IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received February 10, 2018, accepted March 11, 2018, date of publication March 27, 2018, date of current version April 25, 2018.

Digital Object Identifier 10.1109/ACCESS.2018.2819992

Social Connection Aware Team Formation for Participatory Tasks

XIAOYAN YIN[1], (Member, IEEE), CHAO QU[1], QIANQIAN WANG[1], FAN WU[2], BAOYING LIU[1], FENG CHEN[1], XIAOJIANG CHEN[1], AND DINGYI FANG[1]
[1]School of Information Science and Technology, Northwest University, Xi'an 710127, China
[2]Department of Computer Science and Engineering, Shanghai Jiao Tong University, Shanghai 200240, China

Corresponding author: Baoying Liu (paola.liu@nwu.edu.cn)

This work was supported in part by the National Key Research and Development Program of China under Grant 2017YFB1400301 and in part by the International Cooperation Foundation of Shaanxi, China, under Grant 2013KW01-02.

ABSTRACT Performance of a collaborative task is mostly dependent on the collective effort from participants. To accomplish a participatory task effectively and efficiently, the team formation problem (TFP) outweighs all other considerations. It is even more complicated when social connections among candidates is taken into account. As we can imagine, a large number of tasks require members of the team to be socially close. On the contrary, a portion of tasks, e.g., proposal review, pay more attention to a multidimensional view, and team members should be selected from a variety of cliques. Due to the nature of tasks, it is challenging to find a subset that meets the skill requirement of the task as well as socially diversity demand of team members from a pool of candidates. In this paper, we explore the TFP in a social network. Based on different task objectives, we first formulate the TFP as TFP with strong ties (TFP-ST) and TFP with weak ties (TFP-WT), respectively. Both TFP-ST and TFP-WT are proven to be NP-hard, and we then design corresponding heuristic algorithms to solve the two problems. Through extensive simulations, we show that the solution to TFP-ST can achieve significant improvement in terms of collaboration cost, team size, as well as running time, and the solution to TFP-WT can provide better performance than existing approaches at the same time.

INDEX TERMS Team Formation, social network, strong tie, weak tie.

I. INTRODUCTION

Participatory tasks, e.g., software product development, message propagation [1], data offloading [2], etc, combine the collective intelligence of the massive crowd. However, whether a task is accomplished successfully depends not only on the proficiency of participants, but also on how all of participants communicate, cooperate and work together as a team. Therefore, selection of participants is vital for participatory tasks. Diversity of candidates' skills and the social relationship among them make participant selection (team formation, interchangeably) a challenging problem.

The team formation problem [3] is one of the most essential problems for participatory tasks. The duty of group formation is to find a team of participants, that can fulfill the requirements of one certain task, from a pool of candidates. Every participant then makes her contribution to the specific task. To meet different design objectives, a number of solutions [4]–[7] to the team formation problem have been proposed. However, these existing works did not take levels of social connection among team members into consideration. As we can imagine, a large number of tasks require members of the team to be socially close. On the contrary, a portion of tasks, e.g., proposal review, pay more attention to a multidimensional view, team members should be selected from a variety of cliques. As shown in Fig.1, candidate A and B, who are connected by the solid line, are close socially, we call this type of connection as 'strong tie'. On the other hand, due to 'weak tie' between candidate A and H who are from different cliques, the communication cost will be higher and diversity will be better at the same time. Thus, it is crucial how to find a subset that meets the skill requirement of the task as well as socially diversity demand of team members from a social network.

Based on different task objectives, we first formulate the team formation problem as TFP with strong ties (TFP-ST) and TFP with weak ties (TFP-WT), respectively. The TFP-ST problem aims to minimize the collaboration cost among team members. Lappas *et al.* [3] studied the method of forming a team with strong ties, and defined two metrics about collaboration cost to evaluate performance of the result

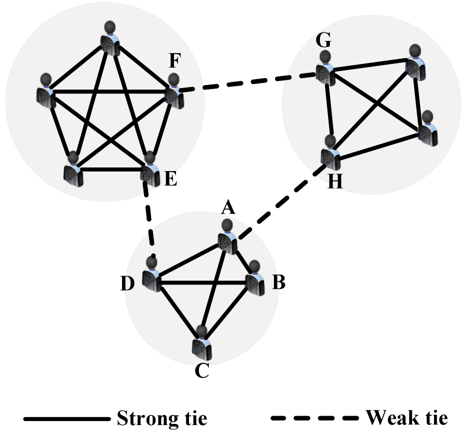


FIGURE 1. The logic structure of a social network.

of team formation. Since then, a series of studies [8]–[10] have been conducted, a few metrics, such as distance collaboration function, diameter cost etc., were discussed, and load balancing was also considered as a factor in reference [8]. Unlike previous works, in this paper, we introduce a new metric, i.e., success ratio of tasks, to evaluate the execution result of tasks by a chosen team, a successful task means the result achieved by the proposed algorithm satisfies a certain quality requirement, below which this task cannot be accomplished successfully. Obviously, every team member has different contribution to a specific task because of the heterogeneity in her skill level. For example, for a proofreading task which requires that the error ratio is less than 1%, a candidate who can guarantee correct ratio less than 99% is unqualified, this task fails if the candidate is chosen exclusively. On the other hand, the TFP-WT problem aims to form a diverse team. There exist only few works on this field. Yin *et al.* [11] defined the social influence metric for a team, and studied the problem about finding a team with minimized social influence. However, parameters, e.g., authority, used in this work are difficult to attain in real life, as analyzed in [12].

Our original contributions in this paper are two-fold. First, we propose TFP-ST and TFP-WT to model the team formation problem for participatory tasks, and provide two corresponding heuristic algorithms to solve TFP-ST and TFP-WT, respectively. Second, we introduce two metrics, i.e., success ratio and collaboration cost, to evaluate our algorithms. We have conducted extensive simulations to assessment the performance of our proposed algorithms. Through extensive simulations, we show that the solution to TFP-ST can achieve significant improvement in terms of collaboration cost, team size as well as running time, and the solution to TFP-WT can provide better performance than existing approaches at the same time.

The remainder of this paper is organized as follows. We survey related works in Section 2. In Section 3, we present the preliminaries, including basic models and metrics, in detail. We define TFP-ST and TFP-WT to model the team formation problem in Section 4. Discussions about the problems and corresponding algorithms are presented in section 5. Simulation results are presented in Section 6. Finally, we conclude this paper in Section 7.

II. RELATED WORK

Traditional team formation problem has been studied widely [4]–[7], [14]. The team formation problem is modelled as an integer linear program problem, and often solved based on simulated annealing [4], Heuristic [5] or genetic algorithms [6]. However, these studies didn't take the underlying social network structure into consideration, which is the main different between previous research and our work. In this paper, we leverage different objectives of tasks, and then present corresponding social connection aware algorithms for TFP-ST and TFP-WT, respectively. The team formation problem has become more complicated when taking social connection, especially the social network structure, into consideration. Social connection related team formation has attracted lots of researchers' attention. Research about forming a team with strong ties is first conducted in [3], in which the collaboration cost of a team is used to evaluate levels of social connection among members. Lappas *et al.* introduced two metrics, i.e., diameter communication cost as well as Minimum Spanning Tree (MST) communication cost. Kargar and An [9] proposed a solution to the problem of finding a team of experts with a leader. They defined the sum of distances collaboration function, and used a metric, i.e., leader distance, to evaluate the proposed algorithm. Majumder *et al.* [10] introduced the bottleneck edge cost, and proposed a novel solution to the team formation problem. In [8] and [14], the density of the subgraph was used to identify a highly collaborative team. Based on both closeness centrality and eigenvector centrality, Ashenagar *et al.* [13] presented an effective solution to minimize the communication cost of a team for the team formation problem. In addition to aforementioned research, a few new metrics which can work as evaluation criteria have been introduced, such as collaboration cost and load balancing [8], [15], personnel cost [16], [17], time limit [18], team formation efficiency [19] and other factors [20], lots of solutions related to these metrics have been proposed to solve the team formation problem. These existing works assumed that a task is accomplished successfully as long as one candidate serves it. However, candidates have heterogeneous skill levels because of their diversity in proficiency, capabilities and experience. Therefore, to find a team with high efficiency, both diversity of candidates and success ratio of tasks should be taken into consideration.

The team formation problem with weak ties is still an open research topic. Yin *et al.* [11] introduced a metric to evaluate the social influence of a team, and then proposed a solution that can minimize the social influence for the team formation problem. However, the metric, e.g., authority, described in [11] is hard to measure. In this paper, we evaluate levels of the social influence by using the collaboration cost of a team, which is widely used in this area. In view of team members from different cliques, the high communication cost can result in the impartiality of the execution result of a task, which is exactly the goal of some certain tasks. In this paper, we use two metrics, i.e., success ratio and collaboration cost, and formulate the team formation problem as TFP-ST and TFP-WT according to different requirements of tasks, and then propose two corresponding heuristic algorithm to solve TFP-ST and TFP-WT, respectively.

III. PRELIMINARIES

In this section, we explain some notations, and define our social connection aware team formation problem with strong ties and weak ties, respectively.

A. SYSTEM MODEL

We consider a social network as an undirected graph $G = (V, E, w)$, where V denotes a set of candidates, edges in E denote social connection between any two candidates, and w is the weight of edge E, where $0 < w \leq 1$. The value of w stands for the collaboration cost between two neighbouring candidates in a social network. A smaller w implies that these two candidates are socially close, a bigger w means that these two candidates can not cooperate well and a resulting higher communication cost.

The collaboration cost among candidates can be estimated by means of several methods, e.g., analyzing historic data or collecting information from the hierarchy structure in an institution. For candidate u and v which are not socially close, the collaboration cost between u and v is defined as sum of the weights of edges which constitute the shortest path between u and v. The distance function satisfies the triangle inequality, i.e., $d(u, v) < d(u, x) + d(x, v)$, where x stands for any candidate. The set of candidates along the shortest path from u to v are denoted as $Path(u, v)$. The distance between candidate u and a set of candidates X is defined as $d(u, X) = \min_{x \in X} d(u, x)$, and the nearest node to u is denoted by $N(u, X) = \{x^* | d(u, x) \geq d(u, x^*), \forall x \in X\}$. Thus, we have $d(u, X) = d(u, N(u, X))$ and $Path(u, X) = Path(u, N(u, X))$.

We assume project P is composed of several tasks, and can be denoted by $P = \{t^1, t^2, \ldots, t^p\}$, $p \geq 1$. We assume that these tasks are independent from each other. Naturally, the result of project P can be deduced from the combination of the results of p tasks. Every task has its own requirement in skill. A task can only be successfully accomplished when its quality demand is satisfied. In other words, task t^p is successful only if a candidate, whose skill level is larger than the quality requirement by task t^p, is chosen to serve it. However, such a qualified candidate does not always exist. Thus, we introduce a metric, success ratio, to represent the probability whether t^p, $p \geq 1$, to be executed successfully. The success ratio is computed as the ratio of the skill level of the chosen candidate to the quality requirement of task t^p. Let S^p denote the quality requirement of task t^p, $0 < S^p \leq 1$. The quality requirement for project P can be described as $\{\langle t^1, S^1\rangle \langle t^2, S^2\rangle \ldots, \langle t^p, S^p\rangle\}$. From another point of view, the success ratio constraint also gives us a chance to differentiate the tasks based on their importance.

Team members can participate in different tasks, and provide a corresponding success ratio for a certain task. Let $M_i = \{t^1, t^2, \ldots, t^m\}$ denote the set of tasks in which member i can participate, q_i^k denote the quality provided by member i for task t^k, which can be estimated using the data mining technology proposed by [21]. Members who can serve the same task t^k form a set for task t^k, denoted by $C(t^k)$.

A team member can provide her contribution to any task. This means that the success ratio of task t^k can be defined as one minus the probability that everyone in the team fails to serve it. Accordingly, the aggregate quality for task t^k should be described as $q^k = 1 - \prod_{i \in V'} (1 - q_i^k)$. We think the team formation problem is solved when team members, which are selected to serve task t^k, $1 \geq k \geq p$, can provide the quality guarantee, i.e., $q^k \geq S^k$, $1 \geq k \geq p$. Accordingly, project P is accomplished successfully only if all tasks are executed successfully, i.e., $q^k \geq S^k, \forall t^k \in P$.

We summarize major parameters and their definitions in Table 1.

TABLE 1. Key Parameters.

Parameter	Definition
$G = (V, E, w)$	the social network, an undirected and weighted graph
$i \in V$	a candidate
$e(u, v) \in E$	the edge connecting candidate u and v
$d(u, v)$	the distance between u and v
$Path(u, v)$	the set of candidates along the shortest path from u to v
$d(u, X)$	the distance between candidate u and set X
$Path(u, X)$	the set of candidates along the shortest path from u to set X
P	project $P = \{t^1, t^2, \cdots, t^p\}$
M_i	the set of tasks that candidate i can participate in, $M_i = \{t^1, t^2, \cdots, t^m\}$
S^k	the quality requirement of task t^k
q_i^k	the quality provided by candidate i for task t^k
$C(t^k)$	the set of candidates who can serve task t^k
$G' = (V', E')$	the team which is composed of chosen candidates
$\|X\|$	the cardinality of set X
$A \backslash B$	the difference set between A and B

B. METRICS

How all of team members cooperate is critical for a project. Obviously, keeping socially close makes collaboration more efficient. However, it is hard to evaluate the level of the social connection among members. As described in aforementioned research, the collaboration cost of a team is adopted to evaluate the level of social connections among members. We use similar definition of collaboration cost to estimate the level of strong ties in this paper. Intuitively, a team can collaborative well to pursue bigger benefit if they have lower collaboration cost. We borrow the definition of collaboration cost of a team from [3].

Definition 1: The collaboration cost of team V', which is composed of chosen candidates from a social network $G = (V, E, w)$, is the cost of the minimum spanning tree (MST) of $G(V', E', w)$, denoted as $C\_MST(V')$.

To evaluate the rating of weak ties, we introduce a metric, i.e., the minimum collaboration cost. The minimum collaboration cost represents the smallest cooperation cost between any two members in a team. Thus, the objective of team formation problem with weak ties is to maximize the minimum collaboration cost of a team.

Definition 2: The minimum collaboration cost of team V', which is a subset of candidates from the social network $G = (V, E, w)$, is denoted as $C\_Min(V')$, where $C\_Min(V') = \min_{u,v \in V'} d(u, v)$, representing the smallest collaboration cost among team members.

IV. PROBLEM FORMULATION

In this section, we formulate the team formation problem with strong ties as TFP-ST based on the Group Steiner Tree problem, and formulate the team formation problem with weak ties as TFP-WT based on the Multiple-choice Cover problem. Evidence suggests that the communication cost among a team is a key factor for forming of a team.

A. TEAM FORMATION PROBLEM WITH STRONG TIES

Problem 1: Given a social network, $G = (V, E, w)$, project P, $\{\langle t^1, S^1\rangle \langle t^2, S^2\rangle \ldots, \langle t^p, S^p\rangle\}$, and the profile of every candidate i, $M_i = \{t^1, t^2, \ldots, t^m\}$, $\{\langle t^1, q_i^1\rangle \langle t^2, q_i^2\rangle \ldots, \langle t^k, q_i^k\rangle\}$, $1 \le i \le n, 1 \le k \le p$, team formation problem with strong ties is to find a group of candidates, i.e., a subset $V' \subseteq V$, which can meet the quality requirement of all tasks and minimize the total collaboration cost at the same time. We call this problem TFP-ST. Mathematically speaking, TFP-ST can be formulated as follows:

$$Minimize\ C\_MST(V') \tag{1}$$

$$Subject\ to:\ V' \subseteq V \tag{2}$$

$$E' \subseteq E \tag{3}$$

$$q^k = 1 - \prod_{i \in V'} (1 - q_i^k) \tag{4}$$

$$q^k \ge S^k, \quad \forall t^k \in P \tag{5}$$

Proposition 1: The TFP-ST problem is NP-hard.

Proof: We prove the proposition by a reduction from the Group Steiner Tree (GST) problem [22], which has been proven to be NP-hard.

An instance of the GST problem is defined as follows: Given an undirected and weighted graph of candidates $G = (V, E, w)$, w is the weight of edge E, and p is the number of subsets (called group) $\{g^1, g^2, \ldots, g^p\}$ with $g^i \subseteq V$, $1 \le i \le p$. The GST problem is to find whether there exists a subtree $G' = (V', E', w)$ of $G = (V, E, w)$ (i.e., $V' \subseteq V$ and $E' \subseteq E$) such that $|V' \bigcap g^i| > 0$, $1 \le i \le p$, and the collaboration cost $\sum_{e \in E'} w(e) \le W$, where W is a predefined constant.

We transform an instance of the GST problem to an instance of the TFP-ST problem as follows: for group g^i, $1 \le i \le p$, we create set $C(t^k)$ for task t^k, $1 \le k \le p$, accordingly. We assume that member i in $C(t^k)$ can serve task t^k with the quality guarantee, such that every task t^k only need to select only one candidate from $C(t^k)$, $1 \le k \le p$, to optimize the communication cost. Thus, the graph G' of the TFP-ST problem is identical to the graph G of the GST problem, where the communication cost function corresponds the weight of the edge in the TFP-ST instance of the problem. Given this mapping, it is easy to indicate that there exists a solution to the GST problem with collaboration cost W at most, if and only if there exists a solution to the TFP-ST problem with communication cost W at most. This problem is trivially in NP.

The GST instance is a special case of the TFP-ST problem. Normally, $\frac{q_i^k}{S^k} \ge 1$, $1 \le i \le n$, $1 \le k \le p$, does not always hold for the TFP-ST problem, our problem should be more complicated than the special case. Therefore, the TFP-ST problem is NP-hard.

B. TEAM FORMATION PROBLEM WITH WEAK TIES

Problem 2: Given a social network, $G = (V, E, w)$, project $P = \{t^1, t^2, \ldots, t^p\}$, $p \ge 1$, and the profile of candidate i, $M_i = \{t^1, t^2, \ldots, t^m\}$, $\{\langle t^1, q_i^1\rangle \langle t^2, q_i^2\rangle \ldots, \langle t^k, q_i^k\rangle\}$, $1 \le i \le n, 1 \le k \le p$, we assume that every candidate in $C(t^k)$ is qualified to serve task t^k. Team formation problem with weak ties is to find a group of candidates, i.e., a subset V', $V' \subseteq V$, to serve every task, $|V' \bigcap C(t^k)| > 0, \forall t^k \in P$, and maximize the communication cost at the same time. We call this problem TFP-WT. Mathematically speaking, TFP-WT can be formulated as follows:

$$Maximize\ C\_Min(V') \tag{6}$$

$$\text{Subject to}:\ V' \subseteq V \tag{7}$$

$$|V' \bigcap C(t^k)| > 0, \quad \forall t^k \in P \tag{8}$$

$$C\_Min(V') = \min_{u,v \in V'} d(u, v) \tag{9}$$

Proposition 2: The TFP-WT problem is NP-hard.

PROOF: We prove the TFP-WT problem is NP-hard by a reduction from the MCC-MD (Multiple-choice cover of maximum dispersion) problem, which was proven to be NP-hard [23].

An instance of the MCC-MD problem is defined as follows: Given a set of vertices, $V = \{1, \ldots, n\}$, matrix $D = (d_{ij}) \in \Re_+^{n \times n}$ representing the distance between any two vertices in V, and a group of subsets, $\{H^1, \ldots, H^p\}$, where $H^k \subseteq V$, $1 \le k \le p$. The MCC-MD problem is to choose a set of vertices V', $V' \subseteq V$, such that $|V' \bigcap H^k| > 0$, $1 \leq k \leq p$, and maximize the distance between the two nearest nodes in V'. Given a constant Y, the solution to the MCC-MD problem is to find whether there exists a set $V' \subseteq V$ satisfying that $|V' \bigcap H^k| > 0$, $1 \leq k \leq p$, and the distance between the two nearest nodes in V' satisfies $\hat{d} = \max\{\min d_{kl} | k, l \in C\} \geq Y$.

We transform an instance of the MCC-MD problem to an instance of the TFP-WT problem as follows: we regard H^k, $1 \leq k \leq p$ as a set of candidates for task t^k, $1 \leq k \leq p$, i.e., $C(t^k)$, $1 \leq k \leq p$, and the element of matrix $D = (d_{ij}) \in \Re_+^{n \times n}$ as the weight of any two candidates. The project P consists of t^k, $1 \leq k \leq p$, the set of chosen members needs to satisfy $|V' \bigcap C(t^k)| > 0$, $1 \leq k \leq p$. It's easy to say that there exists a solution to the MCC-MD problem with cost at least Y, if and only if there exists a solution of the TFP-WT problem with minimum collaboration cost at least Y. Thus, the TFP-WT problem is NP-hard.

V. PROPOSED ALGORITHMS

Since there does not exist polynomial approximation solutions to both the TFP-ST problem and the TFP-WT problem with a bounded error guarantee, we propose two heuristic algorithms to solve the two problems in this section, respectively.

A. THE PROPOSED ALGORITHM FOR THE TFP-ST PROBLEM

The TFP-ST problem needs to find a team which meets the requirement of success ratio for every task. Since the TFP-ST problem is NP-hard, there does not exist a polynomial approximation solution with a bounded error guarantee. Thus, based on the solution of Group Steiner Tree problem [22], we present a heuristic algorithm for the TFP-ST problem.

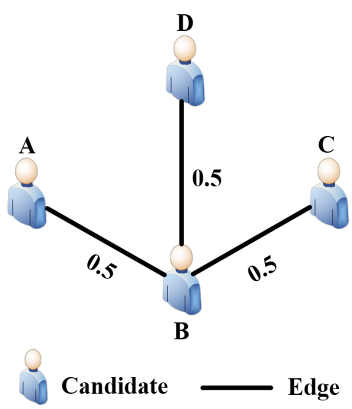


FIGURE 2. An example of a social network.

The MPH algorithm [24], [25] is a classical algorithm to the Steiner tree problem. We propose our solution to the TFP-ST problem based on the MPH algorithm. Given a social network, $G = (V, E, w)$, as shown in Fig.2, A, B, C and D are four candidates, edges in E reflect the social connection among candidates, and the weights of edges imply the communication cost between two neighbouring candidates. We assume that the set of tasks T, $T = \{t^1, t^2, t^3, t^4\}$, $C(t^1) = \{A, D\}$, $C(t^2) = \{A\}$, $C(t^3) = \{B, D\}$, $C(t^4) = \{C, D\}$.

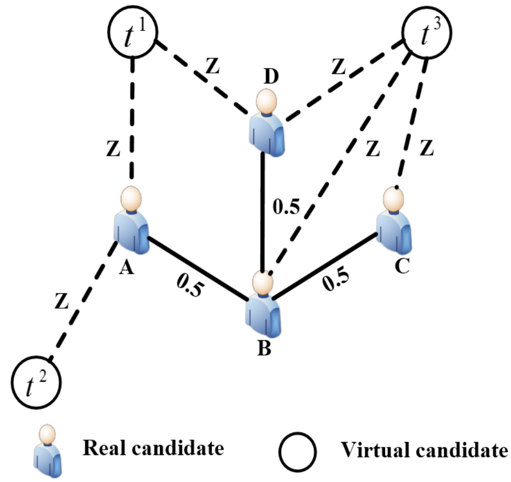


FIGURE 3. Illustration of our proposed algorithm for the TFP-ST problem.

For project P, $P = \{t^1, t^2, t^3\}$, our algorithm for the TFP-ST problem is described as follows: (1) Addition of virtual candidates and virtual edges. First, add task t^1, t^2, t^3 as virtue vertices into graph G, the set of essential nodes is denoted by $R = \{t^1, t^2, t^3\}$. Second, add $|C(t^k)|$ virtual edges for task t^k, where a new virtual edge is produced between task t^k and every candidate in $C(t^k)$, as well as set the weight of all virtual edges to a large value Z. The generated graph is shown in Fig.3. (2) Process of our proposed algorithm for the TFP-ST problem. Add a task with minimum cardinality into set V', i.e., t^2. Then find the nearest candidate in $R \setminus V' = \{t^1, t^3\}$ to the set of chosen candidates V', i.e., t^1, and add t^1 as well as all candidates along the shortest path to set V' into V', where $Path(t^1, V') = Path(t^1, t^2) = \{t^1, A, t^2\}$. Then, we have $V' = \{t^1, A, t^2\}$. Similarly, t^3 is chosen, add t^3 and all candidates along the shortest path to set V' into V', where $Path(t^3, V') = Path(t^3, A) = \{t^3, B, A\}$. Thus, we have $V' = \{t^1, A, t^2, t^3, B, \}$, $R \setminus V' = \emptyset$. (3) Optimization of the chosen team. Delete virtual candidates and virtual edges, we have the set of qualified candidates as a team, i.e., $V' = \{A, B\}$, and the minimized collaboration cost, $C\_MST(V') = 0.5$. The proposed algorithm for the TFP-ST problem is shown in Algorithm 1.

In Algorithm 1, from line 3 to line 8, we add virtual vertices and corresponding edges into graph G, and fix a large number Z as the weight of these virtual edges, where Z is bigger than the sum of weights of edges in graph G. In line 9, we choose the task with the lowest cardinality, i.e., $t^{rarest} = \arg\min_{t^k \in P} |C(t^k)|$, and add it into V'. Let R denote the set of tasks whose requirements are not met. In line 12, the virtual node, which is the nearest

Algorithm 1 RarestSteiner

Input: $G = (V, E, w)$, $P = \{t^1, t^2, \ldots, t^p\}$, M_i, $C(t^k)$, $\{\langle t^k, q_i^k \rangle\}$, $\{\langle t^k, S^k \rangle\}$, $1 \leq i \leq n$, $1 \leq k \leq p$.
Output: Team V'.

1: $R = P$.
2: $V' = \emptyset$.
3: **for** task t^k in P **do**
4:     G.add-node(t^k).
5:     **for** candidate i in $C(t^k)$ **do**
6:         G.add-edge(t^k,i,weight=Z).
7:     **end for**
8: **end for**
9: $t^{rarest} = \arg\min_{t^k \in P} |C(t^k)|$.
10: $V' = \{t^{rarest}\}$.
11: **while** $R \neq \emptyset$ **do**
12:     $v^* = \arg\min_{u \in U \backslash V'} d(u, V')$.
13:     **if** $Path(v^*, V') \neq \emptyset$ **then**
14:         $V' = V' \cup Path(v^*, V')$.
15:     **else**
16:         return failure.
17:     **end if**
18:     $V' = V' \backslash P$.
19:     $R = \{t^k | q^k = 1 - \prod_{i \in V'}(1 - q_i^k) < S^k, t^k \in P\}$.
20:     E^v = virtual edges in $Path(v^*, V')$.
21:     $E = E \backslash E^v$.
22:     **if** $V \backslash V' == \emptyset$ and $R \neq \emptyset$ **then**
23:         return failure.
24:     **end if**
25: **end while**
26: Clean-up(V').
27: return V'.

candidate to V', $v^* = \arg\min_{u \in U \backslash V'} d(u, V')$, is chosen. In line 13-14, candidates, $Path(v^*, V')$, along the shortest path are added into V'. Finally, virtual vertices and virtual edges should be deleted. There might exist some redundant candidates in V'. These redundant candidates should be deleted to improve the performance of the team. This procedure is shown in Algorithm 2. Since we add nodes into V' step by step, it's easy to record the leaf nodes. For every leaf node, we can test whether it is redundant, and then delete the redundant nodes.

B. DISCUSSIONS FOR THE TFP-ST PROBLEM

(1) Weight of virtual edges. Virtual vertex t^k will generate $|C(t^k)|$ virtual edges, $1 \leq k \leq p$. The weight of a virtual edge should be larger than the sum of weights of all edges in graph G. The main reason for this is to prevent these virtual vertices from being selected by the shortest path. As shown in Fig.3, during the process of finding the shortest path between t^3 and $\{t^1, A, t^2\}$, $d(t^3, t^1) = 2 * Z$, $d(t^3, A) = Z + 0.5$, $d(t^3, t^2) = 2*Z+0.5$. With a smaller Z, $2*Z < Z+0.5$ might hold. Thus, we have $Path(t^3, V') = Path(t^3, t^1) = \{t^3, D, t^1\}$. Finally, $V' = \{t^1, t^2, t^3, A, D\}$. $V' = \{A, D\}$ after all virtual vertices are deleted. Unfortunately, project P fails because there is no strong tie between A and D.

Algorithm 2 Clean-Up

Input: $T(V', E', w)$, M_i, $\{\langle t^k, S^k \rangle\}$, $\{\langle t^k, q_i^k \rangle\}$, $1 \leq i \leq n$, $1 \leq k \leq p$.
Output: The team V'.

$L \leftarrow$leaf nodes in $T(V', E')$; $M \leftarrow \emptyset$.
while $L \backslash M$ not empty **do**
    if node-i in $L \backslash M$ is redundant **then**
        tem-node=node-i.neighbor.
        remove node-i in V'.
        if tem-node is a leaf node **then**
            $L = L \cup \{$tem-node$\}$.
        end if
    else
        $M = M \cup \{node_i\}$.
    end if
end while

(2) Success ratio constraint. In this paper, we introduce the success ratio constraint. A candidate provides different success ratios for different tasks. As shown in Fig.3, during the process of adding candidates into set V', after the first step, $V' = \{t^1\}$ becomes $V' = \{t^1, A, t^2\}$, and so on, the final team will be $V' = \{A, B\}$. If $q_A^1 < S^1$, project P fails. To deal with this, let R denote the set of tasks whose quality requirements have not been met. In the beginning, $R = \{t^1, t^2, t^3\}$. When $V' = \{t^1, A, t^2\}$, we need to check whether the success ratio constraint is satisfied. Since $q_A^1 < S^1$, $R = \{t^1\}$, such that other candidates in $C(t^k)$ can be chosen for task t^1 until the success ratio constraint is met.

(3) Selection of the first candidate. As shown in Fig.3, t^1 and D will be added into V' if t^1 is chosen as the first candidate. Then t^2, A and B will be added into V' during the process of finding the shortest path between t^2 and set V'. Finally, $V' = \{A, B, D\}$, and $C\_MST(V') = 1$. As all we know, $C\_MST(V') = 0.5$ if t^2 is selected as the first candidate. Thus, we should choose t^{rarest} as the first candidate, where $t^{rarest} = \arg\min_{t^k \in P} |C(t^k)|$.

(4) The clean-up process. Usually, the final team might contain redundant candidates for the same task. This is because a few candidates in $C(t^k)$ might be added in V' successively. Thus, the chosen team must go through the clean-up process to further optimize the result if redundant nodes exist.

(5) Time complexity. It is clear that RarestSteiner consists of four parts. In the first phase, we add virtual nodes and edges, the running time of this part is $O(pC_{max})$, where C_{max} is the largest cardinality of $C(t^k)$, $1 \leq k \leq p$, in project P, and the worst case is $O(pn)$. The second phase is to choose t^{rarest} and add it into set V', the running time is $O(p)$. The third part calculates v^*, in the worst case, all candidates in graph G are added into V', thus the running time is $O(n^3)$.

The clean-up procedure costs time $O(pn^2)$. Thus, the time complexity of our proposed algorithm for the TFP-ST problem is $O(n^3)$.

C. THE PROPOSED ALGORITHM FOR THE TFP-WT PROBLEM

Since the TFP-WT problem is similar to the MCC-MD problem, and Arkin and Hassin [23] have proved that there is no any polynomial approximation solution to the MCC-MD problem with a bounded error guarantee. Thus, there is no any polynomial approximation solution to our TFP-WT problem with a bounded error guarantee.

Algorithm 3 RarestFirstWT

Input: $G = (V, E, w)$, $C(t^k)$, M_i, $P = \{t^1, t^2, \ldots, t^p\}$, $1 \leq i \leq n$, $1 \leq k \leq p$.
Output: Team V^{best}; best-r.
$V^{best} = \emptyset$.
$best\_r = 0$.
$R = P$.
$t^{rarest} = \arg\min_{t^k \in P} |C(t^k)|$.
for every seed in $C(t^{rarest})$ **do**
  $V' = \emptyset$.
  $V' = \{seed\}$.
  $relation = \infty$.
  $R = \{t^k || V' \bigcap C(t^k)| == 0, t^k \in P\}$.
  while $R \neq \emptyset$ **do**
    $C(R) = \cup_{t^k \in U} C(t^k)$.
    $v^* = \arg\max_{u \in C(R) \setminus V'} d(u, V')$.
    $relation = \min(relation, d(v^*, V')$.
    $V' = V' \cup \{v^*\}$.
    $R = \{t^k || V' \bigcap C(t^k)| == 0, t^k \in P\}$.
  end while
  if $relation > best\_r$ **then**
    $V^{best} = V'$.
    $best\_r = relation$.
  end if
end for
return V^{best},$best\_r$.

We propose a heuristic algorithm to solve the TFP-WT problem. The proposed algorithm for the TFP-WT problem, called as RarestFirstWT, is shown in Algorithm 3. We first choose the task with the lowest cardinality, i.e., $t^{rarest} = \arg\min_{t^k \in P} |C(t^k)|$. Then we select a candidate from $C(t^k)$ as a seed, and add it into a temporary solution V', Let R denote the set of tasks that are not covered yet by the chosen candidates. We add incrementally candidates into V' according to the following rules. The set of tasks in R form a new set $C(R) = \cup_{t^k \in R} C(t^k)$. We choose the candidate, which has maximum distance to current team V', in $C(R) \setminus V'$, add the special candidate v^*, where $v^* = \arg\max_{u \in C(R) \setminus V'} d(u, V')$, into V', and update the minimum collaboration cost of the team finally. We can get a temporary team this way. For every seed in $C(t^{rarest})$, we can get such a temporary team, and then we select the team with the biggest $C\_Min(V')$ as the best team.

We compute the distance matrix of any two candidates in graph G, and store this matrix based on hashtable, which will reduce the running time of the proposed algorithm. The running time of the whole loop is $O(pn^2)$. Thus, the running time of our solution to the TFP-WT problem, RarestFirst, is $O(C(t^{rarest}) \cdot pn^2)$. Finally, we have $O(C(t^{rarest}) \cdot n^2)$ for the time complexity of our proposed algorithm because p is far less than n.

VI. PERFORMANCE EVALUATION

In this section, we evaluate the performance of our proposed algorithms, referred to as RarestStener and RarestFirstWT in the following figures, respectively. We implement a greedy-based algorithm as the benchmark for comparison.

A. THE DBLP DATASET

In this paper, we use the dataset provided by [26], which is applied usually for the team formation problem. The social network structure is extracted from the DBLP data, which records the papers published by different researchers in field of computer science. Social relationship among authors as well as the areas of expertise of authors can be extracted from paper publication. Wang *et al.* [26] provide a data set which contain lots of computer conferences, including $DB = \{SIGMOD, VLDB, ICDE, ICDT, PODS\}$, $T = \{SODA, FOCS, STOC, STACS, ICALP, ESA\}$, $DM = \{WWW, KDD, SDM, PKDD, ICDM, WSDM\}$, $AI = \{IJCAI, NIPS, ICML, COLT, UAI, CVPR\}$. We assume that the author can be a candidate if he or she published at least three papers in upon conferences, social connection between two authors exists if they published at least two papers collaboratively, and the keywords of papers are regarded as tasks.

We calculate the collaboration cost between any two authors i and j as the Jaccard distance. Let P_i and P_j denote the set of publications of author i and j, respectively, then the collaboration cost between i and j can be computed as $1 - \frac{|P_i \bigcap P_j|}{|P_i \bigcup P_j|}$. Unfortunately, we find the data provided by [26] generates a disconnected graph. Thus, we preprocess the network structure, and only used its biggest connected subgraph to conduct our experiments. As a result, we obtain a social network consisting 5880 candidates, 14205 edges and 4090 tasks. We sort tasks based on their cardinality in descending order.

B. EXPERIMENTAL ENVIRONMENT

Each simulation are conducted in our experiments on a PC with an Intel(R) Core(TM) i5-3470 CPU, a RAM of 8GB and a Windows OS. The algorithms are implemented in Python 2.7. The Networkx tool is used to handle the calculation of the graph.

C. EXPERIMENTS FOR THE TFP-ST PROBLEM

We evaluate the proposed algorithm for the TFP-ST problem, and compare the performance of RarestSteiner with a greedy-based algorithm.

1) THE GREEDY-BASED ALGORITHM

We implement a greedy-based algorithm, referred to as EnhancedGreedy in the following figures, as the benchmark for comparison. EnhancedGreedy is a intuitive solution to the TFP-ST problem. Let R denote the tasks whose requirements are not yet met, and $C(R) = \cup_{t^k \in R} C(t^k)$ denote the set of candidates. At first, the candidate $v_0 = \arg\max_{i \in C(R)} |M_i \bigcap R|$ is added into set V'. Then this algorithm iteratively adds other candidates to form a solution. The selected candidate is the one who has the biggest utility, and the utility function is defined as follows:

$$E(i) = \frac{\sum_{t^k \in P} |f_{task}(V' \bigcup \{Path(i, V')\}, t^k)|}{C\_MST(V' \bigcup \{Path(i, V')\})} - \frac{\sum_{t^k \in P} |f_{task}(V', t^k)|}{C\_MST(V' \bigcup \{Path(i, V')\})} \quad (10)$$

where $f_{task}(V', t^k)$ represents the utility produced by task t^k for team V', defined as the following equation. The success ratio of t^k in V' is calculated as $p^k = 1 - \prod_{i \in V'}(1 - p_i^k)$. If the success ratio for t^k is met, the income is 1, otherwise 0.

$$f_{task}(V', t^k) = \begin{cases} 1, & p^k \geq S^k \\ 0, & otherwise \end{cases}$$

At each iteration, candidate i with biggest utility is chosen, where $i^* = \arg\max_{i \in C(R) \setminus V'} E(i)$, and the set of candidates $Path(i^*, V')$ is added to the current solution V'. Then we applied the clean-up procedure proposed in Algorithm 2 to delete redundant candidates.

2) PERFORMANCE EVALUATION

Recall that we define the success ratio as ratio of the skill level of chosen candidate to the quality requirement of a corresponding task. A task can be successfully accomplished only when its quality requirement is met. Accordingly, a project can be successfully accomplished only when all tasks which belong to this project are executed successfully. In this set of simulations, both project P and its quality requirement are generated randomly, let $P(t^k, S^k)$, $1 \leq k \leq p$, denote a project, where p represents the number of tasks in project P, and S^k is the requirement on the success ratio for task t^k in project P, e.g., $P(20, 0.95)$ illustrates that the project has 20 tasks, and the quality requirement for every task is 95%. To simplify performance evaluation, we assume that every task has the same requirement on quality. The success ratio of task t^k provided by member i is chosen randomly from [0.5, 1].

We compare our proposed algorithm for the TFP-ST problem, i.e., RarestSteiner, with EnhancedGreedy in terms of collaboration cost, team size and running time. Each simulation runs for 500 times on aforementioned experimental environment.

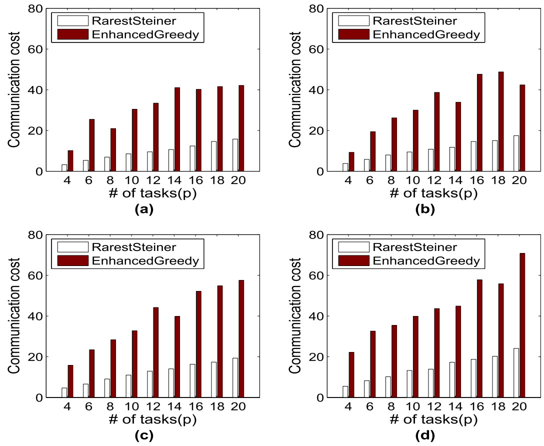


FIGURE 4. Collaboration cost comparison. (a) success ratio is 80%. (b) success ratio is 85%. (c) success ratio is 90%. (d) success ratio is 95%.

Fig.4 illustrates the team collaboration cost, i.e., min C_MST, under different quality requirements and number of tasks. Compared with the benchmark, RarestSteiner can achieve lower communication cost. The collaboration cost of a team, generated by both RarestSteiner and EnhancedGreedy, increases with number of tasks as well as the quality requirement. As expected, bigger S and more tasks will result in a bigger team. Reasonably, the more team members, the higher collaboration cost is.

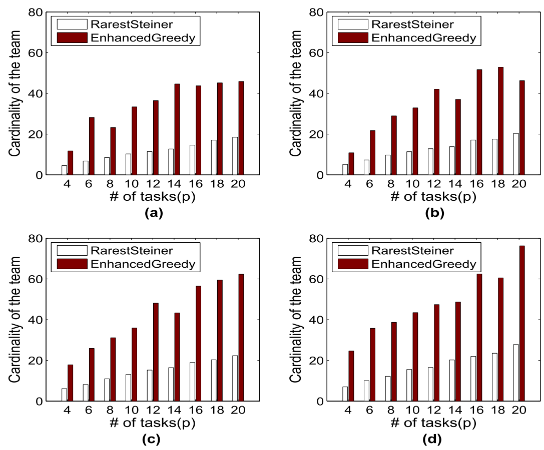


FIGURE 5. Team size comparison. (a) success ratio is 80%. (b) success ratio is 85%. (c) success ratio is 90%. (d) success ratio is 95%.

Actually, the cardinality of a team is equivalent to the size of this team. As shown in Fig.5, RarestSteiner can produce a smaller size for the final team, and EnhancedGreedy will result in a bigger cardinality of the chosen team. This is because that RarestSteiner can select candidates from V more efficiently. We can also know that the team size grows along with the growth of p and S. When the number of tasks and their requirements on success ratio go up, more candidates need to be recruited. Therefore, the cardinality of the team, i.e., the size of the team, is getting bigger and bigger.

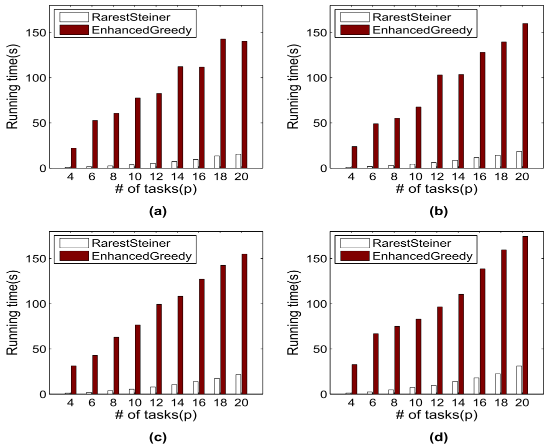


FIGURE 6. **Running time comparison. (a) success ratio is 80%. (b) success ratio is 85%. (c) success ratio is 90%. (d) success ratio is 95%.**

Fig.6 reveals the difference between RarestSteiner and EnhancedGreedy in terms of running time under different success ratio and number of tasks. The running time of RarestSteiner increases along with the requirement on success ratio and number of tasks. Though RarestSteiner and EnhancedGreedy have the similar increasing trend, the running time of RarestSteiner is much shorter than that of the benchmark. The running time should be longer because more task and higher requirements result in a bigger number of iterations.

D. EXPERIMENTS FOR THE TFP-WT PROBLEM

We evaluate the proposed algorithm for the TFP-ST problem, and compare the performance of RarestFirstWT with a greedy-based algorithm.

1) THE BENCHMARK ALGORITHM

We implement a greedy-based algorithm, referred to as GreedyWT in the following figures, as the benchmark for comparison. GreedyWT is a intuitive solution to the TFP-WT problem. Let R denote the set of tasks whose requirements are not yet satisfied, and $C(R) = \cup_{t^k \in R} C(t^k)$ denote the set of candidates. Firstly, a candidate in $C(R)$ is randomly chosen and added into V'. Then GreedyWT iteratively adds other node to V'. The chosen candidate is the one who bring the largest utility. The utility function is defined as follows:

$$E(i) = C - \sum_{t^k \in P} |f_{task}(V', t^k)| - Min(V' \bigcup \{i\}) \cdot \sum_{t^k \in P} |f_{task}(V' \bigcup \{Path(i, V')\}, t^k)| \qquad (11)$$

where $f_{task}(V', t^k)$ presents the utility related to task t^k for team V', defined as the following Equation. If the task t^k can be served by team V' successfully, the utility is 1, and 0 otherwise.

$$f_{task}(V', t^k) = \begin{cases} 1, & |V' \cap C(t^k)| > 0 \\ 0, & otherwise. \end{cases}$$

At every step, candidate i^* with largest utility is found, where $i^* = \arg\max_{i \in C(R) \setminus V'} E(i)$, and node i^* is added to the temporary solution V'.

2) PERFORMANCE EVALUATION

In this set of simulations, both project P and its quality requirement are generated randomly, let $P(p)$ denote a project, where p represents the number of tasks in project P, e.g., $P(20)$ indicates that the project has 20 tasks. To simplify performance evaluation, we assume that every task has the same requirement on success ratio.

We compare our proposed algorithm for the TFP-WT problem, i.e., RarestFirstWT, with GreedyWT in terms of collaboration cost, team size and running time. Each simulation runs for 500 times on aforementioned experimental environment.

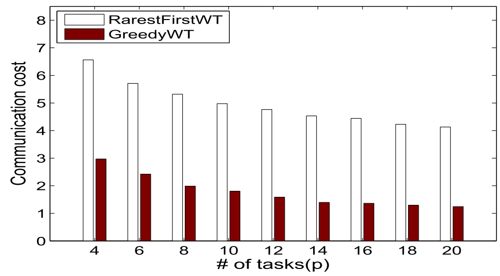


FIGURE 7. **Communication cost comparison.**

Fig.7 reveals the general trend of collaboration cost under different number of tasks. Compared with the benchmark, GreedyWT, RarestFirstWT can obtain higher communication cost. The communication cost of the team, produced by both RarestFirstWT and GreedyWT, raise along with the number of tasks. The reason for this is that a majority of social connections among team members are weak ties, or no connection exists. Fortunately, it happened to be our original intention to find a team with weak ties, such that objective of the result can be guaranteed for a task, e.g., proposal review.

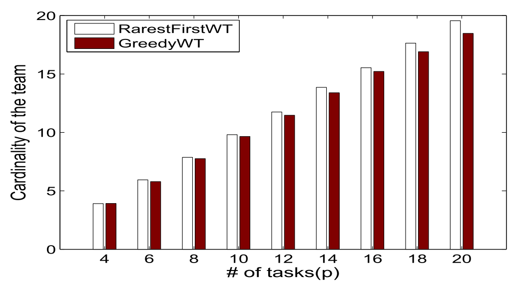


FIGURE 8. **Team size comparison.**

Fig.8 shows the difference of team size, i.e. the cardinality of the team, under different number of tasks. Team size generated by RarestFirstWT is a little bit larger than that of EnhancedGreedy. We can know that the team size goes up along with the growth of p, this is because more candidates are needed to serve more tasks.

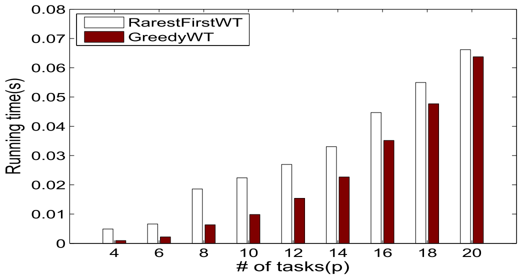


FIGURE 9. **Running time comparison.**

Fig.9 illustrates running time comparison of two algorithms under different number of tasks. After pre-computing the distance matrix between any two candidates in graph G using hashtable, the running time is quite small. Obviously, the running time grows along with the growth of p. As shown in Fig. 9, the running time of RarestFirstWT is longer than GreedyWT. Since we need to find several candidates according to different seed nodes and choose the one with largest C_Min, this may induce longer running time. Though the running time of RarestFirstWT is much longer than that of benchmark, combining with the results of Fig. 7 and Fig.8, it is clear that there is a tradeoff between objective of results and performance improvement. In comparison, RarestFirstWT leverages every candidate for a fair result and a multidimensional view, and take a better advantage of social connection to choose the best candidate, resulting in a team which is composed of different cliques.

VII. CONCLUSIONS

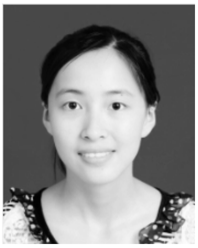
In this paper, we investigated the social connection aware team formation problem for participatory tasks. Due to the nature of tasks, we distinguished the social connection between strong ties and weak ties, and then formulated the TFP-ST problem and the TFP-WT problem, respectively. Since both the TFP-ST problem and the TFP-WT problem are NP-hard, we propose two corresponding heuristic algorithms to solve these two problems. Through extensive simulation results, we verified that our proposed algorithms can achieve desired performance.

REFERENCES

[1] H. Zhou, S. Xu, D. Ren, C. Huang, and H. Zhang, "Analysis of event-driven warning message propagation in vehicular ad hoc networks," *Ad Hoc Netw.*, vol. 55, pp. 87–96, Feb. 2017.
[2] H. Zhou, H. Wang, X. Li, and V. Leung, "A survey on mobile data offloading technologies," *IEEE Access*, vol. 6, pp. 5101–5111, Jan. 2018.
[3] T. Lappas, K. Liu, and E. Terzi, "Finding a team of experts in social networks," in *Proc. KDD*, Paris, France, 2009, pp. 467–476.
[4] A. Baykasoglu, T. Dereli, and S. Das, "Project team selection using fuzzy optimization approach," *Cybern. Syst.*, vol. 38, no. 2, pp. 155–185, Feb. 2007.
[5] E. L. Fitzpatrick and R. G. Askin, "Forming effective worker teams with multifunctional skill requirements," *Comput. Ind. Eng.*, vol. 48, no. 3, pp. 593–608, May 2005.
[6] H. Wi, S. Oh, J. Mun, and M. Jung, "A team formation model based on knowledge and collaboration," *Expert Syst. Appl.*, vol. 36, no. 5, pp. 9121–9134, Jul. 2009.
[7] A. Zzkarian and A. Kusiak, "Forming teams: An analytical approach," *IIE Trans.*, vol. 31, no. 1, pp. 85–97, Jan. 1999.
[8] A. Gajewar and A. D. Sarma, "Multi-skill collaborative teams based on densest subgraphs," in *Proc. AAMAS*, Valencia, Spain, 2012, pp. 365–374.
[9] M. Kargar and A. An, "Discovering top-k teams of experts with/without a leader in social networks," in *Proc. CIKM*, Glasgow, U.K., 2011, pp. 985–994.
[10] A. Majumder, S. Datta, and K. V. M. Naidu, "Capacitated team formation problem on social networks," in *Proc. KDD*, Beijing, China, 2012, pp. 1005–1013.
[11] H. Yin, B. Cui, and Y. Huang, "Finding a wise group of experts in social networks," in *Proc. ADMA*, Beijing, China, 2011, pp. 381–394.
[12] H. Zhou, C. M. V. Leung, C. Zhu, S. Xu, and J. Fan, "Predicting temporal social contact patterns for data forwarding in opportunistic mobile networks," *IEEE Trans. Veh. Technol.*, vol. 66, no. 11, pp. 10372–10383, Nov. 2017.
[13] B. Ashenagar, A. Hamzeh, N. F. Eghlidi, and A. Afshar, "A fast approach for multi-objective team formation in social networks," in *Proc. IKT*, Urmia, Iran, 2015, pp. 1–6.
[14] S. S. Rangapuram, T. Bühler, and M. Hein, "Towards realistic team formation in social networks based on densest subgraphs," in *Proc. WWW*, Rio de Janeiro, Brazil, 2013, pp. 1077–1088.
[15] A. Anagnostopoulos, L. Becchetti, C. Castillo, A. Gionis, and S. Leonardi, "Online team formation in social networks," in *Proc. WWW*, Lyon, France, 2012, pp. 839–848.
[16] M. Kargar, A. An, and M. Zihayat, "Efficient Bi-objective team formation in social networks," in *Proc. ECML PKDD*, Bristol, U.K., 2012, pp. 483–498.
[17] M. Kargar, M. Zihayat, and A. An, "Finding affordable and collaborative teams from a network of experts," in *Proc. SDM*, Austin, TX, USA, 2013, pp. 587–595.
[18] Y. Yang and H. Hu, "Team formation with time limit in social networks," in *Proc. MEC*, Shenyang, China, 2013, pp. 1590–1594.
[19] C. T. Li and M. K. Shan, "Team formation for generalized tasks in expertise social networks," in *Proc. PASSAT-SOCIALCOM*, Minneapolis, MN, USA, 2010, pp. 9–16.
[20] A. Bhowmik, V. Borkar, D. Garg, and M. Pallan, "Submodularity in team formation problem," in *Proc. SDM*, Philadelphia, PA, USA, 2014, pp. 893–901.
[21] H. Sun *et al.*, "Analyzing expert behaviors in collaborative networks," in *Proc. KDD*, New York, NY, USA, 2014, pp. 1486–1495.
[22] G. Reich and P. Widmaye, "Beyond Steiner's problem: A VLSI oriented generalization," in *Proc. WG*, Castle Rolduc, The Netherlands, 1989, pp. 196–210.
[23] E. M. Arkin and R. Hassin, "Minimum-diameter covering problems," *Network*, vol. 36, no. 3, pp. 147–155, Oct. 2000.
[24] H. Takahashi and A. Matsuyama, "An approximate solution of the Steiner problem in graphs," *Math. Japonica*, vol. 24, no. 6, pp. 573–577, Jan. 1980.
[25] P. Winter, "Steiner problem in networks: A survey," *Network*, vol. 17, no. 2, pp. 129–167, Jan. 1987.
[26] X. Wang, Z. Zhao, and W. Ng, "A comparative study of team formation in social networks," in *Database Systems for Advanced Applications*. Cham, Switzerland: Springer, 2015, pp. 389–404.

XIAOYAN YIN received the Ph.D. degree in computer science from Northwestern Polytechnical University, Xi'an, China, in 2010. She is currently an Associate Professor with the School of Information Science and Technology, Northwest University, Xi'an. Her research interests are congestion control, optimization, QoS supports for wireless sensor networks, social networks, and Internet of Things.

CHAO QU received the B.S. degree in electronics science and technology and the M.S. degree in communication and information system from Northwest University, Xi'an, China, in 2013 and 2016, respectively. He is currently a Software Engineer. His research interests include social network and wireless sensor network.



QIANQIAN WANG received the B.S. degree in software engineering from Northwest University, Xi'an, China, in 2016, where she is currently pursuing the M.S. degree. Her main research interests include social networks and wireless networks.

FAN WU received the B.S. degree in computer science from Nanjing University in 2004 and the Ph.D. degree in computer science and engineering from the University at Buffalo, The State University of New York, in 2009. He is currently a Professor with the Department of Computer Science and Engineering, Shanghai Jiao Tong University. His research interests include algorithmic game theory, wireless networks, and mobile computing.

BAOYING LIU received the B.A. degree in western culture from Xi'an Foreign Languages University, China, in 2001, the M.A. degree in western culture and history from Northwest University, Xi'an, China, in 2004, and the M.A. degree in cultural heritage preservation and tourism environment and the Ph.D. degree in cultural heritage form the University of Salento, Italy, in 2005 and 2012, respectively. She is currently with Northwest University. She is the talent introduced by Hundred-Talent Program.

FENG CHEN received the M.S. degree in computer science from Northwest University, Xi'an, China, in 2007, and the Ph.D. degree in computer science from Northwestern Polytechnical University, Xi'an, in 2012. He is currently a Faculty Member with Northwest University, Xi'an. His research interests are in the area of wireless networks, social networks, and Internet of Things.

XIAOJIANG CHEN received the Ph.D. degree in computer software and theory from Northwest University, Xi'an, China, in 2010. He is currently a Full Professor with the School of Information Science and Technology, Northwest University. His current research interests include network and software security, localization, social networks, and Internet of Things.

DINGYI FANG is currently a Professor with the School of Information Science and Technology, Northwest University, Xi'an, China. His current research interests include mobile computing and distributed computing systems, network and information security, localization, social networks, and wireless sensor networks.