YORK UNIVERSITÉ UNIVERSITY
redefine THE POSSIBLE.

Affordable and Collaborative Team Formation in an Expert Network

Mehdi Kargar, Morteza Zihayat and Aijun An

Technical Report CSE-2013-01

January 24 2013

Department of Computer Science and Engineering
4700 Keele Street, Toronto, Ontario M3J 1P3 Canada

Affordable and Collaborative Team Formation in an Expert Network

Mehdi Kargar, Morteza Zihayat and Aijun An
Department of Computer Science and Engineering, York University, Toronto, Canada
{kargar,zihayatm,aan}@cse.yorku.ca

Abstract

Given an expert network, we tackle the problem of finding a team of experts that covers a set of required skills and also minimizes the communication cost as well as the personnel cost of the team. Since two costs need to be minimized, this is a bicriteria optimization problem. We show that the problem of minimizing these objectives is NP-hard. We use two approaches to solve this bicriteria optimization problem. In the first approach, we propose several (α, β)-approximation algorithms that receive a budget on one objective and minimizes the other objective within the budget with guaranteed performance bounds. In the second approach, an approximation algorithm is proposed to find a set of Pareto-optimal teams, in which each team is not dominated by other feasible teams in terms of the personnel and communication costs. The proposed approximation algorithms have provable performance bounds. Extensive experiments on real datasets demonstrate the effectiveness and scalability of the proposed algorithms.

Keywords: team formation, social networks

1 Introduction

An expert network contains a group of professionals who can provide specialized information and service. With the widespread use of the Internet, online expert networks have become popular where more and more businesses seek subject matter experts to complete a task or project. There are many expert network providers, such as Gerson Lehrman Group [1] and the Network of Experts [2]. In such networks, an expert is described by their areas of expertise, education background, location, etc. In addition, an expert can specify his/her consulting rate.

We consider the problem of finding a team of experts from such a network to complete a project. A team must possess a set of required skills in order to complete the tasks of the project. In addition, a project is usually constrained by the budgeted amount of money available for the project. Different experts may incur different fees for conducting the activities of the project. It is desirable to find a team of experts whose total cost is minimized. Furthermore, the success of a project greatly depends on how well the team members of the project communicate and collaborate with each other. Experts located in different countries may not communicate as easily as the ones living in the same city when face-to-face meetings are required. Thus, it is important to minimize the communication cost among the experts. This turns the problem into a bicriteria optimization problem.

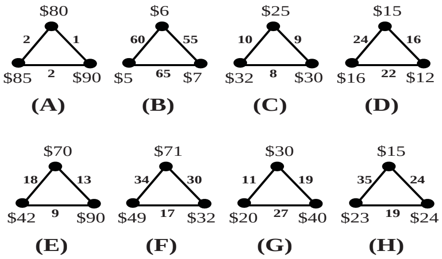

Figure 1: An example of all feasible teams.

The problem of finding a team of experts from a network which minimizes the communication cost has been tackled in [12, 10]. However, previous works in this domain did not consider the budget of the project nor the fees that may be associated with the experts. In the real world, an expert needs to be paid for his/her service, and it is preferred that the personnel cost of a project is minimized or under a budget. Only minimizing the communication cost may result in a team with high personnel cost. For example, assume that all the feasible teams of experts for a project are shown in Figure 1. Each team has three experts that together cover all of the required skills. Assume that the communication cost of a team is calculated using the sum of distances between experts in the team. The communication costs of teams A, B, C, D, E, F, G and H are 5, 180, 27, 62, 40, 81, 57 and 78, respectively. The personnel costs of these teams are \$255, \$18, \$87, \$43, \$202, \$152, \$90 and \$62, respectively. Figure 2 shows these eight teams on a diagram. If one wants to minimize only the communication cost, team A is the best. However, its personnel cost is the highest. On the other hand, if one wants to minimize the personnel cost, team B is the

[1] http://www.glgresearch.com/

[2] http://www.networkofexperts.com/

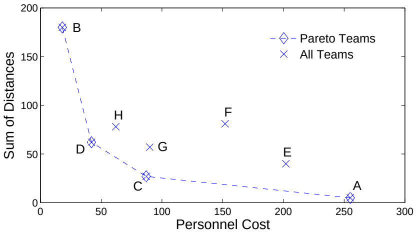

Figure 2: Feasible and Pareto optimal solutions.

best choice but has the highest communication cost. If one wants to have a team in which the members collaborate most effectively and at the same time the personnel cost is the lowest or reasonable, there is not an obvious best choice.

Clearly, there is a trade-off between the personnel cost and the communication cost. A good method should either allow the user to provide a tolerance limit on one of the objectives and produce the best answer on the other objective, or provide a set of best trade-off solutions for the user to choose from. For example, in the above example, if a budget is given on the personnel cost as $300, the best team is A. However, for budgets of $100 or $50, the best team is C or D respectively. Alternatively, if the budget is not available, we can provide users with a set of solutions that are not worse than any other solutions on both objectives. These solutions are called *Pareto-optimal solutions* [9]. Teams A, B, C and D in Figures 1 and 2 are Pareto-optimal solutions since none of them is worse than other teams on both costs. However, the remaining teams (E, F, G and H) are worse than at least one Pareto solution.

The contributions of this paper are summarized as follows. (1) We define the problem of finding an affordable and collaborative team in an expert network. We use two functions to measure the communication cost and one function to measure the personnel cost of a team. (2) We show the problem we tackle is NP-hard and propose a series of new (α, β)-approximation algorithms (to be defined later) to solve the bi-objective team formation problem, which optimizes one objective given a budget on the other objective with proved performance bounds. (3) For finding a set of Pareto-optimal solutions, a new approximation algorithm is proposed that can find solutions with guaranteed performance bounds. (4) The effectiveness and efficiency of the proposed algorithms are evaluated extensively on two large real datasets.

2 Problem Statement

Let $C = \{c_1, c_2, \ldots, c_m\}$ denote a set of m experts, and $S = \{s_1, s_2, \ldots, s_r\}$ denote a set of r skills. Each expert c_i has a set of skills, denoted as $Q(c_i)$, and $Q(c_i) \subseteq S$. If $s_j \in Q(c_i)$, expert c_i has skill s_j. In addition, a subset of experts $C' \subseteq C$ have skill s_j if at least one of them has s_j. For each skill s_j, the set of all experts having skill s_j is denoted as $C(s_j) = \{c_i | s_j \in Q(c_i)\}$. A project $P \subseteq S$ is defined as a set of skills required to complete the project. A subset of experts $C' \subseteq C$ is said to *cover* a project P if $\forall s_j \in P \, \exists \, c_i \in C', s_j \in Q(c_i)$.

The experts are connected together in a network, modeled as an undirected and weighted graph (G). Each node in G represents an expert in C. Below, terms node and expert are used interchangeably. Each node in the graph is associated with a cost representing the amount of money he/she is paid for completing a project. The cost of an expert c_i is denoted as $t(c_i)$. Two experts may be connected by an edge in the graph. The weight on an edge represents the communication cost between the two experts. The lower the weight, the more easily the two experts can collaborate or communicate, and the lower the communication cost between them. The communication cost between two experts can be defined according to the application need. For example, it can be defined as the geometric distance between two experts, which is a good communication cost measure when face-to-face meetings are needed in the project. The communication cost can also be defined by the collaboration ability or familiarity between the two experts. In this case, two nodes are connected by an edge if the experts have communicated or collaborated before, and the weight of the edge represents the strength of the relationships between the two experts. Such relationships can be obtained from social networks (such as LinkedIn), scientific collaboration networks (such as DBLP), or other sources.

DEFINITION 2.1. ***(Team of Experts [10])*** *Given a set C of experts and a project P that requires skills $s_1, s_2, \ldots,$ and s_n, a* team of experts *for P is a set of n skill-expert pairs: $\{\langle s_1, c_{s_1}\rangle, \langle s_2, c_{s_2}\rangle, \ldots, \langle s_n, c_{s_n}\rangle\}$, where c_{s_j} is an expert in C having skill s_j for $j = 1, \ldots, n$. A skill-expert pair $\langle s_i, c_{s_i}\rangle$ means that expert c_{s_i} is responsible for skill s_i in the project.*

Note that an expert in a team may be responsible for more than one required skill, that is, c_{s_i} can be the same as c_{s_j} for $i \neq j$. To evaluate the **communication cost of a team**, we define the *sum of distances* or *diameter* of a team, which has been used in [10] and [12] respectively.

DEFINITION 2.2. ***(Sum of Distances)*** *Given a team T of experts from a graph G for a project: $\{\langle s_1, c_{s_1}\rangle, \langle s_2, c_{s_2}\rangle, \ldots, \langle s_n, c_{s_n}\rangle\}$, the* sum of distances *of T is defined as*

$$sumDistance = \sum_{i=1}^{n} \sum_{j=i+1}^{n} d(c_{s_i}, c_{s_j})$$

where $d(c_{s_i}, c_{s_j})$ is the sum of weights on the shortest path between c_{s_i} and c_{s_j} in G, i.e., the shortest distance between c_{s_i} and c_{s_j}.

The use of the shortest distance in the above definition implies that the communication cost between two experts can be estimated by using their communication costs with a third expert, especially when the two experts are not directly connected. This can be easily justified when, say, travel distances are used as edge weights in the graph. In case familiarity is used to weigh an edge, the use of the shortest distance implies that two people who have not collaborated before can collaborate if they have collaborated with a third person. This can be justified by Newman's finding on scientific networks [16]: two people are much more likely to collaborate if they have both worked with a third person.

DEFINITION 2.3. ***(Diameter)*** *Given a graph G and a team of experts T consisting of some experts in G, the diameter of team T is the largest shortest distance between any two experts of T in G.*

DEFINITION 2.4. ***(Personnel Cost)*** *Let the set of experts in a team T be $\{c_1, c_2, \ldots, c_q\}$. The* personnel cost *of T is defined as:*

$$PCost(T) = \sum_{i=1}^{q} t(c_i)$$

PROBLEM 2.1. ***(Affordable and Collaborative Team Formation)*** *Given a project P and a graph G representing a network of experts, the problem of affordable and collaborative team formation is to find a team of experts T for P from G so that the communication cost of T, defined as either the sum of distances or diameter of T, and the personnel cost of T, defined as $PCost$, are minimized.*

Clearly, Problem 2.1 is a bi-criteria optimization problem. It has been proved that finding a team T of experts in a graph while minimizing the sum of distances or diameter of T is an NP-hard problem [10, 12]. Below we show that minimizing $PCost$ is also NP-hard.

THEOREM 2.1. *Finding a team of experts in a graph G to cover a set of skills while minimizing $PCost$ is NP-hard.*

Proof. Given a set of m elements (called universe) and n sets whose union is the universe, the *set cover* problem is to identify the *smallest* number of sets whose union still contains all elements in the universe. *Weighted set cover* is a variant of *set cover* in which each set has a cost associated with it and the objective is to pick sets that cover the universe with the minimum total cost. Let's consider the set of required skills in our problem as the universe and the experts in G as the sets whose union is the universe. Each expert is associated with a cost. Then, finding a team of experts while minimizing $PCost$ is equivalent to solving *weighted set cover* problem. Since the *weighted set cover* problem is NP-hard [20], finding a team of experts while minimizing $PCost$ is NP-hard.

Since minimizing the *sum of distances*, *diameter* or *personnel cost* is NP-hard, solving Problem 2.1 is NP-hard. Thus, we have to rely on approximation algorithms for solving this problem. Many (if not most) methods for solving bi-criteria optimization problems combine two objectives into a single one by using a weighted sum of two functions [11]. If the weight value is not chosen correctly, the result may not be reliable. Also, such methods are usually very sensitive to small changes in weight values [8]. In this paper we use two other approaches to solve this bicriteria problem. In the first approach, a budget value (bound) is specified on one objective and the other objective is optimized under this budget. In the second approach, the set of Pareto optimal answers [18] are found, which represent optimal trade-offs between the two objectives. Below in Section 3 we propose several (α, β)-approximation algorithms for our problem, which take the first approach, and then in Section 4 we propose an algorithm for finding a set of Pareto-optimal solutions.

3 Finding a Team of Experts with Bounded Budget

We first define the concept of (α, β)-approximation algorithm, and then propose a few (α, β)-approximation algorithms for solving our problem.

DEFINITION 3.1. *An (α, β)-approximation algorithm for an (A, B)-bicriteria problem is defined as a polynomial time algorithm that produces an answer in which the value of the first objective (A) is at most α times a budget, and the value of the second objective (B), is at most β times the minimum for any answer that is within the budget on A.*

3.1 Finding a Team of Experts with a Budget on the Communication Cost

In this subsection, we propose two algorithms for solving Problem 2.1. Both algorithms receive a budget on the communication cost of the team and minimize the personnel cost. The first algorithm uses the *diameter* and the second algorithm uses the *sum of distances* as the communication cost function.

3.1.1 Budget on the Diameter

The algorithm takes a budget on the diameter and minimizes the $PCost$ function. It is a $(2, \log n)$-approximation algorithm where n is the number of required skills of the project. The *diameter* budget is specified as D. The $(2, \log n)$-approximation means that the answer produced by the algorithm has a diameter at most twice the budget (D) and its $PCost$ value is at most $\log n$ times the cost of the minimum $PCost$ for any answer within the D diameter.

The idea of the first algorithm is as follows. It first collects the experts with the rarest required skill s_{rare} (i.e., the required skill with the least number of experts). Then, for each expert cr_i that possesses s_{rare}, all of the experts having

Algorithm 1 $(2, \log n)$-approximation algorithm for solving $(diameter, PCost)$ problem

Input: graph G, project $P = \{s_1, s_2, \ldots, s_n\}$, and budget D on the diameter.
Output: the best team and its personnel cost

```
1:  for $i \leftarrow 1$ to $n$ do
2:      $C(s_i) \leftarrow$ the set of experts with $s_i$
3:  $s_{rare} \leftarrow \arg\min |C(s_i)|, 1 \leq i \leq n$
4:  $C \leftarrow \bigcup_{i=1 \& i \neq rare}^{n} C(s_i)$
5:  $bestTeam \leftarrow \emptyset$
6:  $leastCost \leftarrow \infty$
7:  for each expert $cr_i$ in $C(s_{rare})$ do
8:      $requiredSkill \leftarrow P - Q(cr_i)$
9:      $V \leftarrow \emptyset$
10:     for each expert $c_j$ in $C$ do
11:         if $\langle d(cr_i, c_j) \leq D \rangle$ & $\langle Q(c_j) \cap requiredSkill \neq \emptyset \rangle$ then
12:             add $c_j$ to $V$
13:     $\{c_1^v, \ldots, c_q^v\} \leftarrow V$
14:     $skillV \leftarrow \bigcup_{i=1}^{q} Q(c_i^v)$
15:     if $requiredSkill \subseteq skillV$ then
16:         $team \leftarrow \{\langle q_1, cr_i \rangle, \langle q_2, cr_i \rangle, \ldots, \langle q_k, cr_i \rangle\}$ where $q_1, q_2, \ldots q_k$
            are the required skills that $cr_i$ has, i.e., $\{q_1, q_2, \ldots q_k\} = P \cap Q(cr_i)$
17:         $cost \leftarrow t(cr_i)$
18:         while $requiredSkill \neq \emptyset$ do
19:             Select $k$ s.t. $\frac{|requiredSkill \cap Q(c_k^v)|}{t(c_k^v)}$ is maximized
20:             $team \leftarrow team \cup \{\langle q_1, c_k^v \rangle, \langle q_2, c_k^v \rangle, \ldots, \langle q_k, c_k^v \rangle\}$ where
                $\{q_1, q_2, \ldots q_k\} = requiredSkill \cap Q(c_k^v)$
21:             $cost \leftarrow cost + t(c_k^v)$
22:             $requiredSkill \leftarrow requiredSkill - Q(c_k^v)$
23:         if $cost < leastCost$ then
24:             $bestTeam \leftarrow team$
25:             $leastCost \leftarrow cost$
26:         else
27:             if $cost = leastCost$ and $team.diameter <$
                $bestTeam.diameter$ then
28:                 $bestTeam \leftarrow team$
29: return $bestTeam, leastCost$
```

other required skills than s_{rare} and within D distance from cr_i are collected into a set V. A candidate team based on cr_i is then formed by including cr_i and selecting experts from V to cover all the required skills. The expert selection is a greedy procedure that iteratively selects an expert c_k^v that maximizes the ratio of the number of currently uncovered required skills covered by c_k^v to the cost of c_k^v until all the required skills are covered by the team. That is, the quality of an expert is evaluated using the number of uncovered skills per unit cost. The algorithm outputs the team that has the smallest personnel cost among all the candidate teams built around the experts with s_{rare}. If more than one team has the least cost, the one with the lowest diameter is chosen. The reason for starting a team with an expert with s_{rare} is to keep the number of candidate teams as small as possible.

The pseudo code of this approximation algorithm for solving the $(diameter, PCost)$ problem is presented in Algorithm 1. The algorithm first obtains the set $C(s_i)$ of experts having required skill s_i for each i. This can be done quickly by using a pre-built inverted index that maps a skill to its experts. In the code, $d(cr_i, c_j)$ is the shortest distance between experts cr_i and c_j, which can be efficiently obtained by consulting a pre-built index. Using a pre-built index to obtain the shortest distance between nodes has been used in other graph search methods such as the ones in [14, 19, 10]. The time complexity of Algorithm 1 is $O(|C(s_{rare})| \times (|C| + |V| \times n))$ where $|C(s_{rare})|$ is the number of experts with the rarest required skill, $|C|$ is the number of experts with other required skills, $|V|$ is the number of experts within D distance to a member of $C(s_{rare})$ and n is the number of required skills. Since the number of experts with the required skills is at most the number of all experts in G, i.e. m, the run time of the algorithm in the worst case is $O(m^2 \times n)$. However, in practice, $|C(s_{rare})|$, $|C|$ and $|V|$ are much less than m.

THEOREM 3.1. *Algorithm 1 is a $(2, \log n)$ approximation algorithm for solving $(diameter, PCost)$ problem where n is the number of required skills.*

Proof. The algorithm forms a team by using an expert cr_i with the rarest required skill and the experts within D distance from cr_i, which means that all the experts in the team are within D distance from cr_i. Assume that the two nodes that have the largest shortest distance among all pairs of nodes are n_1 and n_2. Since the shortest distance satisfies the triangle inequality, we have:

$$dist(cr_i, n_1) + dist(cr_i, n_2) \geq dist(n_1, n_2)$$

Since $dist(cr_i, n_1) \leq D$ and $dist(cr_i, n_2) \leq D$, then $dist(n_1, n_2) \leq 2 \times D$. Thus, the diameter of the team is at most $2 \times D$.

Every team should have a member cr_i with s_{rare}. Around cr_i, all of the experts with other required skills and within D distance are considered to be added into a team . Thus, all of the teams containing s_{rare} and with diameter less than or equal to D are considered by the greedy procedure in Lines 18-22. The greedy procedure tries to find the team with the minimum personnel cost by selecting experts using a heuristic that is the same as the one in the greedy algorithm for solving the weighted set cover problem [20]. It has been shown that the algorithm has a $\log n$ approximation ratio [20]. Thus, the personnel cost of the team produced by Algorithm 1 is at most $\log n$ times the cost of the optimal team with diameter at most D.

3.1.2 Budget on the Sum of Distances T

The algorithm takes a budget on the sum of distances and minimizes the $PCost$ function. It is a $(n, \log n)$-approximation algorithm where n is the number of required skills of the project. The pseudo code of this approximation algorithm for solving the $(sumDistance, PCost)$ problem is presented in Algorithm 2. The algorithm is similar to Algorithm 1, but has two major differences. First, instead of using only the rarest skill holders, this algorithm uses all the required skill holders as the seed of a candidate team. Second, for each seed (cr_i), this algorithm only considers adding its neighbors within the radius of $\frac{SD}{n-1}$ into the team, where SD is the *sumDistance* budget. The time complexity of this algorithm is $O(m^2 \times n^2)$ where n is the number of required skills and m is the number of experts in G.

THEOREM 3.2. *Algorithm 2 is a $(n, \log n)$-approximation algorithm for solving the $(sumDistance, PCost)$ problem where n is the number of required skills.*

Proof. The algorithm forms a team by using an expert cr_i which has at least one required skill and the experts within $\frac{SD}{n-1}$ distance from cr_i[3]. It means that all the experts in the team are within $\frac{SD}{n-1}$ distance from cr_i. Assume that the two nodes that have the largest shortest distance among all pairs of nodes are n_1 and n_2. Since the shortest distance satisfies the triangle inequality, we have:

$$dist(cr_i, n_1) + dist(cr_i, n_2) \geq dist(n_1, n_2)$$

Since $dist(cr_i, n_1) \leq \frac{SD}{n-1}$ and $dist(cr_i, n_2) \leq \frac{SD}{n-1}$, then $dist(n_1, n_2) \leq \frac{2 \times SD}{n-1}$. Thus, in the worst case, the maximum distance between each pair of nodes is at most $2 \times \frac{SD}{n-1}$. Since the sum of distances function sums up $\frac{(n)\times(n-1)}{2}$ pairwise distances, the maximum possible value for the sum of distances is as follows:

$$sumDistance \leq \frac{(n) \times (n-1)}{2} \times \frac{2 \times SD}{(n-1)} = n \times SD$$

Therefore, the sum of distances is at most $n \times SD$. The rest of the proof is similar to Theorem 3.1.

Note that the two approximation ratios (n and $\log n$) only occur in the worst case. We will show in Section 5 that the real ratios are much smaller than n and $\log n$ in our experiments on real data sets.

3.2 Finding a Team of Experts with a Budget on the Personnel Cost

In practice, there is often a budget on the personnel cost and the goal is to minimize the communication cost within the personnel budget. Below we propose approximation algorithms that minimize the communication cost under a personnel budget for solving the $(PCost, diameter)$ and $(PCost,\ sumDistance)$ problems.

According to [15], bicriteria problems are generally hard when the two criteria are *hostile* with respect to each other, meaning that the optimization of one criterion conflicts with the optimization of the other criterion. Two minimization objectives in our problem are hostile because the minimum value of one objective is monotonically non-decreasing as the bound (budget) on the value of the other objective is decreased. This can be proved as follows. By decreasing the budget on the communication cost, the set of possible teams under the new budget becomes a subset of possible teams before the budget is decreased. Since the optimal team in a subset cannot be better than the optimal team in the superset, the personnel cost of the optimal team with the lower budget on the communication cost cannot be lower than the personnel cost of the optimal team with a higher budget on the communication cost.

Algorithm 2 $(n, \log n)$-approximation algorithm for solving $(sumDistance, PCost)$ problem

Input: graph G, project $P = \{s_1, s_2, \ldots, s_n\}$, and budget SD on the sum of distances.
Output: the best team and its personnel cost

```
for $i \leftarrow 1$ to $n$ do
    $C(s_i) \leftarrow$ the set of experts with $s_i$
$C \leftarrow \bigcup_{i=1}^{n} C(s_i)$
$bestTeam \leftarrow \emptyset$
$leastCost \leftarrow \infty$
for $h \leftarrow 1$ to $n$ do
    for each expert $cr_i$ in $C(s_h)$ do
        $requiredSkill \leftarrow P - Q(cr_i)$
        $V \leftarrow \emptyset$
        for each expert $c_j$ in $C$ do
            if $\langle d(cr_i, c_j) \leq \frac{SD}{n-1} \rangle$ & $\langle Q(c_j) \cap requiredSkill \neq \emptyset \rangle$ then
                $V \leftarrow V \cup \{c_j\}$
        $\{c_1^v, \ldots, c_q^v\} \leftarrow V$
        $skillV \leftarrow \bigcup_{i=1}^{q} Q(c_i^v)$
        if $requiredSkill \subseteq skillV$ then
            $team \leftarrow \{\langle q_1, cr_i \rangle, \langle q_2, cr_i \rangle, \ldots, \langle q_k, cr_i \rangle\}$ where $q_1, q_2, \ldots q_k$ are the required skills that $cr_i$ has, i.e., $\{q_1, q_2, \ldots q_k\} = P \cap Q(cr_i)$
            $cost \leftarrow t(cr_i)$
            while $requiredSkill \neq \emptyset$ do
                Select $k$ s.t. $\frac{|requiredSkill \cap Q(c_k^v)|}{t(c_k^v)}$ is maximized
                $team \leftarrow team \cup \{\langle q_1, c_k^v \rangle, \langle q_2, c_k^v \rangle, \ldots, \langle q_k, c_k^v \rangle\}$ where $\{q_1, q_2, \ldots q_k\} = requiredSkill \cap Q(c_k^v)$
                $cost \leftarrow cost + t(c_k^v)$
                $requiredSkill \leftarrow requiredSkill - Q(c_k^v)$
            if $cost < leastCost$ then
                $bestTeam \leftarrow team$
                $leastCost \leftarrow cost$
            else
                if $cost = leastCost$ & $team.sumDistance < bestTeam.sumDistance$ then
                    $bestTeam \leftarrow team$
return $bestTeam, leastCost$
```

In [15], a generic procedure was proposed that uses an (α, β)-approximation algorithm for the (A, B) problem to solve the (B, A) problem in polynomial time and with the approximation ratio of (β, α). The procedure applies to only hostile bicriteria problems. Since the two criteria in our problem are hostile and the algorithms we proposed in the last subsection for the ($diameter$, $PCost$) and ($sumDistance$, $PCost$) problems are (α, β)-approximation algorithms, we can adapt the generic procedure in [15] to derive (β, α)-approximation algorithms to solve the $(PCost, diameter)$ and $(PCost, sumDistance)$, respectively. Note that this is the first time that this generic procedure is adapted to find teams of experts.

The $(\log n, 2)$-algorithm for solving the

[3] Note that if the distance is set to a value smaller than $\frac{SD}{n-1}$, some of the teams with $sumDistance$ less than or equal to SD might be missed. On the other hand, if it is set to a value larger than $\frac{SD}{n-1}$, the approximation ratio is increased.

Algorithm 3 $(\log n, 2)$-approximation algorithm for solving $(PCost, diameter)$ problem

```
Input: graph $G$, project $P = \{s_1, s_2, \ldots, s_n\}$, budget $B$ on $PCost$, and precision threshold $\epsilon$.
Output: the best team and its diameter
$MaxDiameter \leftarrow \max dist(c_i, c_j), 1 \le i, j \le m$ and $m$ is the number of nodes in $G$
$D_{prev} \leftarrow MaxDiameter$
$\langle team_{prev}, PCost_{prev} \rangle \leftarrow$ Algorithm1($G$, $P$, $D_{prev}$)
if $PCost_{prev} > B$ then
    return $\emptyset, \infty$
$D_{lower} \leftarrow 0$
while $(D_{prev} - D_{lower}) > \epsilon$ do
    $D_{new} \leftarrow \frac{D_{prev} + D_{lower}}{2}$
    $\langle team_{new}, PCost_{new} \rangle \leftarrow$ Algorithm1($G$, $P$, $D_{new}$)
    if $team_{new} \neq \emptyset$ and $PCost_{new} \le B$ then
        $D_{prev} \leftarrow D_{new}$
        $\langle team_{prev}, PCost_{prev} \rangle \leftarrow \langle team_{new}, PCost_{new} \rangle$
    else
        $D_{lower} \leftarrow D_{new}$
return $team_{prev}, D_{prev}$
```

$(PCost, diameter)$ problem is presented in Algorithm 3. The basic idea of the algorithm is to conduct a **binary search** over the range of diameter values for a diameter value that is as small as possible and at the same time the $PCost$ value of the team is not over the budget B. The algorithm starts with the diameter of the input graph G, and stores it in D_{prev}. It calls Algorithm 1 with D_{prev} as the diameter budget to find the best (approximate) team that minimizes $PCost$. If the $PCost$ value of the found team is greater than the input budget B on $PCost$, no solution exists because if the diameter is lowered, the minimal $PCost$ value will not decrease (due to the hostile relationship between the two objectives). But if the $PCost$ value of the team found by Algorithm 1 is less than B, then there may exist teams with lower diameters and also under the $PCost$ budget. Thus, the algorithm continues and checks the diameter which is half of the previous value in D_{prev} by calling Algorithm 1 with this new diameter value (stored in D_{new}) as the diameter budget. If the $PCost$ value of the answer returned by Algorithm 1 is more than B, there is no solution that has a diameter under or equal to D_{new} (due to the hostile relationship between diameter and $PCost$). The algorithm then increases the value in D_{new} to $(D_{prev} + D_{new})/2$ and calls Algorithm 1 again with the new value in D_{new} to continue the search. However, if the $PCost$ value of the answer returned by Algorithm 1 is less than B, there may exist solutions with lower diameter values and thus the algorithm decreases the value in D_{new} by half and continues the binary search. In each iteration, the optimal diameter lies between D_{prev} and D_{lower}, which are the upper and lower boundaries of the current search range, and D_{new} is the middle value between D_{prev} and D_{lower}. The boundaries are adjusted according to whether the $PCost$ value of the team returned by Algorithm 1 is greater than B or not. Thus, when the search range gets smaller, we get closer to the team with the minimum diameter under the $PCost$ budget. The process stops when the difference between D_{prev} and D_{lower} is smaller than an input precision threshold, and it outputs the last valid team returned by Algorithm 1, stored in $team_{prev}$.

The maximum number of iterations of Algorithm 3 is $\log_2 \frac{MaxDiameter}{\epsilon} + 1$. Thus, the time complexity of Algorithm 3 in the worst case is $O(m^2 \times n \times (\log_2 \frac{MaxDiameter}{\epsilon} + 1))$, where $O(m^2 \times n)$ is the worst case complexity of Algorithm 1. Since $MaxDiameter$ is the largest shortest distance between any two nodes in the input graph G, which is at most m times the maximum edge weight on the shortest path (where m is the number of nodes in G), the algorithm is polynomial in terms of input data.

THEOREM 3.3. *Algorithm 3 is a $(\log n, 2)$-approximation algorithm for solving the $(PCost, diameter)$ problem where n is the number of required skills in the project.*

Proof. If G does not contain a B-bounded team, the algorithm returns $(\emptyset, \infty)$ and terminates. Now, assume that G contains a B-bounded team. Let the minimum (optimal) diameter of a B-bounded team in G be Opt_D. Let $Approx_D$ and $Approx_B$ specify the $diameter$ and $PCost$ values returned by Algorithm 3 respectively. Due to the hostility of the objectives, during any iteration of the binary search, we have $D_{lower} \le Opt_D \le D_{prev}$. Let D_{lower}^{final} and D_{prev}^{final} denote the final values of D_{lower} and D_{prev} respectively. We have $D_{prev}^{final} \le D_{lower}^{final} + \epsilon$. Thus, $D_{prev}^{final} \le Opt_D + \epsilon$. It is not difficult to see that the team returned by the algorithm is generated by Algorithm 1 with D_{prev}^{final} as the diameter budget. Thus, according to the approximation ratio of Algorithm 1, $Approx_D \le 2D_{prev}^{final} \le 2(Opt_D + \epsilon)$. Since ϵ is a small precision threshold, it can be ignored. Thus, Algorithm 3 returns a team from G whose diameter is at most 2 times that of the optimal B-bounded team. Let Opt_B denote the optimal $PCost$ value under the diameter budget D_{prep}^{final}, according to the approximation ratio of Algorithm 1, $Approx_B \le \log nOpt_B$. Since $Opt_B \le B$, we have $Approx_B \le \log nB$.

Since the general structure of Algorithm 3 is generic, it can be changed to solve $(PCost, sumDistance)$ problem by calling the appropriate algorithm at the places where Algorithm 1 is called.

4 Finding Pareto-optimal Teams

The algorithms above allow the user to provide a budget on one objective and finds the best solution on the other objective under the budget. Sometimes, the user may not want to specify budgets, but prefer to see all the optimal choices in the two-objective space so that he/she can select a solution that best fits his/her preferences. To this end, in this section we propose an algorithm that produces a set of optimal solutions that are not dominated by others. Below

Algorithm 4 An approximation algorithm for finding Pareto Set of Team of Experts minimizing $diameter$ and $PCost$.

```
Input: graph $G$, project $P = \{s_1, s_2, \ldots, s_n\}$, and precision threshold $\epsilon$.
Output: $ParetoSet$
 1: $MaxDiameter \leftarrow \max dist(c_i, c_j), 1 \leq i, j \leq m$ and $m$ is the number
    of nodes in $G$
 2: $PT = \emptyset$ /* for storing generated teams */
 3: $Diameter \leftarrow MaxDiameter$
 4: while $Diameter \geq 0$ do
 5:     $\langle team, cost \rangle \leftarrow$Algorithm1($G$, $P$, $Diameter$)
 6:     $flag = 0$ /* for indicating whether $t$ is dominated */
 7:     if $team \neq \emptyset$ then
 8:         if Algorithm 1 is an approximation algorithm then
 9:             for each $t$ in $PT$ do
10:                 if $t \prec team$ then
11:                     $flag = 1$
12:                     break the for loop
13:                 else
14:                     if $team \prec t$ then
15:                         remove $t$ from $PT$
16:         if $flag = 0$ then
17:             insert $team$ into $PT$
18:     else
19:         return $PT$
20:     $Diameter \leftarrow Diameter - \epsilon$
21: return $PT$
```

we define the relevant concepts , present the algorithm and prove the bounds of the solutions produced by the algorithm.

DEFINITION 4.1. ***(Dominance)*** *A team T dominates a team T' (denoted by $T \prec T'$) with respect to the communication and personnel costs if T is better than T' in one objective and not worse than T' in the other objective.*

DEFINITION 4.2. ***(Pareto-optimal team)*** *Given a project P, a team T is a Pareto-optimal team for project P if there does not exist a team T' that contains all the skills required by P such that $T' \prec T$.*

The set of all Pareto-optimal teams for project P is called the **Pareto set** of P. The teams in a Pareto set usually forms a convex curve (called **Pareto curve**) in the two-objective space.

A popular approach for finding Pareto-optimal solutions for multi-objective problems in the literature is to use an evolutionary algorithm, which is a heuristic method that mimics the process of natural evolution. A problem with such a method is that there is no provable bound for the approximation ratio. Here we propose a new general procedure that makes use of the (α, β) approximation algorithms that we proposed in the last section to find a set of (approximate) Pareto-optimal solutions with performance bounds.

The algorithm for producing (approximate) Pareto-optimal answers based on $diameter$ and $PCost$ is presented in Algorithm 4. It repeatedly calls Algorithm 1 with a set of diameter budgets, starting from the diameter value of the input graph and decrementally changing the budget value by ϵ, which is an input precision threshold. In this way, a set of teams is generated each of which minimizes the personnel cost ($PCost$) under a diameter budget. If Algorithm 1 is an exact algorithm, the generated teams are guaranteed to be Pareto-optimal (See the proof of Theorem 4.1 below). If Algorithm 1 is an approximation algorithm (such as the Algorithm 1 proposed in Section 3), Algorithm 4 checks whether a newly-generated team is dominated by (or dominates) a previously-generated team . If it is dominated by a generated team, it is ignored. If it dominates a generated team, the generated team is removed and the new team is added to the set of Pareto teams. The worst case time complexity of Algorithm 4 is $O(\frac{MaxDiameter}{\epsilon} \times (m^2 \times n + \frac{MaxDiameter}{\epsilon}))$ where $m^2 \times n$ is the worst time taken by Algorithm 1.

THEOREM 4.1. *Algorithm 4 produces Pareto-optimal teams if Algorithm 1 in line 5 returns an exact answer.*

Proof. Assume that T is a team returned by Algorithm 4 and that there exists a team Q that dominates T. Q dominates T if and only if one of the three conditions holds. We show none of these conditions can occur.

(1) $PCost(Q) < PCost(T)$ and $diameter(Q) < diameter(T)$. Since Q is better than T in both objectives, if Algorithm 1 is an exact algorithm, T would not be chosen under any diameter budget.

(2) $PCost(Q) = PCost(T)$ and $diameter(Q) < diameter(T)$. This is not possible since for teams with the same personnel cost, Algorithm 1 chooses the one with the least diameter.

(3) $PCost(Q) < PCost(T)$ and $diameter(Q) = diameter(T)$ T would not be chosen by Algorithm 1 if Algorithm 1 is an exact algorithm since Q is better than T in the personnel cost under any diameter budget no less than diameter(T).

Since team Q does not exist, team T is Pareto-optimal.

The following theorem states how close a team generated by Algorithm 4 is to a Pareto-optimal team in the worse case if Algorithm 1 is the approximation algorithm as presented in Section 3.

THEOREM 4.2. *For each team s' produced by Algorithm 4, there exists a team s in the Pareto set such that $diameter(s') \leq 2 \times diameter(s)$ and $PCost(s') \leq \log n \times PCost(s)$.*

Proof. This can be easily derived from Theorems 3.1 and 4.1.

The following theorem states how well the teams in the Pareto Set are represented by the teams produced by Algorithm 4.

THEOREM 4.3. *For each team s in the Pareto set, there exists a team s' produced by Algorithm 4 such that $diameter(s') \leq 2 \times (\epsilon + diameter(s))$ and $PCost(s') \leq \log n \times PCost(s)$, where ϵ is the input precision threshold of Algorithm 4.*

Proof. For any team s in the Pareto set, there is a team r in the Pareto set that is generated by Algorithm 4 if Algorithm 1 were an exact algorithm such that $diameter(r) \leq \epsilon + diameter(s)$, because the interval of the diameter budgets used in Algorithm 4 to call Algorithm 1 is ϵ. Let s' be the team generated by Algorithm 4 when Algorithm 1 is the approximation algorithm under the diameter budget that r were generated with the exact Algorithm 1. According to Theorem 3.1, we have $diameter(s') \leq 2 \times diameter(r)$. Thus, $diameter(s') \leq 2 \times (\epsilon + diameter(s))$.

For $PCost$, we have $PCost(s') \leq \log n \times PCost(r)$. Also, since r is on the Pareto curve, $PCost(r) \leq PCost(s)$. Thus, we have $PCost(s') \leq \log n \times PCost(s)$.

To find the Pareto optimal solutions for minimizing $sumDistance$ and $PCost$, the appropriate algorithms can be used in Algorithm 4 at the places where Algorithm 1 is called. The corresponding approximation bounds can be derived similarly.

5 Experimental Evaluation

5.1 The Datasets and Experimental Setup The DBLP and IMDb data sets are used in the experiments. The DBLP graph is produced from the DBLP XML data[4] taken on April 25, 2011. The dataset contains information about a collection of papers and their authors. The set of experts and skills are generated in the same way as in [12, 10] as follows. For each paper, the name(s) of the author(s), the conference where it was published and the title of the paper are specified. We only keep the papers of some major conferences in computer science: SIGMOD, VLDB, ICDE, ICDT, EDBT, PODS KDD, WWW , SDM, PKDD, ICDM, ICML, ECML, COLT, UAI, SODA, FOCS, STOC, and STACS. The set of experts consists of authors with at least 3 papers in the DBLP. The skills of an expert is the set of terms that appear in the titles of at least two papers of the expert. Two experts are connected together if they have at least two papers together. The weight of the edge between two nodes n_i and n_j is equal to $1 - |\frac{p_{n_i} \cap p_{n_j}}{p_{n_i} \cup p_{n_j}}|$ where p_{n_i} is the set of papers of author n_i. The cost of an expert in the DBLP is set to be the number of publications of the expert, assuming that the more publications an expert has, the more expensive he/she is. The final graph has 6,229 nodes and 9,400 edges.

The part of the IMDb dataset[5] used in our experiments contains information about the actors and the list of movies that each actor played in. It is processed in the same way as described in [12, 10]. The expert cost in IMDb is defined as the number of movies the actor plays in. The IMDb graph has 6,784 nodes and 35,875 edges. Due to space limit, more details on this data set are omitted. In our experiments, the DBLP and IMDb datasets show similar trends. Thus, most of the results presented below are from DBLP to save space.

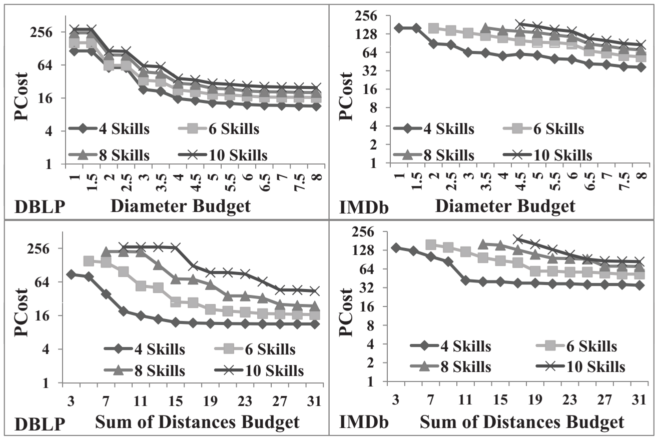

Figure 3: The personnel cost (shown in logarithmic scale) produced by our algorithms that receive a budget on the communication cost on DBLP and IMDb datasets. For some budget values, no team exists in the graph.

The projects used in the experiments are generated as follows. Each project is determined by a set of skills. The number of skills in a project varies from 4 to 10. For each number of skills, 100 projects are generated randomly. The average result over 100 projects is computed and used as the result of each algorithm. Skills have different frequencies. The frequency of a skill is the percentage of experts in the expert network that possess the skill. The frequency of the skills in our experiments varies from 0.1 to 5.0 percent. Other values of skill frequency show similar trends. All the algorithms are implemented in Java. The experiments are conducted on an Intel(R) Core(TM) i7 2.80 GHz computer with 4 GB of RAM.

5.2 Results of Algorithms with Given Budget

5.2.1 Hostility between two objectives Figure 3 shows the $PCost$ values of the teams produced by our algorithms that receives budget on the communication cost for different budgets on $diameter$ or $sumDistance$. Since the two objectives are hostile, by increasing the communication budget, the personnel cost decreases. For teams with the same budget, the more the required skills, the higher personnel costs. The results also show that our algorithms are able to find teams with both small personnel cost and small communication cost. For example, for 4 required skills, our Algorithm 1 is able to find a team with a $PCost$ value of 16 and within a diameter budget of 4. Such a team cannot be found by the single objective methods that minimize either communication cost or personnel cost

[4] http://dblp.uni-trier.de/xml/

[5] http://www.imdb.com/interfaces

5.2.2 Quality of Approximation Algorithms We compare our approximation algorithms with the exact algorithms in terms of the quality of the answers. The answers of the exact algorithms are obtained using exhaustive search. Figure 4 shows the communication and personnel costs of the teams produced by the exact algorithms and Algorithms 1 and 2 for different budget values on diameter or sum of distances for projects with four skills. Due to the poor performance and long run time of the exhaustive search, the results of higher number of skills and higher communication cost budgets are not presented. The results show that the costs of the teams produced by our approximation algorithms are very close to those produced by the exact algorithms. The ratio of the diameter from Algorithm 1 or sum of distances from Algorithm 2 to the one from the exact algorithm is at most 1.29 or 1.68 respectively, although the theoretical bound for the approximation ratio is 2 (as shown in Theorem 3.1) or 4 (which is the number of required skills as shown in Theorem 3.2). This means that our approximation algorithms perform very well in practice, much better than the worse case scenario. The results also show that the $PCost$ values of the teams produced by Algorithms 1 and 2 are sometimes slightly smaller than the ones from the exact algorithm. This seems a surprise. However, the reason is that some of the teams returned by the approximation algorithms have larger diameter/sum of distances than the budget. These teams are not considered by the exact algorithm. Therefore, they might have smaller personnel cost than the teams that actually lie within the communication budget. Note that these results do not violate the $(2, \log n)$ approximation ratio of Algorithm 1. The personnel cost of the approximation algorithm is at most $\log n$ times of the personnel cost of the exact answer. In this case, it is even smaller than the cost of the exact answer. Due to the space limit, only the results of Algorithms 1 and 2 are presented. Other approximation algorithms have similar performance.

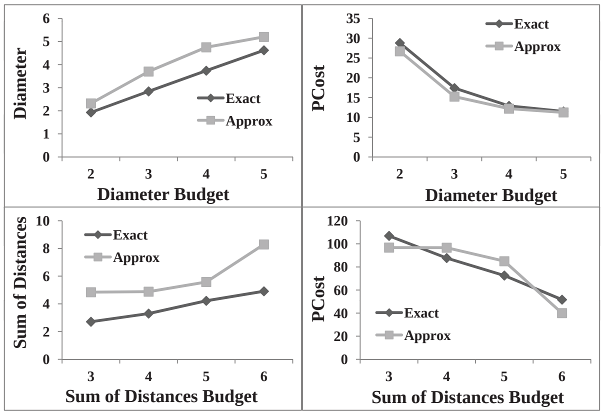

Figure 4: The costs of the teams from exact algorithms and Algorithms 1 and 2 on DBLP for projects with 4 skills.

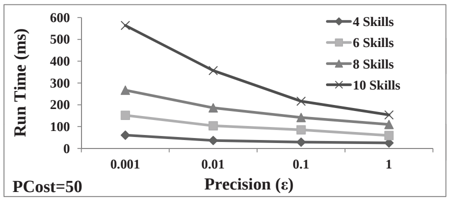

Figure 5: The run time of Algorithm 3 for different values of ϵ on DBLP dataset.

5.2.3 Precision vs. Run Time As discussed before, the value of ϵ in Algorithm 3 determines the precision of the output teams. However, by increasing the precision (i.e., decreasing the value of ϵ), the run time increases. Figure 5 shows how the run time of Algorithm 3 changes with the ϵ value for different numbers of required skills on the DBLP dataset. As expected, by decreasing the value of ϵ, the run time increases close to linearly. It is because the run time is logarithmically related to the ratio of the diameter of the graph G to ϵ.

5.3 Results of the Pareto Set Algorithm In this section the effectiveness and efficiency of the proposed method for finding Pareto solutions are evaluated. The proposed method (Algorithm 4) is referred to as *Approx-Pareto*. To the best of our knowledge, there does not exist a Pareto optimization method for team formation. However, we implemented the following methods to compare with *Approx-Pareto*: (1) ***Exact-Pareto***: The exact Pareto set is found using exhaustive search. (2) ***Random-Pareto***: This method randomly selects a set of connected teams (1% of total teams), and then removes the teams that dominated by other generated teams. (3) ***GA-Pareto*** [8]: We apply a genetic algorithm for finding Pareto solutions proposed in [8] to our team formation problem. All the parameters are set in the same way as in [8].

We use the following performance measures: (1) ***Hypervolume*** (HV) [21]: It measures (in percentage) the volume of the dominated space by a generated Pareto set within search space composed by bounds of objective values. The higher the value, the better the Pareto set. (2) ***Average Distance*** (D_{avg}) and ***Maximal Distance*** (D_{max}) [4]: Given a true Pareto set R and a set S of approximate Pareto teams, D_{avg} is the average distance from each $y \in R$ to the closest team in S and D_{max} is the maximum distance between them. For both measures, lower values are preferred. (3) ***Precision*** and ***Recall***: $Precision = \frac{|True\ ParetoSet \bigcap Retrieved\ ParetoSet|}{|Retrieved\ ParetoSet|}$, $Recall = \frac{|True\ ParetoSet \bigcap Retrieved\ ParetoSet|}{|True\ ParetoSet|}$, where $|\cdot|$ denotes set cardinality. (4) ***Run time***. The result of the Exact-Pareto method is used as the true Pareto set for cal-

Table 1: Results of algorithms for finding Pareto set (For *Approx-Pareto*, ϵ is set to 0.1).

# Skill	Method	HV(%)	D_{max}	D_{avg}	Precision(%)	Recall (%)	Time (ms)
3	Exact-Pareto	**47.6**	**0**	**0**	**100**	**100**	10,563
3	Approx-Pareto	**42.6**	**4.31**	**0.89**	**64**	**34.5**	498
3	GA-Pareto	29.8	150.81	39.60	3	1.1	353
3	Random-Pareto	30	102.83	18.23	7.1	3.6	**104**
4	Exact-Pareto	**45.8**	**0**	**0**	**100**	**100**	42,376
4	Approx-Pareto	**40.2**	**12.5**	**1.44**	**70.4**	**28.23**	647
4	GA-Pareto	34	58	10.06	4.4	2.9	921
4	Random-Pareto	37	90.8	15.87	2	0.87	**242**
5	Exact-Pareto	**55.3**	**0**	**0**	**100**	**100**	92,235
5	Approx-Pareto	**50.87**	**17**	**3.47**	**21.43**	**13.64**	968
5	GA-Pareto	50.14	28.01	8.37	0	0	2030
5	Random-Pareto	49.35	140	13.22	6.67	4.55	**496**

culating H, D_{avg} and D_{max} indicators.

Table 1 shows the results of the algorithms for different numbers of required skills. The overall best results and best results among non-exact methods are highlighted in bold. Not surprisingly, *Exact-Pareto* gives the best or perfect results on all the quality measures (HV, D_{avg}, D_{max}. $Precision$ and $Recall$). However, its run time is orders of magnitude longer than those of the three non-exact methods. This indicates the need for non-exact algorithms. The results also indicate that *Approx-Pareto* significantly outperforms the other non-exact methods in terms of all the quality measures (HV, D_{avg}, D_{max}. $Precision$ and $Recall$). Its HV values are close to those of the exact method. In run time, the random method is the fastest as expected. Compared to the GA method, *Approx-Pareto* is slower than *GA-Pareto* when the number of required skills is 3, but is much faster than GA when the number of skills becomes a bit bigger. It is because by increasing the number of required skills from 3 to 4 or 5, the search space expansion of *GA-Pareto* is much more than *Approx-Pareto*.

6 Related Work

Discovering a team of experts in a social network is introduced in [12], in which two communication cost functions are proposed. Authors of [13] generalize this problem by associating each required skill with a specific number of experts, but no approximation ratio is provided for the algorithms. The authors of [10] propose the *sum of distances* communication function and a 2-approximation algorithm for minimizing the *sum of distances*. They also introduce the problem of finding a team of experts with a leader. The authors of [6] propose another communication cost function based on the density of the induced subgraph on selected nodes. They also reported improvements over [12]. Authors of [1] minimize the maximum load of the experts in the presence of several tasks. They do not consider finding teams with low communication cost. Recently, the problem of online team formation is studied in [2], which creates teams of experts with minimized work load and communication cost. Balancing the work load while minimizing the communication cost is also studied in [5]. The personnel cost of the experts is not considered in [2, 5]. In [11], the authors propose to find a team of experts while minimizing both communication and personnel cost. They merged the two objective functions into one function using an input threshold from the user. In this work, we solve the problem using two fundamentally different approaches, finding the solutions within the given budget and finding the Pareto front.

Another line of research in the database community related to finding Pareto sets is the skyline computation [3, 17]. A skyline of a set of objects (i.e. records) contains all the records that are not dominated by any other record, which is the same as a Pareto set. However, in skyline computation, the set of records from which a skyline is found is given in the database. Assuming n is the number of records, a naive algorithm is able to compute the skyline in $O(n^2)$ [3]. The main purpose of the skyline algorithms is to reduce this complexity. In contrast, the possible teams in our work is not given and our algorithms have to walk through a search space to find the (approximate) best or Pareto-optimal teams. The number of possible teams is exponential with respect to the number of required skills. Thus, it is not feasible to produce all of the teams and then find the Pareto set from it (i.e. run a skyline algorithm on all of the teams).

The mechanisms for creating the structure of the collaboration networks in the self assemble teams are studied in [7]. The proposed model of the self assembly teams is based on the following three parameters: the size of the team, the fraction of the newcomers and the rate of repeating previous collaboration. The authors suggest that team assembly mechanisms determine collaboration network structure and team performance.

7 Conclusion

We studied the problem of finding an affordable and collaborative team from an expert network that minimizes two objectives: the communication cost among team members and the personnel cost. We proved that the problem we tackle is NP-hard. Two functions are used to measure the communication cost of a team and another function is proposed to evaluate the personnel cost of the team. A suite of algorithms classified into two approaches are proposed to solve this bi-criteria problem. In the first approach, a budget is given on one objective and the purpose is to minimize the other objective under the budget. The budget could be either on the communication cost or the personnel cost. In the second approach, a set of approximate Pareto-optimal solutions are generated in which there exists no other team that dominates the solution in both of the costs. All of the proposed algorithms have provable approximation bounds. We evaluated the proposed algorithms on the DBLP and IMDb datasets and showed that our proposed algorithms are effective and efficient.

References

[1] A. Anagnostopoulos, L. Becchetti, C. Castillo, A. Gionis, and S. Leonardi. Power in unity: Forming teams in large-scale community systems. In *Proc. of CIKM'10*, 2010.

[2] A. Anagnostopoulos, L. Becchetti, C. Castillo, A. Gionis, and S. Leonardi. Online team formation in social networks. In *Proc. of the WWW'12*, 2012.

[3] S. Borzsony, D. Kossmann, and K. Stocker. The skyline operator. In *Proc. of ICDE'01*, 2001.

[4] P. Czyzzak and A. Jaszkiewicz. Pareto simulated annealing a metaheuristic technique for multiple objective combinatorial optimization. *Journal of Multi Criteria Decision Analysis*, 7:34–47, 1998.

[5] S. Datta, A. Majumder, and K. Naidu. Capacitated team formation problem on social networks. In *Proc. of KDD'12*, 2012.

[6] A. Gajewar and A. D. Sarma. Multi skill collaborative teams based on densest subgraphs. In *Proc. of the SDM'12*, 2012.

[7] R. Guimera, B. Uzzi, J. Spiro, and L. A. N. Amaral. Team assembly mechanisms determine collaboration network structure and team performance. *Science*, 308:697–702, 2005.

[8] J. Horn, N. Nafpliotis, and D. E. Goldberg. A niched pareto genetic algorithm for multiobjective optimization. In *Evolutionary Computation, 1994. IEEE World Congress on Computational Intelligence., Proceedings of the First IEEE Conference on*, 1994.

[9] D. Kalyanmoy. Multi-objective optimization. In *Search Methodologies*, pages 273–316. Springer US, 2005.

[10] M. Kargar and A. An. Discovering top-k teams of experts with/without a leader in social networks. In *Proc. of CIKM'11*, 2011.

[11] M. Kargar, A. An, and M. Zihayat. Efficient bi-objective team formation in social networks. In *Proc. of ECML-PKDD'12*, 2012.

[12] T. Lappas, L. Liu, and E. Terzi. Finding a team of experts in social networks. In *Proc. of KDD'09*, 2009.

[13] C. Li and M. Shan. Team formation for generalized tasks in expertise social networks. In *Proc. of IEEE International Conference on Social Computing*, 2010.

[14] G. Li, B. C. Ooi, J. Feng, J. Wang, and L. Zhou. Ease: Efficient and adaptive keyword search on unstructured, semi-structured and structured data. In *Proc. of SIGMOD'08*, 2008.

[15] M. V. Marathe, R. Ravi, R. Sundaram, S. Ravi, D. J. Rosenkrantz, and H. B. Hunt. Bicriteria network design problems. *Journal of Algorithms*, 28(5):142–171, 1998.

[16] M. Newman. The structure of scientific collaboration networks. In *Proc. of the National Academy of Sciences*, 2001.

[17] D. Papadias, Y. Tao, G. Fu, and B. Seeger. Progressive skyline computation in database systems. *ACM Transactions on Database Systems*, 30:41–82, 2005.

[18] C. H. Papadimitriou and M. Yannakakis. On the approximability of trade-offs and optimal access of web sources. In *Proc. of FOCS'00*, 2000.

[19] L. Qin, J. Yu, L. Chang, and Y. Tao. Querying communities in relational databases. In *Proc. of ICDE'09*, 2009.

[20] V. Vazirani. *Approximation Algorithms*. Springer, 2001.

[21] E. Zitzler. *Evolutionary Algorithms for Multiobjective Optimization: Methodsand Applications*. PhD thesis, ETH Zurich, Switzerland, 1999.