r(t, g_j)$	the demand of task $t$ for $g_j$
$E(t)$	the total value (not profit) that task $t$ can bring to $I_t$
$e(t, g_j)$	the value that $g_j$ can bring to $I_t$ in task $t$
$N_t$	the supply network of task $t$
$\Omega_t$	the set of employed members of $N_t$
$O(t)$	the set of order details contained in $N_t$
$q(a_i, g_j, t)$	the quantity of $g_j$ supplied by $a_i$ for task $t$ in the order
$p(a_i, g_j, t)$	$a_i$ 's payment for supplying $g_j$ for task $t$
$\eta(I_t, a_i)$	the communication cost coefficient between $I_t$ and $a_i$
$Us(t)$	unmet requirement of task $t$
$q_{us}(t, g_j)$	the unmet requirement of $g_j$ in task $t$
$\mu(t, a_i, g_j)$	the ratio of the quantity of $g_j$ supplied by $a_i$ to $r(t, g_j)$ in the contract of task $t$
$\lambda(t, g_j)$	the ratio of current remaining unallocated demand in $t$ 's sub demand $g_j$ to $r(t, g_j)$
$th(t, g_j)$	the threshold value to measure whether it is in its interest to employ a node to supply $g_j$ for task $t$

(1 +  $\eta(I_t, a_i)) * c(a_i, g_j)$ . The above symbol definitions are given in Table 1.

### IV. SUPPLY-DEMAND MATCHING IN NON-COOPERATIVE SOCIAL NETWORKS

In this paper, a distributed negotiation-based mechanism is used to make equal decision-making between supply and demand sides driven by interests, and a supply network covering task demand is constructed. In this process, the social relations exposed by the expansion of supply network are quantified to evaluate the cost of communication between SN nodes, which will have an impact on employment results. Finally, after the initial construction of the supply network, it coordinates with other task requesters in SN, exchanges some members of the supply network on the premise of meeting the interests of both sides, and improves the individual and group benefits. The above process is completed by three algorithms respectively, and the relationship between them is shown in Figure 1. We will describe these three algorithms respectively.

#### A. SUPPLY NETWORK CONSTRUCTION ALGORITHM

Input the demand of SN and  $I_t$ , and output a preliminary supply network in line with the interests of both the task supplier and demander. Before describing the algorithm, several role concepts should be first defined.

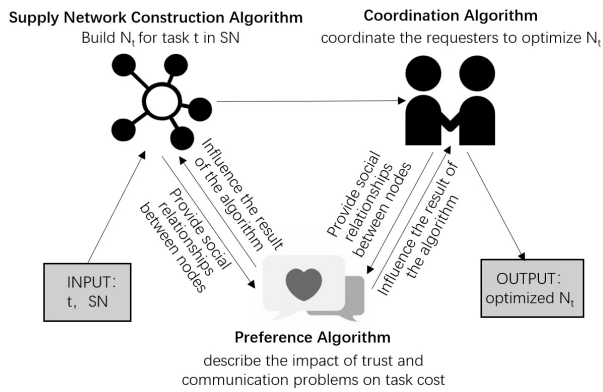


FIGURE 1. The relationships between the algorithms.

**Definition 1 (Freelancer, Contractor, Supplier):** For a product  $g_j$ , if the node  $a_i$  with  $g_j$  supply capability does not have  $g_j$  order at a certain time, then  $a_i$  is called freelancer; If  $a_i$  has  $g_j$  orders, but  $N_t$  has not been completed and  $g_j$  supply has not yet started,  $a_i$  is called contractor; after the start of supply and until the end of supply,  $a_i$  is called a supplier.

## 1) EXPANSION ALGORITHM

This algorithm drives the expansion of supply network.

### Algorithm 1 (Expand the Supply Network)

1. Initialize  $\Omega_t = I_t, \forall g_j \in G(t), q_{us}(t, g_j) = r(t, g_j), O(t) = \emptyset$
2. Initialize  $N_t = \langle \Omega_t, O(t), Us(t) \rangle$
3. For Each  $a_i \in \Omega_t$
4. For  $a_x \in N(a_i) \cup a_i$
5. setOptPre( $a_i, a_x$ )
6. decide( $N(t), a_x$ )
7. update  $N(t)$  and  $a_x$
8. if  $\forall g_j \in G(t), \lambda(t, g_j) = 0$
9. Terminate this Algorithm
10. End for
11. End for

The member initializing  $N(t)$  in step 1-2 only contains  $I_t$ , without any order, and the remaining unsatisfied demand equals the complete initial demand of  $t$ . Traverse each included member of the supply network  $N_t$  and their neighbor nodes (step 3-4), execute the decide algorithm (step 6) on it, and update the supply network  $N_t$  (step 7) according to the information returned by the decide algorithm until all the requirements of  $t$  are met (step 8-9). The algorithm involved in step 5 will be given in a “preference algorithm” later.

## 2) DECIDE ALGORITHM

This algorithm describes the details of step 6 in algorithm 1 and completes the decision-making of both the supplier and the demander in the way of three-stage distributed negotiation.

### a: OFFER STAGE

In this stage,  $I_t$  issues an order offer to the nodes that meet the requirements.

### Algorithm 2 (Decide-Offer Algorithm)

- /\* $Q_{temp}$  is the task amount of  $a_i$  in the negotiation;  
 $P_{temp}$  is  $a_i$ 's order compensation in the negotiation\*/
1. initialize  $th(t, g_j) = E(t)/r(t, g_j)$
  2. for each  $g_j \in G(a_i)$
  3. if  $c(a_i, g_j)^*(1+\eta(I_t, a_i)) \leq th(t, g_j)$
  4. if  $\lambda(t, g_j) = 1$
  5.  $\mu(t, a_i, g_j) = 1$
  6.  $Q_{temp} = \mu(t, a_i, g_j)^*r(t, g_j)$
  7.  $P_{temp} = Q_{temp}^*c(a_i, g_j)$
  8. send offer( $t, a_i, g_j, Q_{temp}, P_{temp}$ )
  9. else if  $0 \leq \lambda(t, g_j) < 1$
  10.  $\mu(t, a_i, g_j) = \lambda(t, g_j) + \sum \{ \mu(t, a_{exp}, g_j) | c(a_{exp}, g_j)^*(1+\eta(I_t, a_{exp})) > c(a_i, g_j)^*(1+\eta(I_t, a_i)) \ \&\& \ a_{exp} \in \Omega_t \}$
  11.  $Q_{temp} = \mu(t, a_i, g_j)^*r(t, g_j)$
  12.  $P_{temp} = Q_{temp}^*c(a_i, g_j)$
  13. send offer( $t, a_i, g_j, Q_{temp}, P_{temp}$ )
  14. end for

Step 1 Define the threshold  $th$  to measure whether an employment is in  $I_t$ 's interest. Traverse each product supply capacity  $g_j$  of  $a_i$  to determine whether its unit cost (if there is no special description, the “cost” in this paper includes product cost and communication cost) is less than  $th$  (Step 2-3). If it is, it will send  $g_j$  order offer to  $a_i$  in a “saturated” way. Its “saturation” is reflected in: ① when the  $g_j$  demand of task  $t$  has not been allocated, the order quantity in offer is all  $g_j$  demand (Step 4-8); ② when the  $g_j$  demand has been allocated, it tends to meet all the remaining unallocated  $g_j$  demand first, and then replace all the allocated orders (Step 9-13) in current  $N_t$  with higher cost than  $a_i$ .

### b: RESPONSE STAGE

In this stage,  $a_i$  responds to  $I_t$ 's offer according to its own situation. Several concepts of capacity should be clarified.

**Definition 2 (Free Capacity, Locked Capacity, Forbidden Capacity):** For product  $g_j$ , the surplus capacity of  $g_j$  owned by freelancer is called free capacity; the  $g_j$  capacity of the contractor is called locked capacity; the  $g_j$  capacity of the supplier is called forbidden capacity. Due to a supplier can only supply a particular product to one demander at a time, only the free capacity can be provided to any demander by supplier freely.

$a_i$  responds according to its  $g_j$  capacity type after receiving offer:

- Free capacity. After modifying the order quantity  $Q_{temp}$  to the smaller value of “its remaining  $g_j$  supply capacity  $\max(a_i, g_j)$ ” and “the order quantity  $Q_{temp}$  suggested by  $I_t$  in offer”,  $a_i$  make a positive response, agreement( $t, a_i, g_j, Q_{temp}, P_{temp}$ ), to accept the offer;

- Locked capacity. As an important basis for the follow-up “coordination algorithm”,  $a_i$  responds to  $I_t$  with  $\text{mark}(t, a_i, g_j, Q_{\text{temp}}, P_{\text{temp}}, t_2, q(a_i, g_j, t_2))$  to declare “the quantity of  $I_t$ 's orders that could have been accepted by  $a_i$  if  $a_i$  have not been employed by  $I_{t2}$ ”, that is,  $\min[Q_{\text{temp}}, \max(a_i, g_j) + q(a_i, g_j, t_2)]$ ;
- Forbidden capacity. Make a negative response to refuse the offer.

### c: CONFIRM STAGE

At this stage,  $I_t$  determines the result of this decision according to the response content of  $a_i$ :

- If the response is  $\text{agreement}(t, a_i, g_j, Q_{\text{temp}}, P_{\text{temp}})$ ,  $a_i$  will be allocated to supply the unmet  $g_j$  demand  $\lambda(t, g_j)$  in  $N_t$  first; if  $Q_{\text{temp}}$  can cover  $\lambda(t, g_j)$  and there is still surplus, continue to replace (or partially replace) the high-cost  $g_j$  orders in current  $N_t$  in the order of “high-cost orders  $\rightarrow$  low-cost orders”
- If the response is  $\text{mark}(t, a_i, g_j, Q_{\text{temp}}, P_{\text{temp}}, t_2, q(a_i, g_j, t_2))$ , record this mark for subsequent “coordination algorithm”;
- If the response is refusal, the employment of  $a_i$  to supply  $g_j$  will be abandoned.

## B. PREFERENCE ALGORITHM

Preference algorithm describes the impact of trust and communication problems on task cost according to the relationship between nodes in SNS. The key lies in:

- determine the optimal precursor;
- calculate the shortest distance;
- design an appropriate preference function to evaluate the communication loss according to the shortest distance.

*Definition 3 (Previous Supply Network, Precursor, Optimal Precursor, Shortest Distance):* Previous supply network ( $\text{pre\_}N_t$ ) of task  $t$  is a network that includes all the nodes that have been employed in the construction process of  $N_t$  and their precursors; In  $\text{pre\_}N_t$ , if the demander  $I_t$  can access  $a_i$ 's direct neighbor  $a_x$  through  $a_i$ , then  $a_i$  is called the “precursor” of  $a_x$ , which is recorded as  $a_i \in \text{pre}(t, a_x)$ ; Among the precursors of  $a_x$ , the one with the shortest distance from  $I_t$  is called “optimal precursor”——  $\text{opre}(t, a_x)$ ; In  $\text{Pre\_}N_t$ , the shortest distance  $\text{Dist}(t, a_x)$  between  $I_t$  and  $a_x$  is  $n$  if  $I_t$  can be reached by  $a_x$  after  $a_x$  visit optimal precursor iteratively  $n$  times.

### 1) DETERMINE THE OPTIMAL PRECURSOR

Step 5 of algorithm 1 gives the time to determine the optimal precursor of the node, which is determined by method  $\text{setOpre}(a_i, a_x)$ .

First, judge whether the visited node in Step 5 of algorithm 1 is its own precursor, if it is, do nothing (step 1-2); if not, record  $a_i$  as the precursor of  $a_x$ , traverse the precursor set of  $a_x$  to update  $a_x$ 's optimal precursor in task  $t$  (step 3-11). In this process, if the optimal precursor of  $a_x$  changes to

### Algorithm 3 (Determine the Optimal Precursor, $\text{setOpre}(a_i, a_x)$ )

---

```

1. if  $a_i == a_x$ 
2.   do nothing
3. else
4.    $\text{pre}(t, a_x) = \text{pre}(t, a_x) \cup a_i$ 
5.   for each  $a_y$  in  $\text{pre}(t, a_x)$ 
6.     if  $\text{Dist}(t, a_i) \leq \text{Dist}(t, a_y)$ 
7.        $\text{Opre}(t, a_x) = a_i$ 
8.        $\text{calDist}(t, a_x)$ 
9.   else
10.    do nothing
11. end for

```

---

$a_i$ , the shortest distance between  $a_x$  and  $I_t$  is recalculated by  $\text{calDist}(t, a_x)$  (step 8).

### 2) CALCULATE THE SHORTEST DISTANCE

Step 8 of algorithm 3 shows when to calculate the shortest distance between a node and  $I_t$ . The  $\text{calDist}(t, a_x)$  method is used to calculate: set a pointer at  $a_x$ , move the pointer in the direction of the optimal precursor of the current node until it reaches  $I_t$ . Then the number of the pointer moves are taken as the shortest distance between  $a_x$  and  $I_t$ .

### 3) PREFERENCE FUNCTION

*Definition 3 (Communication Cost Coefficient, Upper Limit of Communication Cost Coefficient):* The communication cost coefficient  $\eta(I_t, a_i)$  is calculated by the preference function, which describes the additional cost loss when network members cooperate; the upper limit of communication cost coefficient  $\eta_{\text{max}}$  is set because the communication loss will not increase endlessly with the increase of social relationship distance, and it must converge to a certain upper limit, which needs to be determined according to the actual application research. For example, the supply of precision instrument parts, such as aircraft instruments, requires more communication to ensure compliance with specifications and quality requirements, so the communication cost and  $\eta_{\text{max}}$  are both higher; for orders such as food packaging bags,  $\eta_{\text{max}}$  is lower because there is no need for many additional communication costs.

The preference function determines the communication cost coefficient between nodes according to the social relationship, which should meet the following constraints: ① the function should be in a positive proportion to the shortest distance between nodes; ② for the nodes with a long social distance, the change of communication cost caused by the increase of distance is no longer obvious, so the growth rate of function should decrease with the increase of distance, that is, the differential coefficient of the function decreases monotonically. ③ According to the theory of six degrees of separation, there are no more than six intermediate nodes between two points in social network, so the communication

cost coefficient should be close to the upper limit when the distance is 6.

With these constraints, it is found that the function  $-e^{-x} + 1$  can meet the above requirements well: the function monotonically increases on  $[0, +\infty]$ , the derivative gradually decreases to 0, the function finally converges to 1, and approaches the upper limit of the function when  $x = 6$ . The function image is shown in Figure 2:

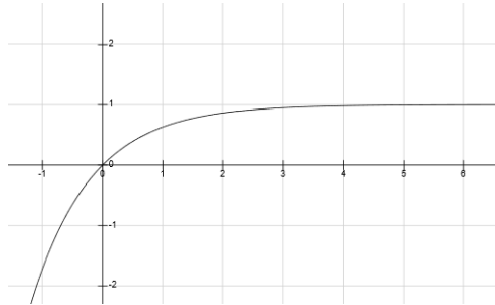


FIGURE 2.  $(-e^{-x}+1)$  function image.

In addition, to meet the requirement of upper limit  $\eta_{\max}$ , the preference function is defined as  $\eta(I_t, a_i) = \eta_{\max}^* (-e^{-x} + 1)$ , where  $x$  represents the shortest distance  $\text{Dist}(t, a_i)$  between  $a_i$  and  $I_t$ .

### C. COORDINATION ALGORITHM

In order to optimize the high-cost supply network caused by SNs privacy, the coordination algorithm coordinates the requesters with unbalanced social resources and exchanges their rights to employ the contractors, so as to improve the individual and group benefits at the same time.

The coordination algorithm is proposed based on this situation:  $I_t$  receives  $a_1$ 's response, mark  $(t, a_i, g_j, Q_{\text{temp}}, P_{\text{temp}}, t_2)$ , during the construction of  $N_t$ ,  $I_t$  has to give up  $a_1$  temporarily. After  $I_t$  builds  $N_t$ ,  $I_t$  finds that if  $a_1$  is employed to supply  $g_j$ , cost  $C_1$  could be saved compared with current  $N_t$ ; and for contractor  $a_1$ 's employer  $I_{t2}$ , if there were other alternative suppliers besides  $a_1$  to complete this part of  $g_j$  supply, and the increased cost after substitution  $C_2$  is smaller than  $C_1$ , there would be room for coordination between  $I_t$  and  $I_{t2}$ .

The execution time of coordination algorithm is after  $I_t$  completes "supply network construction algorithm". It traverses every mark response  $I_t$  receives in the process of  $N_t$  construction, executes coordination algorithm, and completes the coordination of two requesters in the way of three-stage distributed negotiation.

#### 1) REQUEST STAGE

In this stage,  $I_t$  decides whether and how to issue a coordination request to  $I_{t2}$  based on its own situation and the content of mark.

Based on the order quantity that  $a_i$  responded in mark,  $I_t$  updates  $N_t$  by "replacing the contractors whose unit supply cost of  $g_j$  is higher than that of  $a_i$  in the order of cost from

#### Algorithm 4 (Cooperate-Request Algorithm)

```

1. The nodes  $a_{\text{exp}}$  in  $\{a_{\text{exp}} | c(a_{\text{exp}}, g_j) * (1 + \eta(I_t, a_{\text{exp}})) > c(a_i, g_j) * (1 + \eta(I_t, a_i)) \text{ \& \& } a_{\text{exp}} \in \Omega_t\}$  are arranged in descending order according to the size of  $c(a_{\text{exp}}, g_j) * (1 + \eta(I_t, a_{\text{exp}}))$ , and the  $\mu(t, a_{\text{exp}}, g_j)$  of the arranged  $a_{\text{exp}}$  are filled in the list in turn.
2.  $\mu'(t, a_i, g_j) = Q_{\text{temp}}/r(t, g_j)$ 
3.  $x = 0$ 
4.  $C_{\text{pre}} = 0$ 
5. if list is empty
6.   do nothing
7. else if  $\mu'(t, a_i, g_j) \leq \text{list}(x)$ 
8.    $C_1 = Q_{\text{temp}}^* (c(a_{\text{exp}}, g_j) * (1 + \eta(I_t, a_{\text{exp}})) - c(a_i, g_j) * (1 + \eta(I_t, a_i))) / * a_{\text{exp}}$  makes  $\mu(t, a_{\text{exp}}, g_j) == \text{list}(x) / *$ 
9.   send request( $I_t, I_{t2}, a_i, g_j, Q_{\text{temp}}, C_1$ )
10. else
11.    $Q_{\text{temp}} = 0$ 
12.   while  $\mu'(t, a_i, g_j) > \text{list}(x)$ 
13.      $Q_{\text{temp}} + = \text{list}(x) * r(t, g_j)$ 
14.      $C_{\text{pre}} + = \text{list}(x) * r(t, g_j) * c(a_{\text{exp}}, g_j) * (1 + \eta(I_t, a_{\text{exp}})) / * a_{\text{exp}}$  makes  $\mu(t, a_{\text{exp}}, g_j) == \text{list}(x)$ 
15.      $\mu'(t, a_i, g_j) = \mu'(t, a_i, g_j) - \text{list}(x)$ 
16.      $x++$ 
17.   end while
18.  $Q_{\text{temp}} + = \mu'(t, a_i, g_j) * r(t, g_j)$ 
19.  $C_{\text{pre}} + = \mu'(t, a_i, g_j) * r(t, g_j) * c(a_{\text{exp}}, g_j) * (1 + \eta(I_t, a_{\text{exp}})) / * a_{\text{exp}}$  makes  $\mu(t, a_{\text{exp}}, g_j) == \text{list}(x) / *$ 
20. if  $Q_{\text{temp}} \geq q(a_i, g_j, t_2)$ 
21.    $C_1 = C_{\text{pre}} - Q_{\text{temp}}^* c(a_i, g_j) * (1 + \eta(I_t, a_i)) / * a_{\text{exp}}$  makes  $\mu(t, a_{\text{exp}}, g_j) == \text{list}(x) / *$ 
22.   send request( $I_t, I_{t2}, a_i, g_j, Q_{\text{temp}}, C_1$ )
23. else
24.   do nothing

```

high to low" by means of simulation. The replaced  $g_j$  order quantity  $Q_{\text{temp}}$  is compared with the order quantity  $q(a_i, g_j, t_2)$  of  $a_i$  at  $I_{t2}$ . If  $Q_{\text{temp}} \geq q(a_i, g_j, t_2)$ , calculate the cost  $C_1$  that can be saved by the new  $N_t$  after simulation compared with the original  $N_t$ , and issue a coordination request (step 1-22); otherwise, give up the coordination (step 23-24) (this is because the coordination will reduce the order quantity of  $a_i$ , which is not in line with the interests of  $a_i$  and the principle of non-cooperation).

#### 2) JUDGE STAGE

when  $I_{t2}$  receives the request  $(I_t, I_{t2}, a_i, g_j, Q_{\text{temp}}, C_1)$ , it judges whether and how to accept the coordination according to its own situation. The judgment basis includes:

- whether  $N_{t2}$  has started to supply;
- whether  $I_{t2}$  can find other nodes with  $g_j$  supply capacity to meet the order quantity  $q(a_i, g_j, t_2)$  to replace with;

- whether the cost loss  $C_2$  of  $I_{t2}$  caused by  $a_i$  replacement is less than the cost  $C_1$  saved by  $I_t$ .

**Algorithm 5** (Cooperate-Judge Algorithm)

```

1.  $C_{ai} = c(a_i, g_j) * q(a_i, g_j, t_2) * (1 + \eta(I_{t2}, a_i))$ 
2.  $C_{new} = 0$ 
3. if  $N_{t2}$  has started to supply
4.   refuse
5. else
6.    $\lambda_{temp}(t, g_j) = q(a_i, g_j, t_2)$ 
7.   for each  $a_k \in \Omega_{t2}$ 
8.     for each  $\{a_x | a_x \in N(a_k) \cup a_k \ \&\& \ \exists g_j \in G(a_x)\}$ 
9.       setOptPre( $a_k, a_x$ )
10.      PreDecide( $g_j, a_x, q(a_i, g_j, t_2)$ )
11.       $C_{new} + = c(a_x, g_j) * q(a_x, g_j, t_2) * (1 + \eta(I_{t2}, a_x))$ 
12.      if  $\lambda_{temp}(t, g_j) = 0$ 
13.        Terminate this Algorithm
14.      end for
15.      if  $\lambda_{temp}(t, g_j) = 0$ 
16.        break
17.    end for
18.    if  $\lambda_{temp}(t, g_j) > 0$ 
19.      refuse
20.    else
21.       $C_2 = C_{new} - C_{ai}$  /*  $C_2$  may be negative */
22.      if  $C_2 \geq C_1$ 
23.        refuse
24.      else
25.        agree( $I_t, I_{t2}, g_j, a_i, Contribution$ )

```

If  $N_{t2}$  has started to supply,  $I_{t2}$  will directly reject the coordination request of  $I_t$  (step 3-4), otherwise, it will try to find a node that can replace  $a_i$  to supply  $g_j$  (step 5-17). The PreDecide method is used to meet the  $g_j$  requirements of  $q(a_i, g_j, t_2)$  (step 10) (the method will be detailed below) until all requirements are met (step 15-16). If  $\lambda_{temp}(t, g_j) > 0$ , it indicates that the  $a_i$  order cannot be replaced successfully, the coordination request is rejected (step 18-19). Otherwise, the lost cost  $C_2$  when  $a_i$  is replaced is calculated (step 21). When  $C_2 \geq C_1$ , the coordination request is rejected. When  $C_2 < C_1$ , the coordination request is agreed and the contribution request between  $[C_2, C_1]$  is proposed (step 22-25).

The PreDecide method in step 10 is similar to the decide algorithm in “supply network construction algorithm”, in which the demander sends out an offer and the node who receives the offer makes a response. However, it has the following differences:

- It is called “Pre-Decide” because it seeks only alternative nodes and does not include the actual hiring process known as the confirm stage.
- Since contributions can be requested to  $I_t$  to cover losses, PreDecide are made regardless of cost and do not require threshold  $th$  limits.

- Since “locked capacity” and “forbidden capacity” cannot meet the coordination needs immediately, it is not necessary to distinguish between the two types of capacity in the response stage of PreDecide, only “free capacity” can be used for replacement.

3) CONFIRM STAGE

At this stage,  $I_t$  receives the judge result from  $I_{t2}$  for its coordination request, and makes different actions according to different results.

- If the result is agree( $I_t, I_{t2}, g_j, a_i, Contribution$ ):
  - $I_{t2}$  cancels all  $g_j$  orders of  $a_i$ , allocates the canceled orders to the nodes that make positive response in  $I_{t2}$ 's PreDecide algorithm, completes the change of supply network  $N_{t2}$ .
  - $I_t$  gets the right to employ  $a_i$ , replaces the high-cost  $g_j$  orders of  $N_t$  in algorithm 8, and hires  $a_i$  to provide  $g_j$  (the quantity of the supply is  $Q_{temp}$ ), and pays  $I_{t2}$  contribution, ConMoney, as the compensation for coordination.
- If  $I_t$  receives  $I_{t2}$ 's response as refuse: Abandon this coordination.

V. VERIFICATION AND CONCLUSION

The performance difference between the proposed algorithm and the traditional CN model is verified by a series of experiments with task cost as a measurement index.

A. DATA SET

In order to observe the performance of the algorithm under different conditions, specific parameter combination is used instead of the actual data set. Three groups of experiments were set up, and each group was executed 100 times. The parameter distribution is shown in Table 2.

TABLE 2. Parameter setup.

Parameter	Value1	Value2	Value3	Define
$th(t, g_j)$	$2 * c$	$3 * c$	$2 * c$	the threshold value to measure whether it is in its interest to employ a node to supply $g_j$ for task $t$
$c_s(a_i, g_j)$	$c$	$c$	$c$	the expectation of the cost of $a_i$ supply unit $g_j$
$r(t, g_j)$	$5 * Q$	$5 * Q$	$10 * Q$	the demand of task $t$ for $g_j$
$max_s(a_i, g_j)$	$Q$	$Q$	$Q$	The expectation of the maximum upper limit of $g_j$ supplied by $a_i$

Value1~3 simulated the normal scenario, high-threshold scenario and high-demand scenario respectively.

The parameters of each independent experiment were obtained from the normal distribution. The values in table are the expectations and the standard deviation is 1.

B. COMPARISON MODEL AND EVALUATION INDEX

- Algorithm in this paper
- Traditional CN method: in the process of supply network expansion,  $I_t$  only employs  $a_i$  to supply  $g_j$  under the

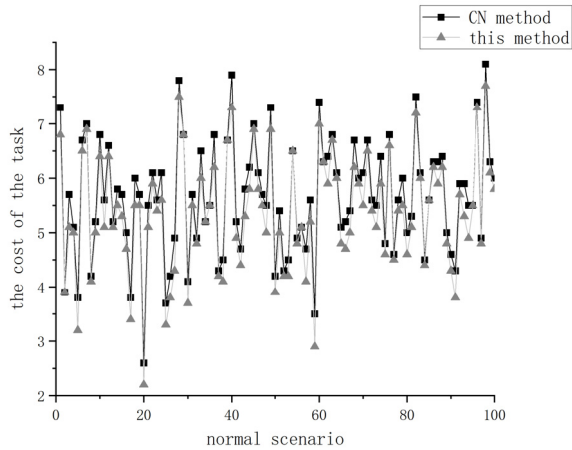


FIGURE 3. The experiment result of normal scenario.

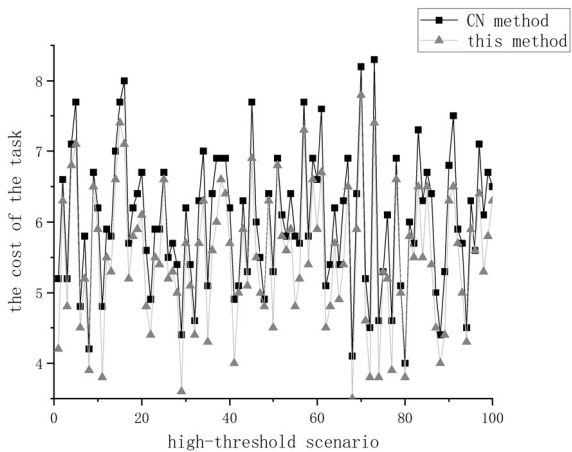


FIGURE 4. The experiment result of high-threshold scenario.

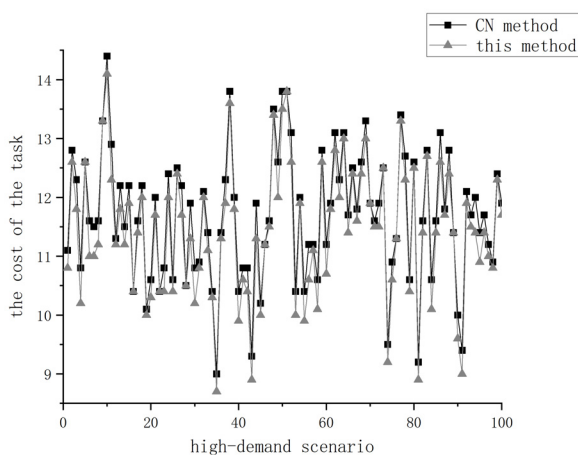


FIGURE 5. The experiment result of high-demand scenario.

condition that the  $g_j$  demand of  $N_t$  has not been met and the  $g_j$  cost of  $a_i$  is lower than the threshold  $th$ . Compared to this method, our model's advantage is the ability to continually update the supply network to select the

lowest cost suppliers and the opportunity to coordinate requesters to optimize the supply network.

### C. EVALUATION CRITERIA

The cost of task  $t$  is taken as the measurement index of the algorithm.

### D. EXPERIMENTAL RESULT

The experiment result is shown in Figure 3~5:

### E. CONCLUSION

Experimental results show that the proposed method has better performance than the traditional CN model. Although the performance differences between the two algorithms is not stable, the average cost of the three groups of experiments with different parameter settings was reduced by 5.60%, 8.26% and 2.57% respectively compared with the CN model. The following conclusions can be drawn:

- With the increase of threshold value, the compensation cost of the two models will increase at the same time, but the advantage of the algorithm in this paper is more significant.
- With the increase of task demand, the salary cost of the two models will expand at the same time, but the cost difference between the two models will remain unchanged in numerical value, which makes the performance advantage brought of the algorithm no longer obvious.
- The algorithm proposed in this paper performs better in the scenario of high measurement threshold and low demand for a single kind of product.

### REFERENCES

- [1] F. Buccafurri, G. Lax, S. Nicolazzo, A. Nocera, and D. Ursino, "Driving global team formation in social networks to obtain diversity," in *Web Engineering (Lecture Notes in Computer Science)*, vol. 8541, 2014, pp. 410–419.
- [2] J. Chamberlain, "Groupsourcing: Distributed problem solving using social networks," in *Proc. AAAI Conf. Hum. Comput. Crowdsourcing, Hcomp*, 2014, pp. 22–29, doi: [10.13140/2.1.3912.9607](https://doi.org/10.13140/2.1.3912.9607).
- [3] J. Jiang, B. An, Y. Jiang, C. Zhang, Z. Bu, and J. Cao, "Group-oriented task allocation for crowdsourcing in social networks," *IEEE Trans. Syst., Man, Cybern. Syst.*, early access, Aug. 28, 2019, doi: [10.1109/tsmc.2019.2933327](https://doi.org/10.1109/tsmc.2019.2933327).
- [4] M. Kargar, A. An, and M. Zihayat, "Efficient bi-objective team formation in social networks," in *Machine Learning and Knowledge Discovery in Databases (Lecture Notes in Computer Science)*, vol. 7524, 2012, pp. 483–498, doi: [10.1007/978-3-642-33486-3\\_31](https://doi.org/10.1007/978-3-642-33486-3_31).
- [5] Q. Liu, T. Luo, R. Tang, and S. Bressan, "An efficient and truthful pricing mechanism for team formation in crowdsourcing markets," in *Proc. IEEE Int. Conf. Commun.*, Jun. 2015, pp. 567–572, doi: [10.1109/ICC.2015.7248382](https://doi.org/10.1109/ICC.2015.7248382).
- [6] I. Lykourantzou, S. Wang, R. E. Kraut, and S. P. Dow, "Team dating: A self-organized team formation strategy for collaborative crowdsourcing ioanna," in *Proc. CHI Conf. Extended Abstr. Hum. Factors Comput. Syst.*, May 2016, pp. 1243–1249, doi: [10.1145/2851581.2892421](https://doi.org/10.1145/2851581.2892421).
- [7] M. Rokicki, S. Zerr, and S. Siersdorfer, "Groupsourcing: Team competition designs for crowdsourcing," in *Proc. 24th Int. Conf. World Wide Web WWW*, 2015, pp. 906–915, doi: [10.1145/2736277.2741097](https://doi.org/10.1145/2736277.2741097).
- [8] Y. Sun, W. Tan, and L. Huang, "A team discovery model for crowdsourcing tasks to social networks," *Int. J. Web Eng. Technol.*, vol. 12, no. 1, pp. 21–44, 2017, doi: [10.1504/IJWET.2017.084023](https://doi.org/10.1504/IJWET.2017.084023).



- [9] L. Tran-Thanh, S. Stein, A. Rogers, and N. R. Jennings, "Efficient crowdsourcing of unknown experts using multi-armed bandits," *Front. Artif. Intell. Appl.*, vol. 242, pp. 768–773, Feb. 2012, doi: [10.3233/978-1-61499-098-7-768](https://doi.org/10.3233/978-1-61499-098-7-768).
- [10] W. Wang, Z. He, P. Shi, W. Wu, and Y. Jiang, "Truthful team formation for crowdsourcing in social networks," in *Proc. Int. Jt. Conf. Auton. Agents Multiagent Syst. AAMAS*, 2016, pp. 1327–1328.
- [11] J. B. Huang, X. J. Sun, Y. Zhou, Z. Lü, H. L. Sun, and X. L. Jia, "Research Survey on Team Formation in Social Networks," *Ruan Jian Xue Bao/J. Softw.*, vol. 28, no. 4, pp. 993–1009, 2017, doi: [10.13328/j.cnki.jos.005152](https://doi.org/10.13328/j.cnki.jos.005152).
- [12] Y. Jiang and J. C. Jiang, "Understanding social networks from a multiagent perspective," *IEEE Trans. Parallel Distrib. Syst.*, vol. 25, no. 10, pp. 2743–2759, Oct. 2014, doi: [10.1109/TPDS.2013.254](https://doi.org/10.1109/TPDS.2013.254).
- [13] M. Kokkodiis, P. Papadimitriou, and P. G. Ipeirotis, "Hiring behavior models for online labor markets," in *Proc. 8th ACM Int. Conf. Web Search Data Mining WSDM*, 2015, pp. 223–232, doi: [10.1145/2684822.2685299](https://doi.org/10.1145/2684822.2685299).
- [14] G. Miao, S. Tao, W. Cheng, R. Mouluc, L. E. Moser, D. Lo, and X. Yan, "Understanding task-driven information flow in collaborative networks," in *Proc. 21st Int. Conf. World Wide Web WWW*, 2012, pp. 849–858, doi: [10.1145/2187836.2187951](https://doi.org/10.1145/2187836.2187951).
- [15] P. De Meo, E. Ferrara, D. Rosaci, and G. M. L. Sarne, "Trust and compactness in social network groups," *IEEE Trans. Cybern.*, vol. 45, no. 2, pp. 205–216, Feb. 2015, doi: [10.1109/TCYB.2014.2323892](https://doi.org/10.1109/TCYB.2014.2323892).
- [16] T. Lappas, K. Liu, and E. Terzi, "Finding a team of experts in social networks," in *Proc. 15th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining KDD*, 2009, pp. 467–475, doi: [10.1145/1557019.1557074](https://doi.org/10.1145/1557019.1557074).
- [17] A. Majumder, S. Datta, and K. V. M. Naidu, "Capacitated team formation problem on social networks," in *Proc. 18th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining KDD*, 2012, pp. 1005–1013, doi: [10.1145/2339530.2339690](https://doi.org/10.1145/2339530.2339690).
- [18] T. Voice, M. Polukarov, and N. R. Jennings, "Coalition structure generation over graphs," *J. Artif. Intell. Res.*, vol. 45, pp. 165–196, Oct. 2012, doi: [10.1613/jair.3715](https://doi.org/10.1613/jair.3715).
- [19] S. Liemhetcharut and M. Veloso, "Weighted synergy graphs for effective team formation with heterogeneous ad hoc agents," *Artif. Intell.*, vol. 208, pp. 41–65, Mar. 2014, doi: [10.1016/j.artint.2013.12.002](https://doi.org/10.1016/j.artint.2013.12.002).
- [20] L. Sabattini, C. Secchi, and N. Chopra, "Decentralized estimation and control for preserving the strong connectivity of directed graphs," *IEEE Trans. Cybern.*, vol. 45, no. 10, pp. 2273–2286, Oct. 2015, doi: [10.1109/TCYB.2014.2369572](https://doi.org/10.1109/TCYB.2014.2369572).
- [21] W. Wang, J. Jiang, B. An, Y. Jiang, and B. Chen, "Towards efficient team formation for crowdsourcing in non-cooperative social networks," *IEEE Trans. Cybern.*, vol. 47, no. 12, pp. 4208–4222, Dec. 2017, doi: [10.1109/TCYB.2016.2602498](https://doi.org/10.1109/TCYB.2016.2602498).
- [22] S. L. Lim and C. Ncube, "Social networks and crowdsourcing for stakeholder analysis in system of systems projects," in *Proc. 8th Int. Conf. Syst. Syst. Eng.*, Jun. 2013, pp. 13–18, doi: [10.1109/SYSoSE.2013.6575235](https://doi.org/10.1109/SYSoSE.2013.6575235).
- [23] H. Jiang and S. Matusbara, "Efficient task decomposition in crowdsourcing," in *PRIMA, Principles and Practice of Multi-Agent Systems* (Lecture Notes in Computer Science), vol. 8861, 2014, pp. 65–73, doi: [10.1007/978-3-319-13191-7\\_6](https://doi.org/10.1007/978-3-319-13191-7_6).
- [24] L. Tran-Thanh, T. D. Huynh, A. Rosenfeld, S. Ramchun, and N. R. Jennings, "BudgetFlx: Budget limited crowdsourcing for interdependent task allocation with quality guarantees," in *Proc. 13th Int. Conf. Auton. Agents Multiagent Syst. AAMAS*, vol. 1, Apr. 2014, pp. 477–484.
- [25] T. Wolf, A. Schröter, D. Damian, L. D. Panjer, and T. H. D. Nguyen, "Mining task-based social networks to explore collaboration in software teams," *IEEE Softw.*, vol. 26, no. 1, pp. 58–66, Jan. 2009.



**XU-DONG LU** received the master's degree from the Department of Computer Science and Technology, Shandong University, in 2001, and the Ph.D. degree in engineering from Shandong University, in 2010. He is currently a Lecturer with the School of Software, Shandong University. He has participated in more than ten national, provincial, and ministerial level projects. He has published more than ten articles in important journals or conferences. He holds more than ten patents for invention in application field. His main research interests include big data technology and intelligent data analysis. He received one First Class Prize for the Ministry of Education Teaching Achievement Award and one Second Class Prize for the Teaching Achievement Award of Shandong Province.



**SHI-PENG WANG** was born in 1995. He received the bachelor's degree from the School of Computer Science and Technology, Shandong University, in 2017, where he is currently pursuing the Ph.D. degree. His research interests include reinforcement learning, multiagent systems, and mental model.



**WEI GUO** was born in 1978. He received the master's degree from the Department of Computer Science and Technology, Shandong University, in 2005, and the Ph.D. degree in engineering from Shandong University, in 2015. He is currently an Engineer and a master's supervisor. He has participated in more than ten national, provincial, and ministerial level projects. He has published more than 20 articles in important journals or conferences. He has published an academic monograph. He holds more than 10 patents for invention in application field, of which seven are first invention applicants. His current research interests include big data technology and intelligent data analysis. He is with the Committee of CCF TCSC and CCF TCCC. He was a recipient of several prizes for the Scientific and Technology Progress (STP), including one Second Class Prize for the STP of State Education Ministry of China, one First Class Prize, and two Second Class Prizes for the STP of Shandong Province. He was also a recipient of one First Class Prize for the Ministry of Education Teaching Achievement Award and one Second Class Prize for the Teaching Achievement Award of Shandong Province.



**SHAO-JIE ZHANG** was born in 1998. He received the bachelor's degree from the School of Software, Shandong University, in 2020. He is currently pursuing the master's degree in artificial intelligence with The University of Manchester. He has participated in four national, provincial, and ministerial level projects. His research interests include multiagent systems, machine learning, and reinforcement learning. He is a member of CCF.