World Wide Web (2020) 23:153–174
https://doi.org/10.1007/s11280-019-00712-x

Incorporating geographical location for team formation in social coding sites

Liang Chen[1] · Yongjian Ye[1] · Angyu Zheng[1] · Fenfang Xie[1] · Zibin Zheng[1] · Michael R. Lyu[2]

Received: 30 October 2017 / Revised: 30 June 2019 / Accepted: 22 July 2019 /
Published online: 12 September 2019

Abstract
With the proliferation of open source software and community, more and more developers from different background (e.g., culture, language, location, skill) prefer to work collaboratively and release their works in social coding sites (e.g., Github). Given a collaborative project with a set of required skills, it is an important and challenging task to form a team of developers that have not only the required skills but also the minimal communication cost. Previous works mainly leverage historical collaboration records among team members to model the communication cost, while ignoring the impact of geographical location of each developer. In this paper, we aim to exploit and incorporate the geographical information to improve the performance of team formation in social coding sites. Specifically, we conduct two objective functions for the collaboration records and geographical proximity correspondingly, and propose two optimization algorithms. Comprehensive experiments on a real-world dataset (e.g., GitHub) demonstrate the performance of the proposed model with the comparison of some state-of-the-art ones.

Keywords Team formation · Geographical location · Social coding sites · Genetic algorithm

1 Introduction

Commercial softwares are generally developed by team collaboration in companies. A successful software development not only heavily relies on the core competence of members but also hinges on the efficiency and quality of their collaboration. Traditional software companies usually take the way of writting examinations or interviews to select members that may meet the required skills in the corresponding task. However, these methods are time-consuming and ignore the historical collaboration records among members.

✉ Zibin Zheng
zhzibin@mail.sysu.edu.cn

Extended author information available on the last page of the article.

Since the social coding site[1] is becoming increasingly popular with developers from different countries and regions, it accumulates a large number of users who are willing to show off their own projects or contribute their skills to open source projects cooperatively. On the one hand, it is practical and easy to evaluate the competence of skills based on his/her coding on GitHub. On the other hand, it presents the collaboration network which provides more detail about his/her cooperative experience. Our goal is to take full advantage of information of skills mastery and collaborations posted in GitHub to form an optimal team so as to meet the task requirements. This problem is called TEAM FORMATION problem.

For team formation problems, given a social coding network $G = (V, E)$ and a set of $skills$ required for $T = \{skill_1, skill_2, ..., skill_n\}$, the goal of this problem is to find an optimal team $Team_t = \{v_1, v_2, ..., v_k\}$ in a pool of individuals $V = \{v_1, v_2, ..., v_m\}$ to complete the task. Each User v is marked with skills $skills(u) = \{skill_1, skill_2, ..., skill_q\}$ that have been mastered. The weights of the edges E in graph G can be interpreted as a kind of indicator to measure how closely the corresponded individuals communicate or collaborate with each other.

The problem of finding a team of experts from a network by minimizing the communication cost has been tackled in [2, 10, 13, 21, 22, 26]. In this paper, we additionally consider the impact of geographical information in the team formation problem. Geographical information can affect the cost of communication. Developers collaborate with the ones located closely with bringing less communication cost due to the same/similar culture background, language, and time zone. Therefore, taking geographical information into consideration will be useful for an effective team formation.

As shown in Figure 1, a simple social coding network contains three developers and each individual connects to several skills that he/she has mastered. The left corner presents some target projects that need to find a set of developers to complete the projects. Each project has some required skills. In addition, the social network presents the locations among users, User A located in North America while user B and user C both reside in southeast Asia. In this scenario, the development of Bootstrap involves two skills JS and CSS, and two teams (i.e., $\{A, B\}$, $\{B, C\}$) are able to meet the skill requirements. However, team $\{B, C\}$ is much more appropriate than the other due to the geographical advantage.

In this paper, we believe that the cost of communication or collaboration is heavily connected with geographical correlation. Since Lappas et al. [13] have proved that the team formation problem is NP-hard, we propose two solutions based on the idea of genetic algorithm. For the first solution, we take both collaboration and geography into account, giving two objective cost functions and propose a variant algorithm based on non-dominated sorting to optimize it. For the second one, we combine two objective cost into one and solve it by a genetic algorithm. The experimental results have shown that our proposed algorithms outperform the state-of-the-art ones.

In particular, the main contribution of this paper could be summarized as follows:

– In this paper, we investigate both collaboration and geography in team formation problem and propose two objective cost functions to solve problem. We employ a variant algorithm based on non-dominated sorting for multi-objective optimization. In addition, we combine two objective cost into one and use a genetic algorithm to optimize it. For these two standpoints of optimization, we make a detailed analysis and comparation with their advantages and disadvantages.

[1] https://github.com/

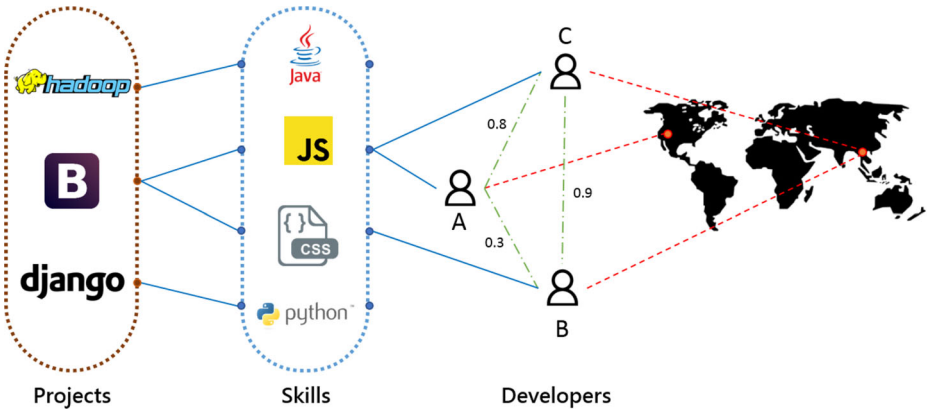

Figure 1 Schema of geographical information based team formation

– We conduct a set of experiments to evaluate the performance of our algorithms. We crawl a large scale of dataset from GitHub, constructing the social network graph with information of skills, collaboration and geography. By conducting experiments on this dataset, the result has shown that our algorithm achieves a better performance in terms of converging near the Pareto-optimal front and maintaining diversity among obtained solutions.

The rest of this paper is organized as follows: Section 2 highlights the related work of team formation problem. Section 3 provides the problem definition and models for communication cost and geographical information. Section 4 introduces two proposed optimization algorithms. Section 5 describes the experimental results. Finally, Section 6 concludes the paper.

2 Related work

2.1 Team formation

Team formation problem is widely studied in the field of computer assisted cooperative systems. Lappas et al. [13] first formally define the TEAM FORMATION problem in presence of a social network of individuals and first take the communication cost into consideration for finding a team to satisfy the project requirements. In their article, given a set of skills, their goal is to find the most optimal team which has the lowest communication cost. Besides, Balog et al. [3] state that the profile of experts should not only consider their individual skills, but also a description of their collaboration network. Some papers put forward several factors and constraints in the follow-up extensions considering various real scenarios. Gionis et al. [1] take into account the work load of experts in tasks and propose the first online algorithms to group teams in a fair allocation of the workload by minimizing the load on team members. Majumder et al. [18] add capacity constrains into the team formation problem, avoiding the overload in assignments among users. In [15], the authors specify the number of experts for each skill in given task. Kargar et al. [10] study the issue about discovering a team of experts and its leader from a social network. Li et al. [16] pay more

attention in finding the optimal candidate to replace a team member who becomes unavailable to complete the task. Basiri et al. [4] use brain drain optimization to solve the team formation problem. Farhadi et al. [7] propose a novel approach based on the skill grade of experts to improve the efficiency and quality of task accomplishment. Kargar et al. [11] propose three heuristic algorithms based on different intuitions to solve the bi-objective team formation problem. Recently, some new questions about team formation are raising. Li et al. [17] combine the ideas of team formation and influence maximization to formulate a novel research problem, i.e., Influential Team Formation.

In terms of geographic information used in team formation, a small amount of papers have relevant research. Rangapuram et al. [20] consider geographical information as a constraint in team formation by stipulating that the distance between any two experts should not be larger than given value. Han et al. [9] prefer teams that belong to the same country. Differently, in our papers, we enrich the definition of geographical proximate, and make a tradeoff between geographical proximity and overall communication cost in forming a optimal team to complete the task. Most of existing research still do not or just simply consider geographic information. To the best of our knowledge, we are the first one to explore the geographic information for team formation problem, and consider it as single-objective and multi-objective optimization problems respectively.

2.1.1 Social network

The solution of team formation problem has been proposed in the operation research community, involving branch and bound, genetic algorithms and simulated annealing [5, 8, 23]. A trend in this line of work is to take into account the social graph structure of the individuals in the process of team formation. In social network, each vertex indicates one of the experts with a set of skills, and the value of the edge records the communication cost between any two experts.

Recently, location-based social networks have been widely studied [27]. Some research focus on location based POI (Point of Interest) recommendation [25, 28]. This kind of work explores check-in data and supports real-time recommendation. Yin et al. [32] propose a latent class probabilistic generative model to learn location based interest information. Yin et al. [31] propose a spatial-aware hierarchical collaborative deep learning model and introduced a late feature fusion strategy into the model to deal with the multimodal heterogeneous features of the POI. In [29, 30], location based social network is studied and the behavior prediction methods are proposed.

The estimation of communication cost in social networks has been pursued by many researchers. Previous work [24] in social networks mainly focus on binary friendship relations. These binary versions lead to a coarse indication of relationship. In many existing researches, the mainstream of relationship strength is to set the edges of social networks as non-negative value. However, it is practical that social interaction involves positive and negative relations between any two people. Positive relationships indicate the approval or support of each other, while negative relationships indicates the distrust or reluctance to cooperate. Huttenlocher et al. [14] study online socials with positive or negative relationships. Such models used a mix of positive and negative links have a good performance. However, we focus on the number of times of cooperation between any two people due to lack of peer interactions and rich cooperation information in our dataset, where the links in every two users will be assigned to positive value.

2.1.2 Communication cost for a team

In order to determine which team is better than others, it is necessary and practical to extend the definition of communication cost for a team with more than two users. In social network, any subgraph can be formed a team, and therefore the definition of communication cost for a team is based on the subgraph. Generally, there are several measures used to estimate the communication cost for a team [13, 18]. One is to consider the diameter of the subgraph as the overall communication cost for a team, which is the largest shortest path between any two vertices in the subgraph. Another estimation is to calculate the minimum spanning tree (MST) of the subgraph [13]. Since the diameter and MST does not measure the cost of all the required communication, a cost function called Sum of Distances are widely popular in relevant research, using the sum of the shortest distances between any two experts [10].

For the solution of team formation in social network, Lappas et al. [13] and Kargar et al. [10] have proved the team formation problem is NP-hard and they also propose several approximate algorithms for their solution, the RarestFirst algorithm for the diameter problem and Cover-Steiner and Enhanced-Steiner algorithms for MST problem. However, the social network of team formation problem is large and sparse, it is not suitable to calculate the diameter or MST of a graph. Since the problem is combinatorial and it has been proved to be np-hard, we apply genetic algorithm (GA) [12] and a non-dominated sorting based multi-objective evolutionary algorithm (NSGA-II) [6, 12] to find the optimal team.

2.2 Geographical information

Geographical proximity is generally considered as an important factor in successful collaboration. In [19], the authors elaborate the relation between geographical proximity and knowledge exchange, believing that geographical proximity can be beneficial for building mutual trust due to frequent interaction and face-to-face contacts. Which helps overcome possible difficulties came from differences in institutional or organizational backgrounds. There are some reasons in supporting the importance of geography in collaboration: 1) greater physical distances generate more cost in communication; 2) different language and culture constrain interaction of two people in different countries; 3) social ties between coauthors, like in any social network, are geographically biased.

3 Model

Given a set of experts and the communication costs between them, we can form an undirected graph $G = (V, E, w)$ to present a social coding network. $v \in V$ denotes an expert and $e(v_i, v_j) \in E$ denotes that a pair of experts v_i and v_j have participated in common projects before. $w(v_i, v_j)$ denotes the weight between v_i and v_j. We assume that the weight between two experts is related to the fraction of projects they contribute to together, then w is defined as:

$$w(v_i, v_j) = 1 - \frac{|N_{v_i} \cap N_{v_j}|}{|N_{v_i} \cup N_{v_j}|} \tag{1}$$

where N_v denotes the set of projects in which expert v is listed as a contributor and the range of w is [0, 1]. The *Communication Cost (cc)* is defined as the sum of weights on the shortest path between two experts. If the graph is not connected, the communication cost is defined as $+\infty$.

3.1 Model the communication cost

To measure the communication cost among the experts in a team, we define *Sum of Communication Cost (SCC)* as:

Given a team T, we can generate a complete graph $G(T)$ where the edges in $G(T)$ represent cc between each pair of the experts in T. Sum of Communication Cost can be calculated as:

$$SCC(T) = \sum_{i=1}^{n} \sum_{j=i+1}^{n} cc(e_i, e_j) \tag{2}$$

where T denotes a specified team with n experts and e_i denotes the i-th expert in team T (Corresponding to v_i in (1)).

3.2 Model the geographical location

To measure the geographical proximity of the team of experts, Geographical proximity (gp) and Sum of Geographical Proximity (SGP) is defined in [9].

Geographical proximity measures the distance between two regions:

$$gp(u, v) = \begin{cases} 0, & \textit{If u and v in the same country} \\ 1. & \textit{Otherwise} \end{cases} \tag{3}$$

where u and v represents the specified two experts. Sum of Geographical Proximity measures the geographical proximity of the team of experts:

$$SGP(T) = \sum_{i=1}^{n} \sum_{j=i+1}^{n} gp(e_i, e_j) \tag{4}$$

where T denotes a specified team with n experts and e_i denotes the i-th expert in team T.

The above method simply takes into account whether two experts are in the same country (or region). While many other geographical based factors affect the team formation performance, such as language, time zone, etc. Meanwhile, people in some countries(or regions) communicate more closely, such as European countries, China mainland and Hong Kong, etc. So we propose a new model named *Multi-view Sum of Geographical Proximity (MSGP)* by considering the above factors as below:

Region view We take the above method as a view named *Region View*, which indicates whether two experts are in the same country. *Region Proximity (rp)* and *Sum of Region Proximity (SRP)* is defined as:

$$rp(u, v) = gp(u, v) \tag{5}$$

$$SRP(T) = \sum_{i=1}^{n} \sum_{j=i+1}^{n} rp(e_i, e_j) \tag{6}$$

Language view Although English is becoming the universal language of the world, communicating in native language can greatly improve work efficiency. So we consider language as an important view and define *Language Proximity (lp)* and *Sum of Language Proximity (SLP)* as:

$$lp(u, v) = \begin{cases} 0, & \textit{If u and v speak the same language} \\ 1. & \textit{Otherwise} \end{cases} \tag{7}$$

$$SLP(T) = \sum_{i=1}^{n} \sum_{j=i+1}^{n} lp(e_i, e_j) \tag{8}$$

We use the official language of the country of the expert as his or her language. Noting that some countries have a variety of official languages, we specify the most widely used one.

Time zone view The world is divided into 24 time zones (from -12 to +12). People in the same or adjacent time zones can collaborate better. So we introduce the time zone view into our geographical model and define *Time Zone Proximity (tp)* and *Sum of Time Zone Proximity (STP)*:

$$tp(u, v) = \frac{1}{12} \times \begin{cases} (|t_u - t_v|), & |t_u - t_v| < 12 \\ 24 - (|t_u - t_v|), & otherwise \end{cases} \tag{9}$$

$$STP(T) = \sum_{i=1}^{n} \sum_{j=i+1}^{n} tp(e_i, e_j) \tag{10}$$

where t_u denotes the time zone (UTC) of expert u and $tp \in [0, 1]$. Noting that some countries cross several time zones (e.g. Russia, Australia), we use the time zone of their capitals.

History records view The above views measure the proximity of the two regions using the objective attributes of them instead of history records. Obviously proximity of two regions in the past can be shown by history records. So we define *History Proximity (hp)* and *Sum of History Proximity (SHP)* as:

$$hp(u, v) = 1 - \frac{|N_{r_u} \cap N_{r_v}|}{|N_{r_u} \cup N_{r_v}|} \tag{11}$$

$$SHP(T) = \sum_{i=1}^{n} \sum_{j=i+1}^{n} hp(e_i, e_j) \tag{12}$$

where r_u denotes the region of expert u and N_r denotes the set of projects in which experts in region r are listed as contributors.

Multi-view sum of geographical proximity We consider the views above and use them to measure geographical proximity. A linear model is used to combine these views:

$$MSGP = \alpha_R SRP + \alpha_L SLP + \alpha_T STP + \alpha_H SHP \tag{13}$$

4 Optimization

The team formation model above is a bi-objective problem, so we conduct two optimization algorithms treating it as a single-objective (si-objective) problem and a multi-objective problem respectively. Genetic Algorithm (GA) and Non-dominated Sorting Genetic Algorithm II (NSGA-II) based methods are as follows:

4.1 Genetic algorithm for si-objective problem

The team formation problem above is a bi-objective problem and the proposed model has been proved to be NP-hard problem. Because of its nonlinearity and noncontinuous, it is hard to find a general mathematical programming methods to gain the optimum solution. There are many famous evolution approaches to solve the problem, such as genetic algorithm, simulated annealing, ant colony optimization. We choose Genetic Algorithm (GA) for the following reasons: 1) The search domain of GA is wide, so the optimum value can be found more easily. 2) GA is fit for the scenario of team formation, because the chromosomes is able to model the teams naturally as the genes model the skills.

4.1.1 Combined objective function

It is a bi-objective optimization problem to form a team of experts with the minimized communication cost as well as geographical proximity among them. The target is to optimize $SCC(T)$ and $MSGP(T)$ simultaneously. A common method is to convert the bi-objective optimization problem into a si-objective optimization problem via combining the objective functions linearly. Combined Cost Function (CCF) can be defined as:

$$CCF = (1 - \lambda) \times SCC(T) + \lambda \times MSGP(T) \tag{14}$$

where parameter λ varying from 0 to 1 indicates the tradeoff between $SCC(T)$ and $MSGP(T)$.

4.1.2 Preparation

Given the required skills and corresponding number of people, our goal is to find a set of people, in which the members not only meet the skill requirements of the task, but also have low geographical and communication cost. Applied to genetic algorithm in our problem, population is the collection of possible solution, in which each chromosome is represented as a candidate team who can complete the task. Here, the chromosomes are synonymous with the individual or member. So the length of chromosome equals to the number of people required in task and each gene in chromosome indicates a member in team who master at least a skill in task requirement. In Figure 2, given required skills $skill_1$, $skill_2$, $skill_3$

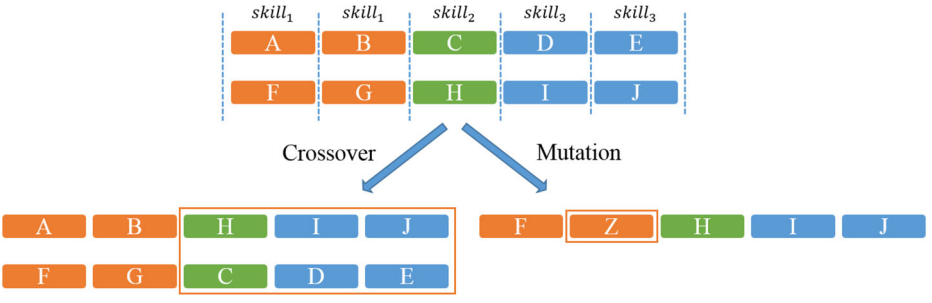

Figure 2 Genetic algorithm

and corresponding number of required members, we exemplify a chromosome in which developers meet the skill requirements.

4.1.3 Initialization

For the beginning, we need to generate a population as the first generation for subsequent evolution, typically containing several hundreds or thousands of possible solutions. We assign an appropriate value to the population size and initialize same number of chromosomes randomly.

4.1.4 Selection

According to the natural law of survival of the fittest, individuals in any generation have different survival and reproductive abilities. We measure these abilities by calculating their fitness. The higher the fitness, the more likely it is to reproduce. Individuals selected by fitness will participate in the generation of offspring through genetic operators, which includes crossover and mutation.

4.1.5 Genetic operators

In the process of reproduction, genetic operators is used to generate the next generation population of solutions from those selected. It contains two parts: crossover and mutation. By using these two methods, a new solution is created shared many of the characteristics of its "parents". After repeatedly creations, solutions is accumulated and finally form the next population.

Crossover Crossover is one of a genetic operator to vary the programming of chromosomes from one generation to the next. On a certain probability P_C, the allelomorph on two chromosome will interchange in reproducing. In other words, several developers in one team will be exchanged with that of the same positions in the other team results in two candidate teams. Crossover benefits delivering several genes of parents to the next generation, which may preserve good genes in evolution.

Mutation Mutation, the other genetic operator, is to produce new genes in the next generation. Several individuals in chromosome will be replaced by other individuals who mastered the same skills occurred at a certain probability P_m. Mutation helps generate new offspring which increases the diversity in generation.

Main loop Algorithm 1 shows the pseudocode of the Genetic Algorithm for the team formation problem with geographical information. In the first step, we initialize the population P_{ini} as the first generation and assign it to the current population P_{cur}. Later it will go into a loop stimulating evolution in which each iteration will reproduce a new generation from the current population P_{cur}. In every iteration, we calculate the fitness of each individual in current population P_{cur}. Select top-k individuals $S_k(c) = \{c_1, c_2, ..., c_k\}$ who have the

highest fitness in P_{cur} and add directly to the next generation. From the rest of P_{cur}, select two individuals randomly participate in crossover and mutation to fulfill the new population according to the possibility P_c and P_m. Until convergence, the algorithm completes the operation and finally obtains the optimal solution.

Algorithm 1 Genetic algorithm.

Input: sets of chromosome $S(c)$, population size s, a crossover possibility p_c, a mutation possibility p_m ,parameter k
Output: output a optimal chromosome

```
Initial P_ini ;
P_cur ← P_ini ;
while true do
    P_new = ∅ ;
    foreach individual ∈ P_cur do
        calculate individual_fitness ;
    end
    select top-k individuals S_k(c) of highest fitness in P_cur;
    add S_k(c) to P_new;
    for i in range(s) do
        select parents c_Alice and c_Bob ;
        if random() > p_c then
            c_Alice, c_Bob ← crossover(c_Alice, c_Bob) ;
        end
        if random() > p_m then
            c_Alice, c_Bob ← mutation(c_Alice, c_Bob) ;
        end
        add c_Alice, c_Bob to P_new ;
    end
    P_cur ← P_new ;
    if Converge then
        break;
    end
    select an individual c_opt of highest fitness in P_cur ;
    print c_opt ;
end
```

4.2 NSGA-II for bi-objective problem

Since the team formation problem above is a bi-objective optimization problem, we employ NSGA-II to find multiple Pareto-optimal solutions. NSGA-II is one of the multi-objective evolutionary algorithms. Compared with other algorithms like PAES, SPEA and NSGA, NSGA-II has better spread of solutions and better convergence near the Pareto-optimal front. This section we will introduce several modules which is an important part of the algorithm and give the main loop to combine these modules into a complete algorithm.

4.2.1 A fast non-dominated sorting approach

For the population, we need to apply non-dominated sorting so as to categorize these solutions. We divide the population into several layers of non-domination. Individuals in each layer of non-domination dominate ones in the subsequent layers. We begin initialize an empty temporary set to stimulate the first front and put a random individual p_r from population into the set. Through the loop, we find the individuals belong to non-dominated class in population and consider it as the first front. Then we repeat the process to find the subsequant fronts.

Since each solution need to compare with other solutions, the worse case that only one solution in each front requires at most $O(N^2)$ computations. For M objectives, the entire computations is $O(MN^2)$.

Algorithm 2 Fast-NonDominated-sorting-approach.

```
Input: population p_cur
Output: several
P_tmp ← P_cur ;
Front = { p_r } ;
for p ∈ P_tmp do
    for q ∈ Front do
        if p dominates q then
            Add p to Front ;
            Front = Front - {q} ;
        end
    end
end
Fronts.add(front) ;
P_tmp = P_tmp -{ Front } ;
```

4.2.2 Density estimation and crowded comparison

For each front obtained by above algorithm, we will introduce a formula to estimate the density of individuals in same front. First, we sort the solutions in front using the value of objective m. Then we calculate each individual in sorted front. Define d_m as the difference of the value of objective functions m between solutions $p-1$ and solution $p+1$, which is on either side of solution p, and $p_{distance}$ as (also call crowding distance) accumulated value on the d_m of each objective function m of solution i. Specially, the crowding distance of first and the last solutions will be assigned to a max distance. Algorithm 2 presents the calculation of crowding distance.

In order to decide which is better in any two solutions in fronts, we define that solution a is better than solution b if a is in the higher front than b or the density of a is larger than that of b in the same front. To be more formular, define $a \prec b$ if $a_{frontOrder} < b_{frontOrder} \vee (a_{frontOrder} = b_{frontOrder} \wedge a_{distance} > b_{distance})$. Now we can sort the solutions in fronts by this regulation.

The worse case that all solutions are in one front requires $O(mNlogN)$ computations.

Algorithm 3 Calculate crowding distance.

Input: a certain front $front$, max distance max_d
Output: crowding distance of each individual in front

```
foreach p ∈ Front do
    p_distance = 0
end
size = len(front) ;
foreach m ∈ objective do
    sort(Front) ;      /* according to the value of objective m */
    front[0]_distance = max_d ;
    front[size − 1]_distance = max_d ;
    for 0 < i < len do
        d_m = front[i+1].m - front[i-1].m;
        front[i]_distance = front[i]_distance +d_m;
    end
end
```

Algorithm 4 Complete algorithm.

Input: a certain front $front$, max distance max_d
Output: crowding distance of each individual in front

```
Initialize population P_cur /* the same as GA                    */
while Evolving() do
    Fronts = Fast-NonDominated-sorting-approach(P_cur) sortedSolutions = [ ] ;
    foreach Front ∈ Fronts do
        calculate-crowding-distance(Front,max_d);
        solutions = sort(Front, cmp = density-sort-cmp) ;
        sortedSolutions.append(solutions);
    end
    parents = sortedSolutions[0:N-1];
    crossover and mutation to generate the next population P_new;
    P_cur = P_cur + P_new;
end
```

4.2.3 Complete algorithm

For a start, we initialize the population P_{cur} randomly as the first generation. we sort the P_{cur} using fast non-dominated sorting approach and select the first N solutions to be the ”parents”. These selected parents will participate in crossover and mutation results in the ”children”. Combining the ”parents” and ”children”, we will gain the next generation. Repeat the above process to stimulate evolution and finally converge the Pareto-optimal solutions.

Since the total algorithm is related with several parts, containing non-dominated with $O(MN^2)$ computations, calculations of crowding distance with O(MNlogN) and sort of solutions with $O(2Nlog(2N))$, the overall complexity of the above algorithm is $O(MN^2)$.

5 Experiments

5.1 Dataset

The experiments are conducted on a real-word dataset from GitHub. It is one of the most popular Web-based Git or version control repository and Internet hosting service. Users in GitHub often collaborate on projects, and then naturally establish a coding social network. The GitHub dataset can meet the requirements of our experiment well.

The GitHub dataset is crawled by using the APIs which are provided by GitHub. Our dataset include 28,362,019 projects, 15,647,255 users and the relationships between them. Because only some users provide their geographical location information, we filter these users out and label the country (or region) of them. As a result, we finally obtain 36,701 users and 3,532,453 projects in our dataset.

5.2 Analyzation

The focus of this paper is geographic information, so the following is a statistical analysis of different aspects of geographic information.

Figure 3 shows the proportion of developers in different countries. It can be seen that most of developers are from USA. And then, the second order and the third order is UK and China respectively. Moreover, developers from USA, UK, China and India account for more than half of all developers. From the distribution of developers' countries, we can observe that developers from USA are the most active users of GitHub. Possible reason is that GitHub is well known for USA people. Another observation is the diversity of developers' countries. This reflects the popularity of GitHub.

Figure 4 shows the proportion of developers speak different languages. We can observe that English is the most popular language among developers (e.g, 57.0 %). The followed languages are Chinese and German. The reason may be that English is the most widely used language in the world. The diversity of languages further indicates the popularity of GitHub. Obviously, different language is not a major obstacle but still an obstacle to working together.

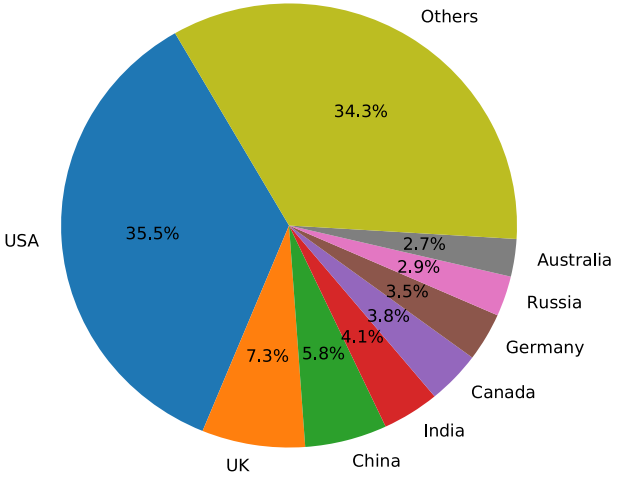

Figure 3 Top Countries

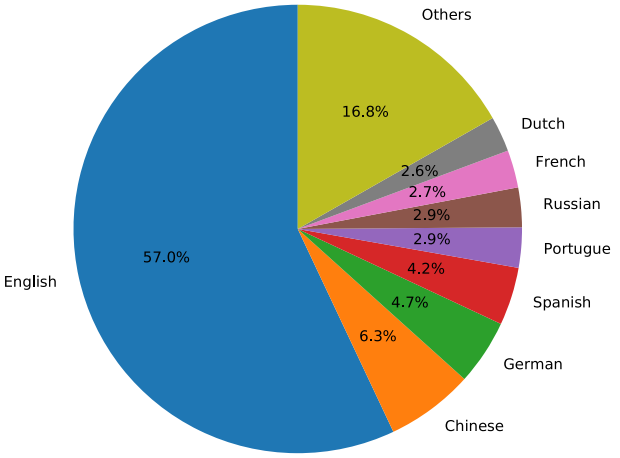

Figure 4 Top Languages

Figure 5 shows the number of developers in different time zones. It can be seen that developers are mainly distributed around -5, +1 and +8 time zones. The time zones are corresponding to North America, Europe and Asia (and Australia). Developers in the same time zone will cooperate better.

Figure 6 shows the diversity of geographical location. We make statistics on the region of contributors of all the teams (projects) in GitHub dataset. The x-axis denotes the number of contributor's regions of one team and the y-axis denotes the percentages of each kind of team. As can be seen that about 36% teams whose contributors are from one country (or region) and less than 50% teams whose contributors are from more than two countries. The corresponding percentage of project contributors reduces quickly with the increase of diversity. A conclusion can be drawn that developers tend to work with people from the same region.

To analyze the degree of cooperation between developers in different countries, we calculate the History Proximities (hp) of every pair of countries. hp is an important indicator to evaluate the tightness of developers in two countries. The smaller the hp value, the more likely they will be going to work together. Table 1 shows that developers in European countries and American countries work more closely. For example, the hp value of Russia and USA is 0.885 which is larger than that of USA and UK (e.g., hp value between them is 0.677).

5.3 Setup

This paper considers programming languages in GitHub dataset as skills. If a developer contributes to a project with a specified skill (language), we match the developer to the skill.

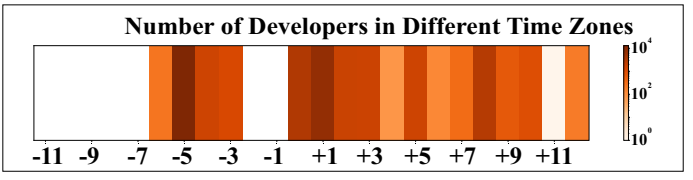

Figure 5 Number of developers in different time zones

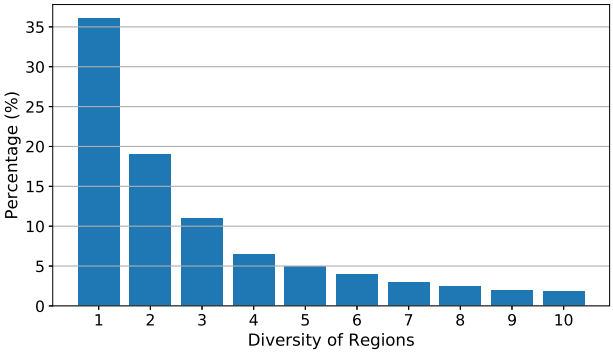

Figure 6 Diversity of geographical location

For all the experiments, we set the number of skills as 10 and λ is set as 0.05 for comparison. The population size of GA is set as 200 and the generation is set as 100. Consider the different effect of the views that were analyzed in Section 5.2, we set α_R, α_L, α_T and α_H as 0.2, 0.15, 0.15 and 0.5. It is worth noting that all these parameters are set experimentally, under which our algorithm can achieve the best performance.

5.3.1 Metrics

In this section, we use three metrics which are commonly adopted in conventional studies to evaluate the performance of the GA-Based and NSGA-II-Based algorithms. The metrics are listed as follows:

Sum of communication cost (SCC) SCC measures the communication cost of the team. It indicates the efficiency of the communication between the team developers.

Multi-view sum of geographical proximity (MSGP) $MSGP$ measures the geographical proximity of the team. It indicates the closeness of the team at the geographical level.

Combined cost (CombCost) $CombCost$ combines Sum of Communication Cost and Multi-view Sum of Geographical Proximity. It indicates the trade-off between the two metrics above.

Table 1 History proximities

Country A	Country B	hp
USA	UK	0.677
Canada	UK	0.686
UK	Germany	0.693
Germany	Canada	0.698
Germany	Australia	0.705
Russia	Netherlands	0.737
...		
Russia	USA	0.855

5.3.2 Baselines

Kargar et al. [11] studied the same problem as this paper and the dataset structure is similar, so we choose the two models proposed in this work as baselines and compare our proposed model with them:

Approximation rare algorithm (ARA) Approximation Rare Algorithm selects one skill with the least supporters. An expert with the skill is firstly selected as a seed expert. And then an expert is added with the minimum communication cost to the seed expert with each of other required skills into the team. Finally, a team is created with the minimum cost.

Minimum cost contribution rare algorithm (MCCR) Minimun Cost Contribution Rare Algorithm selects an expert with the skill with the rarest supporters as the initial member of the team. Then a series of team members are added by considering their communication cost comparing with current team members. Consequently, a team is created with the minimum cost.

5.4 Results

NSGA-II produces a set of candidate teams, while the other algorithms produce only one team. We compare GA with ARA, MCCR and NSGA-II respectively.

5.4.1 Performance of GA

In this experiment, we set the number of skills to 10. Figures 7 and 8 shows the performance of ARA, MCCR and GA. The following conclusion can be observed:

On the SCC metric: GA-based model achieves better performance. The reason is that GA-based model has a large search space while the other algorithms has a small one. But the advantage of GA-based model is not very obvious. The reason is that the

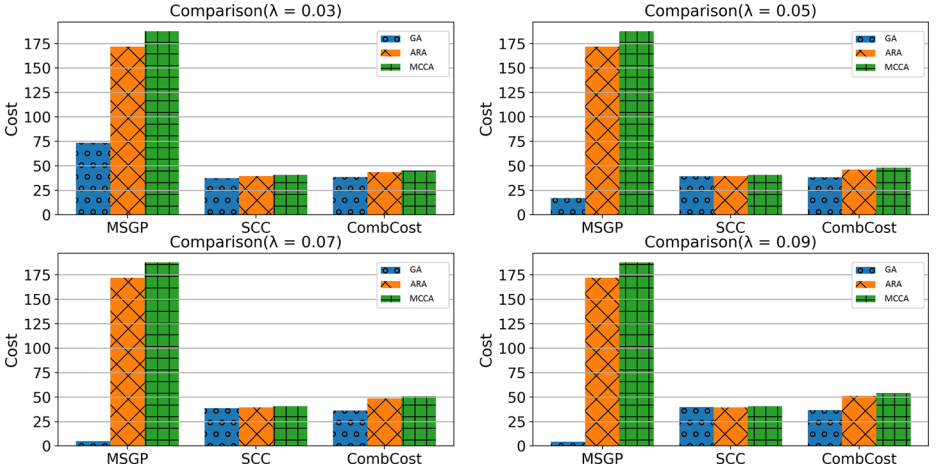

Figure 7 Performance comparison

GA-based model takes into account the geographical factor instead of optimizing the communication cost only, so it cannot achieve a great advantage on the SCC metric.

On the MSGP metric: GA-based model achieves better performance obviously. The reason is that GA-based model takes into account the geographical factor while the other models do not. As the value of λ increases, GA-based model performs better on $MSGP$ due to the increase of the weight of geographical factor.

On the CombCost metric: GA-based model also achieves better performance obviously. The reason is that GA-based model considers both communicate cost and the geographical factor, then it can get a good balance between the two factors by adjusting the parameter λ. As the value of λ increases, GA-based model also performs better on $CombCost$ due to the increase of the weight of geographical factor.

Stability: The performance of all models depends on the random seeds to a certain extent, so the models are not stability. Figure 8 shows the standard deviation of all models on the three metrics. It is obvious that GA-based model has a lower standard deviation on all metrics. The reason is that GA-based model has a larger search space and it can effectively reduce the model variance. The two baseline models have similar and large standard deviation on the $MSGP$ metrics, because they do not take into account the geographical factor.

5.4.2 Impact of number of skills

To evaluate the impact of number of skills, we set skills number from 2 to 10 with a step value of 2. Figure 9 shows that as the number of skills increases, the cost of all algorithms are going to higher and higher with different metrics. But the proposed GA-based model can always achieve the best performance. It indicates that GA-based model has good stability when the number of skills changes.

5.4.3 Comparison of GA and NSGA-II

NSGA-II algorithm produces a set of candidate teams instead of manually specifying the λ value. The teams in the set have good performance under each objective, a team can be selected according to requirements.

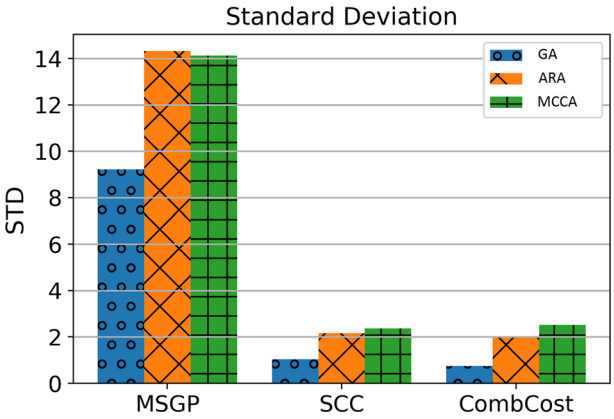

Figure 8 Standard Deviation of the models

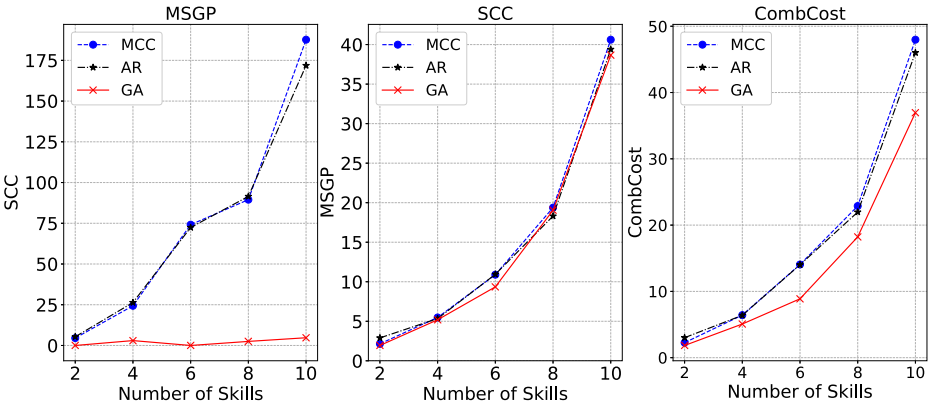

Figure 9 Impact of number of skills

In order to evaluate the performance of GA and NSGA-II with different λ values, we set λ from 0.0 to 0.1 with the step value of 0.01 and optimize the objective function SCC with the corresponding λ. It is worth noting that NSGA-II does not need to specify λ, so we select the team in the candidate set with the minimum SCC.

Performance Figure 10 shows the performance of GA-based model and NSGA-II-based model with different λ. The NSGA-II model achieves better performace with all λ. When the value of λ increases, the advantage of NSGA-II model tends to be more obvious. The reason is that NSGA-II can search the optimal solution by considering the two objects simultaneously.

Discussion of λ Although the tradeoff parameter λ varies from 0 to 1, SCC and MSGP are different in magnitude. In our approach, GA is a multi-objective optimization problem integrated two objectives. It will choose developers from the same region with a MSGP value of 0, when the λ value is more than 0.1. Obviously, the larger λ value has no significance for

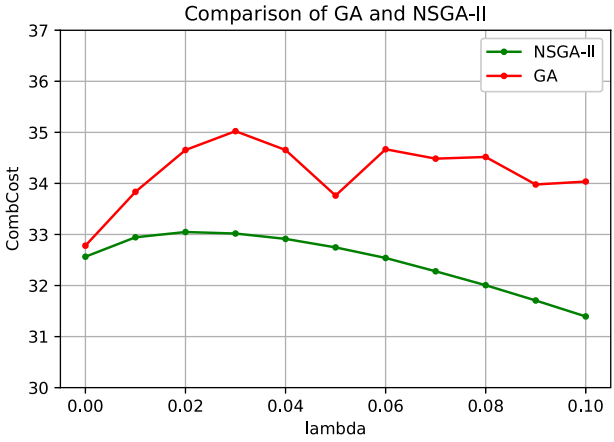

Figure 10 Performance of GA and NSGA-II

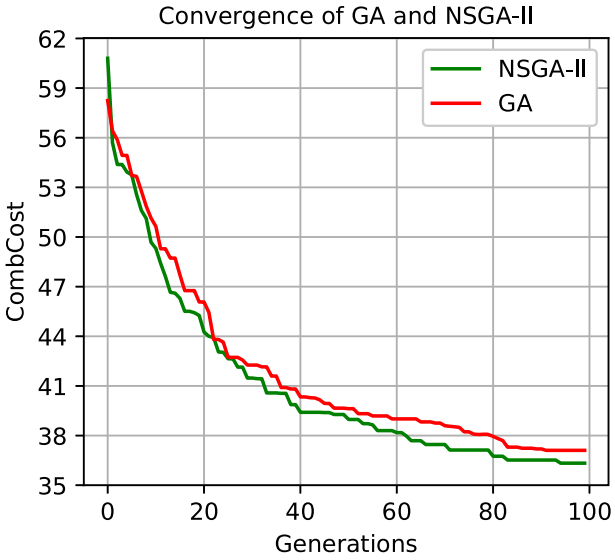

Figure 11 Convergence of GA and NSGA-II

this multi-objective optimization problem. Therefore, we vary it from 0 to 0.1. The choice of λ depends on the requirement of team formation. If a team with less communication cost is need, a smaller λ is needed. If a team with less geographical distance, a larger λ is needed. When the requirement is not certain, a NSGA-II algorithm can be conducted to generate a series of alternative teams. One team can be chosen manually in light of the actual conditions.

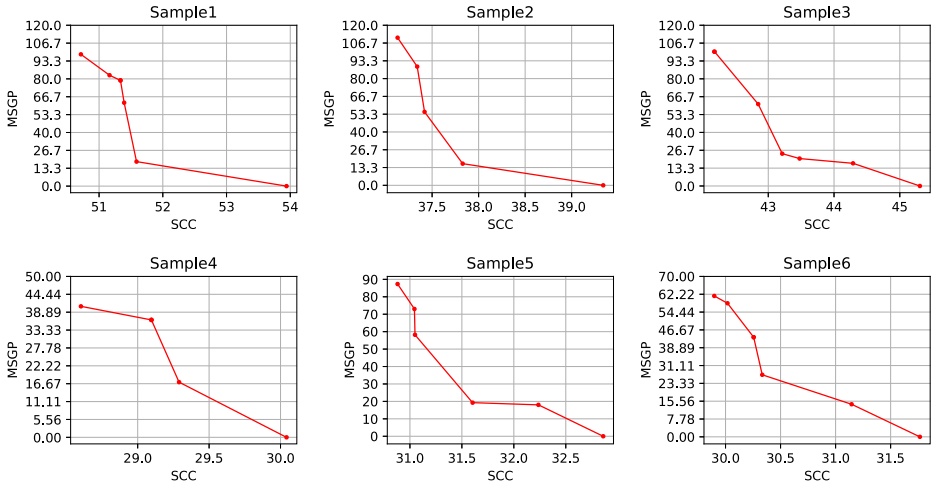

Figure 12 Relation of SCC and $MSGP$ in NSGA-II

Convergence Figure 11 shows the convergence of GA-based model and NSGA-II-based model. The two models tend to converge when the generation is greater than 50. Furthermore, NSGA-II model always performs better than GA model.

Features of NSGA-II NSGA-II gives a set of candidate teams with different SCC and $MSGP$. To explore the relationship between SCC and $MSGP$, we sample a set of candidate teams of six different team formation tasks. Figure 12 shows that SCC and $MSGP$ are negatively correlated. It is because that the target of NSGA-II is to seek a balance between the two objectives. When a team has both larger SCC and $MSGP$ than another team, it cannot be selected as a candidate.

6 Conclusion and future work

In this paper, we exploit and incorporate geographical information to improve the performance of team formation and solve the bi-objective optimization problem in two different ways. Specifically, we investigate both collaboration records and geographical information, and propose two objective cost functions. For the first optimization way, we employ a variant algorithm based on non-dominated sorting for multi-objective optimization. For the second one, we combine these two objective cost functions into a unified one, and use a genetic algorithm for optimization. To comprehensively evaluate the performance of the proposed methods, we crawl a large scale of dataset from Github, including 28,362,019 software projects and 15,647,255 users. The experiments demonstrate the performance of the proposed models with the comparison of state-of-the-art ones.

In our future work, we plan to exploit the impact of social media on team formation. Social media (e.g., Facebook, Twitter) information could be used to further depict the relationship among the developers and to improve the performance of team formation.

Acknowledgements The work described in this paper was supported by the National Key Research and Development Program (2017YFB0202200), the National Natural Science Foundation of China (61702568, U1711267), the Program for Guangdong Introducing Innovative and Enterpreneurial Teams (No.2017ZT07X355) and the Fundamental Research Funds for the Central Universities under Grant (17lgpy117).

References

1. Anagnostopoulos, A., Becchetti, L., Castillo, C., Gionis, A., Leonardi, S.: Online team formation in social networks. In: Proceedings of the 21st International Conference on World Wide Web, pp. 839–848. ACM (2012)
2. Ashenagar, B., Eghlidi, N., Afshar, A., Hamzeh, A.: Team formation in social networks based on local distance metric. In: International Conference on Fuzzy Systems and Knowledge Discovery (FSKD), pp. 946–952 (2015)
3. Balog, K., De Rijke, M.: Determining expert profiles (with an application to expert finding). IJCAI **7**, 2657–2662 (2007)
4. Basiri, J., Taghiyareh, F., Ghorbani, A.: Collaborative team formation using brain drain optimization: A practical and effective solution. World Wide Web **20**(6), 1385–1407 (2017)
5. Baykasoglu, A., Dereli, T., Das, S.: Project team selection using fuzzy optimization approach. Cybern. Syst.: Int. J. **38**(2), 155–185 (2007)
6. Deb, K., Agrawal, S., Pratap, A., Meyarivan, T.: A fast elitist non-dominated sorting genetic algorithm for multi-objective optimization: NSGA-II. In: International Conference on Parallel Problem Solving from Nature, pp. 849–858. Springer, Berlin (2000)

7. Farhadi, F., Sorkhi, M., Hashemi, S., Hamzeh, A.: An effective expert team formation in social networks based on skill grading. In: 2011 IEEE 11th International Conference on Data Mining Workshops, pp. 366–372. IEEE (2011)
8. Fitzpatrick, E.L., Askin, R.G.: Forming effective worker teams with multi-functional skill requirements. Comput. Indust. Eng. **48**(3), 593–608 (2005)
9. Han, Y., Wan, Y., Chen, L., Xu, G., Wu, J.: Exploiting geographical location for team formation in social coding sites. In: Pacific-Asia Conference on Knowledge Discovery and Data Mining, pp. 499–510. Springer, Cham (2017)
10. Kargar, M., An, A.: Discovering top-k teams of experts with/without a leader in social networks. In: Proceedings of ACM International Conference on Information and Knowledge Management, CIKM 2011, pp. 985–994 (2011)
11. Kargar, M., An, A., Zihayat, M.: Efficient bi-objective team formation in social networks. Mach. Learn. Knowl. Discov. Databases, 483–498 (2012)
12. Konak, A., Coit, D.W., Smith, A.E.: Multi-objective optimization using genetic algorithms: A tutorial. Reliab. Eng. Syst. Safety **91**(9), 992–1007 (2006)
13. Lappas, T., Liu, K., Terzi, E.: Finding a team of experts in social networks. In: Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 467–476. ACM (2009)
14. Leskovec, J., Huttenlocher, D., Kleinberg, J.: Predicting positive and negative links in online social networks. In: Proceedings of the 19th International Conference on World Wide Web, pp. 641–650. ACM (2010)
15. Li, C.T., Shan, M.K., Lin, S.D.: On team formation with expertise query in collaborative social networks. Knowl. Inf. Syst. **42**(2), 441–463 (2015)
16. Li, L., Tong, H., Cao, N., Ehrlich, K., Lin, Y.R., Buchler, N.: Replacing the irreplaceable: Fast algorithms for team member recommendation. In: Proceedings of the 24th International Conference on World Wide Web, pp. 636–646. International World Wide Web Conferences Steering Committee (2015)
17. Li, C.T., Huang, M.Y., Yan, R.: Team formation with influence maximization for influential event organization on social networks. World Wide Web, 1–21 (2017)
18. Majumder, A., Datta, S., Naidu, K.: Capacitated team formation problem on social networks. In: Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 1005–1013. ACM (2012)
19. Ponds, R., Van Oort, F., Frenken, K.: The geographical and institutional proximity of research collaboration. Papers Reg. Sci. **86**(3), 423–443 (2007)
20. Rangapuram, S.S., Buhler, T., Hein, M.: Towards realistic team formation in social networks based on densest subgraphs. In: Proceedings of the 22nd International Conference on World Wide Web, pp. 1077–1088. ACM (2013)
21. Wang, X., Zhao, Z., Ng, W.: A comparative study of team formation in social networks. In: 2015 International Conference on Database Systems for Advanced Applications (DASFAA), pp. 389–404 (2015)
22. Wang, X., Zhao, Z., Ng, W.: USTF: A unified system of team formation. IEEE Trans. Big Data **2**(1), 70–84 (2016)
23. Wi, H., Oh, S., Mun, J., Jung, M.: A team formation model based on knowledge and collaboration. Expert Syst. Appl. **36**(5), 9121–9134 (2009)
24. Xiang, R., Neville, J., Rogati, M.: Modeling relationship strength in online social networks. In: Proceedings of the 19th International Conference on World Wide Web, pp. 981–990. ACM (2010)
25. Xie, M., Yin, H., Wang, H., Xu, F., Chen, W., Wang, S.: Learning graph-based poi embedding for location-based recommendation. In: Proceedings of the 25th ACM International on Conference on Information and Knowledge Management, pp. 15–24. ACM (2016)
26. Yang, Y., Hu, H.: Team formation with time limit in social networks. In: International Conference on Mechatronic Sciences (MEC), pp. 1590–1594 (2013)
27. Yin, H., Cui, B.: Spatio-Temporal Recommendation in Social Media. Springer, Singapore (2016)
28. Yin, H., Cui, B., Zhou, X., Wang, W., Huang, Z., Sadiq, S.: Joint modeling of user check-in behaviors for real-time point-of-interest recommendation. ACM Trans. Inf. Syst. (TOIS) **35**(2), 11 (2016)
29. Yin, H., Hu, Z., Zhou, X., Wang, H., Zheng, K., Nguyen, Q.V.H., Sadiq, S.: Discovering interpretable geo-social communities for user behavior prediction. In: 2016 IEEE 32nd International Conference on Data Engineering (ICDE), pp. 942–953. IEEE (2016)
30. Yin, H., Chen, H., Sun, X., Wang, H., Wang, Y., Nguyen, Q.V.H.: SPTF: A scalable probabilistic tensor factorization model for semantic-aware behavior prediction. In: 2017 IEEE International Conference on Data Mining (ICDM), pp. 585–594. IEEE (2017)

31. Yin, H., Wang, W., Wang, H., Chen, L., Zhou, X.: Spatial-aware hierarchical collaborative deep learning for POI recommendation. IEEE Trans. Knowl. Data Eng. **29**(11), 2537–2551 (2017)
32. Yin, H., Zhou, X., Cui, B., Wang, H., Zheng, K., Nguyen, Q.V.H.: Adapting to user interest drift for poi recommendation. IEEE Trans. Knowl. Data Eng. **28**(10), 2566–2581 (2016)

Affiliations

Liang Chen[1] · Yongjian Ye[1] · Angyu Zheng[1] · Fenfang Xie[1] · Zibin Zheng[1] · Michael R. Lyu[2]

Liang Chen
chenliang6@mail.sysu.edu.cn

Yongjian Ye
yeyj7@mail2.sysu.edu.cn

Angyu Zheng
20142005031@m.scnu.edu.cn

Fenfang Xie
xieff5@mail2.sysu.edu.cn

Michael R. Lyu
lyu@cse.cuhk.edu.hk

1 School of Data and Computer Science, Sun Yat-Sen University, Guangzhou, China

2 Department of Computer Science and Engineering, The Chinese University of Hong Kong, Shatin, N.T., Hong Kong