

Team Formation with the Communication Load Constraint in Social Networks

Yui-Chieh Teng, Jun-Zhe Wang, and Jiun-Long Huang^(✉)

Department of Computer Science, National Chiao Tung University,
Hsinchu City, Taiwan, ROC
yuichey@gmail.com, {jzwang, jlhuang}@cs.nctu.edu.tw

Abstract. Given a project requiring a set of skills, the team formation problem in social networks aims to find a team that can cover all the required skills and has the minimal communication cost. Previous studies considered the team formation problem with a leader and proposed efficient algorithms to address the problem. However, for large projects, a single leader is not capable of managing a team with a large number of team members. Thus, a number of leaders would be formed and organized into a hierarchy where each leader is responsible for only a limited number of team members. In this paper, we propose the team formation problem with the communication load constraint in social networks. The communication load constraint limits the number of team members a leader communicates with. To solve the problem, we design a two-phase framework. Based on the proposed framework, we first propose algorithm Opt to find an optimal team, under the communication load constraint, with minimal communication cost. For large social networks, we also propose algorithm Approx to find a nearly-optimal team. Experimental results show that algorithm Opt is able to find optimal teams and is more efficient than the brute-force algorithm. In addition, when nearly-optimal teams are acceptable, algorithm Approx is much more scalable than algorithm Opt for large social networks.

Keywords: Team formation · Degree-constrained minimum spanning tree · Social network

1 Introduction

To carry out a project with various required skills, it is usually necessary to form a team of experts that are able to cover the skill set. The fulfilment of the required skills for a project is fundamental to accomplish the project. However, a more important key to the success of the project is whether the experts in the team can effectively communicate and collaborate with each other. Thus, given a project, it is desirable to organize a team of experts possessing all the skills required for the project and with minimal communication cost.

Lappas et al. [9] were the first to consider the communication factor of team formation in social networks. They presented two communication cost metrics,

namely *diameter communication cost* and *minimum spanning tree cost*, to evaluate communication effectiveness of a team. Kargar and An [8] proposed to use the *sum of distances*, a more stable cost metrics, to define the communication cost of a team. They also considered the team formation problem with a leader where the leader is responsible for coordinating all team members and each team member directly communicates with the leader. To measure the communication cost of a team with a leader, they introduced *leader distance* which is the sum of the shortest distances between the leader and the corresponding team member for each required skill. A brute-force algorithm was developed to identify the best leader and the corresponding team. Juang et al. [7] proposed two algorithms, called algorithm BCP pruning and algorithm SSP pruning, to accelerate the discovery of the best leader and team. These two algorithms were shown to be more efficient and scalable than the algorithm proposed in [8].

Although employing a good leader managing and coordinating team members is beneficial, a single leader is not sufficiently capable of administering a large project requiring a number of experts since the leader may not have enough time to communicate with all team members. For a large project in the real world, the experts usually would be divided into different groups based on the tasks of the project and the skills of the experts. As such, instead of a single leader, a number of leaders are employed to co-work for a large project where (1) these leaders would be organized into a hierarchical structure (i.e., a tree with height larger than two) and (2) any leader communicates with no more than d leaders at the next low level or team members. d is referred to as the *communication load constraint*. In other words, finding the team and hierarchy with minimal communication cost under the communication load constraint is crucial for a large project in practice. Such characteristic distinguishes our paper from others.

In view of this, we propose in this paper the team formation problem with the communication load constraint in social networks. Specifically, the proposed problem is to discover the team and hierarchy that covers the required skills, has the minimal communication cost and guarantees each leader to have the predefined, acceptable load. To facilitate the process of identifying the desired team and hierarchy, we present a two-phase framework consisting of the team generation phase and the hierarchy establishment phase. The team generation phase is to find all the teams *qualified* for a given project (i.e., the team members possess all the required skills). In the hierarchy establishment, the hierarchy with the minimal communication cost of each team is created and the team and hierarchy having the minimal communication cost is then determined. Note that the hierarchy of a team meeting the load constraint and incurring the minimal communication cost is recognized as the *degree-constrained minimum spanning tree*, a well-known NP-complete problem. To solve the team formation problem with the communication load constraint, we first develop a naive algorithm, called algorithm Brute-Force, to find the best team and hierarchy by enumerating all the eligible teams and hierarchies. By analysing the breakdown of the execution time of algorithm Brute-Force, we observe that the hierarchy establishment phase accounts for the execution time since finding the minimal spanning trees

with the load constraint of all the qualified teams is very time-consuming. Thus, we design algorithm Opt to employ the lower bounds of the communication cost to prune some qualified teams in the first phase, thereby reducing the number of teams evaluated in the second phase. Although algorithm Opt is more efficient than algorithm Brute-Force, algorithm Opt suffers from heavy computation in large social networks. Since nearly-optimal solutions have been widely acceptable in many situations, we devise algorithm Approx to find nearly-optimal teams by applying the 2-opt change approximation [10] to construct nearly-minimum spanning trees with degree constraint. Experimental results show that algorithm Opt outperforms algorithm Brute-Force in terms of the execution time. Moreover, algorithm Approx is much more scalable than algorithm Opt at a cost of small communication cost increase (less than 10% in our experiments).

The remainder of this paper is organized as follows. Section 2 presents a survey of related work in the literature of team formation problem, followed by the formulation of the team formation problem under the communication load constraint. We then describe algorithm Brute-Force, algorithm Opt and algorithm Approx in Sect. 3 in detail. The experimental results are reported in Sect. 4. Finally, Sect. 5 concludes this paper.

2 Preliminaries

2.1 Related Work

Team Formation Problem in Operation Theory. The formation of the multi-functional teams, which are the teams required a set of skills to be handled, is crucial and has gained attention in recent years. Zakarian and Kusiak combined the different factors of the team selection problem into a hierarchical structure [11]. They also provided the quality functional development method and used the integer linear program (ILP) to model the problem. Fitzpatrick and Askin developed and tested mathematical models for formation of effective human teams based on the Kolbe Conative Index and presented a programming formulation for the problem [4]. Chen and Lin proposed a working relation model with respect to the multi-functional knowledge and the capability of each individual, and the co-work relation between members [3]. Different from the work in [3], Baykasoglu et al. used the fuzzy optimization model and the problem is solved by the annealing algorithm [2]. Gaston et al. considered about not only the impact of the network structures on the team performance but the network structure between individual [6].

Team Formation Problem in Social Networks. Lappas et al. [9] first addressed the problem with considering the individuals of social networks. They defined each node in a social network to represent an expert having a set of skills and the communication cost between two experts is the weight of the edge connecting them in the social network. If no such edge exists, the distance of the shortest path between these two experts is used to measure the communication cost between them. The communication cost indicates how effectively

the two experts collaborate. Thus, the smaller the communication cost is, the easier they can co-work. They proposed to use the *minimal spanning tree* and the *diameter* to measure the communication cost of a team. They also proved the NP-hardness of the two communication cost metrics for the team formation problem and proposed the approximation algorithms. In [8] Kargar and An argued that the communication cost metrics proposed in [9] were insensitive thus introduced a new communication cost metric: *sum of distance* to reveal all the skills needed to be communicated between any two experts. Furthermore, they also introduced the team formation problem with a leader. The leader is responsible for the team and is always coordinating with other team members. Kargar and An also proposed in [8] the *leader distance* to measure the communication cost of a team with a leader. In [7] Juang et al. proposed two efficient algorithms, algorithm BCPruning and algorithm SSPruning, for the team formation problem with a leader and these two algorithms outperform the previous work proposed in [8]. However, the team organizations in previous works, including [7, 8], are trees with height two, and the communication cost of a team lies on the communication responsibility of the leader. In practices, a leader can only directly communicated with limited team members, and thus, organizing a team into a hierarchy (a tree being able to with height larger than two) satisfying the communication cost constraint is necessary for large projects.

Team Formation Problem with Load Balance. In light of the fair work load for each member in the team, Anagnostopoulos et al. [1] considered the multi-project situation in team formation problem. The scenario in the work is to dispatch projects to people that can deal with more than one project at same time. But the load balance is considered and every expert has the upper bound of the number of projects involved. In addition to the fairest workload, they also find the team that has the minimal communication cost and thus making the problem more applicable in the real world. Nevertheless, the problem only probed into the multi-project allocation in on-line team formation and focused on minimizing the communication cost while having a fair work for every expert. In a real world, the communication between experts is as important as the technical expertise of each individual and thus the communication load of experts should not be neglected.

2.2 Problem Formulation

Given a group of experts $X = \{x_1, x_2, \dots, x_n\}$, the skill set composed of the skills which all the experts in X have is denoted by $S = \{s_1, s_2, \dots, s_m\}$. Let $skill(x)$ be the set of the skills that expert x possesses. We model an expert social network as an undirected graph $G(X, E)$ where each edge in E , say (x_i, x_j) , indicates that x_i and x_j had co-worked with each other before. In addition, the weight of (x_i, x_j) is the communication cost of x_i and x_j .

Definition 1. *Given a project $P \subseteq S$, a team $T \subseteq X$ is qualified for P if for each skill, say s in P , there exists one expert in T , say x , so that $s \in skill(x)$.*

Given a team T and an expert social network G , the communication graph is defined as below.

Definition 2. *The communication graph of T on G , denoted as $G_T(T, E_T)$, is a weighted and undirect complete graph, where the weight of each edge, say (x_i, x_j) , in E_T indicates the communication cost between experts x_i and x_j .*

If x_i and x_j had co-worked in the past, the edge (x_i, x_j) should exist in E and the weight of the edge (x_i, x_j) in G_T is set to the weight of the edge (x_i, x_j) in G . Otherwise, when the edge (x_i, x_j) is not in E , similar as the prior work [7, 8], the weight of the edge (x_i, x_j) in G_T is defined as the distance of the shortest path between x_i and x_j in G .

Definition 3. *Given a communication graph $G_T(T, E_T)$, the hierarchy of T under the communication load constraint d is a spanning tree where the degree of each node is at most d . In addition, the communication cost of the hierarchy is defined as the sum of the weights of the edges in the hierarchy.*

Definition 4. *Given an expert social graph $G(X, E)$ and a project P , the team formation problem with the communication load constraint d is to find a team T and the corresponding hierarchy with the minimal communication cost under the communication cost constraint.*

3 Proposed Algorithms

3.1 Algorithm Brute-Force

To solve the team formation problem with the communication load constraint, we propose a two-phase framework that comprises the team generation and hierarchy establishment phases below.

1. *Team generation phase:* Find all teams qualified for the given project.
2. *Hierarchy establishment phase:* Determine the team and hierarchy with the minimal communication cost under the communication load constraint by establishing the hierarchy with minimal communication cost for each team discovered in the team generation phase.

Based on the proposed two-phased framework, we develop a naive algorithm, called algorithm Brute-Force, to solve the team formation problem with the communication load constraint. Algorithm Brute-Force first use the method used in [7] to generate all the qualified teams in the team generation phase. Then, for each qualified team, algorithm Brute-Force generates the degree-constrained minimum spanning tree by enumeration. Note that the hierarchy establishment is the well-known NP-complete problem, the degree constrained minimum spanning tree problem on $G_T(T, E_T)$. Finally, algorithm Brute-Force takes the team and the spanning tree with minimum communication cost (hierarchy) as the answer to the team formation problem.

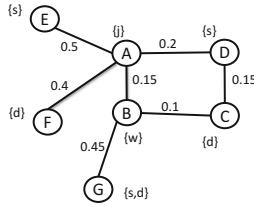


Fig. 1. A social network of experts

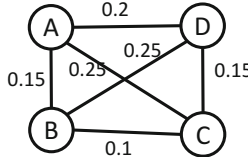


Fig. 2. The communication graph of team (A, B, C, D)

We use Fig. 1 and Table 1 as an example to illustrate the process of algorithm Brute-Force. Figure 1 shows an expert social network where a circle represents an expert and the letter(s) next to a circle indicate(s) the skills the expert possesses. The letter s, w, d and j mean the skills “software engineering,” “web programming,” “data mining” and “java,” respectively. The weight of an edge represents the communication cost between the two experts connected by the edge. Consider a project $P = \{j, s, d, w\}$ that requires four skills, namely java, software engineering, data mining and web programming. The communication load constraint is set to 2.

Table 1 shows the process of algorithm Brute-Force. In the team generation phase, five qualified teams ((A, B, C, D), (A, B, C, E), (A, B, F, D), (A, B, F, E) and (A, B, G)) are found. In team (A, B, C, D), A, B, C and D are responsible for java, software engineering, data mining and web programming, respectively. To calculate the minimum cost of team (A, B, C, D), algorithm Brute-Force first generates the communication graph of team (A, B, C, D), as shown in Fig. 2. Then, all hierarchies (i.e., degree-constrained spanning trees), under the communication load constraint, of the team are enumerated. After calculating the communication cost of each generated hierarchy, we can find the degree-constrained minimum spanning tree of the minimal communication cost $0.15 + 0.15 + 0.1 = 0.4$. The communication costs of other qualified teams are also calculated in a similar manner. Finally, algorithm Brute-Force reports that the best team is (A, B, C, D) of communication cost 0.4.

3.2 Algorithm Opt

Since enumerating all the hierarchies of each qualified team, algorithm Brute-Force obviously performs poorly in large social networks. In order to improve the

Table 1. The process of algorithm Brute-Force

Skills				Minimum communication cost
Java	Software engineering	Data mining	Web programming	
A	B	C	D	0.4
A	B	C	E	0.75
A	B	F	D	0.75
A	B	F	E	1.05
A	B	G	G	0.6

Table 2. Breakdown of execution time of algorithm Brute-Force (unit (ms))

	Number of required skills				
	3	4	5	6	7
Team generation	41594	57045	87750	102258	122345
Hierarchy establishment	5209834	7175670	10758594	12701720	3160379

performance, we conduct a preliminary experiment to observe the distributions of the execution time of algorithm Brute-Force. The social network consists of 30 nodes and 58 edges. The communication load constraint is set to 2 while the number of required skills ranges from 3 to 7. The experimental result is shown in Table 2. We can see that hierarchy establishment phase accounts for more than 99% of the total execution time. As the number of nodes in the social network increases, more teams will be generated in the team generation phase. More importantly, the execution time of the hierarchy establishment phase will increase significantly since the problem (i.e., degree-constrained minimum spanning tree problem) addressed in the hierarchy establishment is NP-complete [5]. Such an observation motivates us to proposed algorithm Opt to reduce the number of teams evaluated in the hierarchy establishment phase. Specifically, algorithm Opt takes advantage of the currently best communication cost to prune those qualified teams that cannot be of the minimal communication cost, thereby substantially saving the execution time of hierarchy establishment.

To avoid the high cost of finding the minimal-cost hierarchy (a degree-constrained minimum spanning tree) of a qualified team, we exploit the property of spanning trees that each node, except the root, contributes one edge to the spanning tree. In other words, there are $k - 1$ edges in a spanning tree formed by a team consisting of k experts. With the property, we introduce the following definitions.

Definition 5. Given a team T and the corresponding communication graph $G_T(T, E_T)$, the lower bound of the communication cost contributed by an expert x_i , denoted as $LB(x_i)$, is defined as

$$LB(x_i) = \min_{\forall (x_i, x_j) \in E_T} \text{weight of } (x_i, x_j).$$

Table 3. The process of algorithm Opt

Skills				LB(T)	Minimum communication cost
Java	Software engineering	Data mining	Web programming		
A	B	C	D		0.4
A	B	C	E	0.35	0.75
A	B	F	D	0.5	-
A	B	F	E	0.5	-
A	B	G		0.3	0.6

Based on Definition 5, we can derive the low bound of a team T on $G_T(T, E_T)$ below.

Definition 6. Given a team T and the corresponding communication graph $G_T(T, E_T)$, the low bound of the communication cost of T , denoted as $LB(T)$, can be formulated as

$$LB(T) = \sum_{\forall x_i \in T} LB(x_i) - \max_{\forall x_i \in T} LB(x_i).$$

We use Fig. 2 and Table 3 to illustrate the process of algorithm Opt. Similar to algorithm Brute-Force, algorithm Opt first generates all qualified teams and the qualified teams are listed in Table 3. In the hierarchy establishment phase, algorithm Opt first calculates the communication cost of team (A, B, C, D) by the method used in algorithm Brute-Force and obtains the communication cost of (A, B, C, D) to be 0.4. Algorithm Opt marks (A, B, C, D) as the candidate of the best team. Before proceeding to calculate the communication cost of team (A, B, C, E), algorithm Opt calculates the lower bound of the communication cost of team (A, B, C, E). Since the lower bound of the communication cost of team (A, B, C, E) is less than the communication cost of the candidate (A, B, C, D), algorithm Opt finds the hierarchy of the minimum communication and calculates the actual minimum communication cost of (A, B, C, E). Algorithm Opt continues to evaluate (A, B, C, E). Since the low bound of the communication cost of (A, B, C, E) is larger than the communication cost of the candidate best team (A, B, C, D), (A, B, C, E) can be pruned without identifying the hierarchies. Algorithm Opt processes the remaining teams following the above process and finally determines the best team to be (A, B, C, D).

3.3 Algorithm Approx

As mentioned earlier, the hierarchy establishment problem is recognized as the degree-constrained spanning tree problem, a well-known NP-complete problem [5]. Thus, it is computationally expensive to discover the hierarchy of the minimum communication cost for a team. Although algorithm Opt is able to reduce the time of the hierarchy discovery by pruning those qualified teams that cannot be of minimum communication cost, algorithm Opt still suffers from heavy computation on

Table 4. The conferences selected from the DBLP dataset

Category	Conferences
Database	SIGMOD, VLDB, ICDE, ICDT, EDBT, PODS
Data mining	KDD, WWW, SDM, PKDD, ICDM
Artificial intelligence	ICML, ECML, COLT, UAI
Theory	SODA, FOCS, STOC, STACS

hierarchy establishment for the other teams. In a real world, approximate answers are acceptable for those NP-complete problems in many situations. As such, we devise algorithm Approx to obtain nearly-optimal teams and hierarchies. To find nearly-optimal hierarchies of qualified teams, we adopt the 2-opt algorithm proposed in [10] to find a nearly-optimal the degree-constrained minimum spanning tree for each qualified team. To be specific, the main difference between algorithm Approx and algorithm Opt is that algorithm Approx utilizes an approximate algorithm, 2-opt algorithm, for hierarchy establishment of each qualified team rather than the enumeration method used in algorithm Opt to achieve faster problem resolving.

4 Performance Evaluation

4.1 Experimental Setup

To reveal the real social network information, we extract the expert social network from the DBLP bibliography on November 7, 2011. Referring to [7, 8], the DBLP dataset is composed of only the papers published in the prestigious conferences in the areas of database, data mining, artificial intelligent and theory. The selected conferences of the four areas are listed in Fig. 4. To generate the expert social network, we select only the authors publishing at least three papers. Two experts (authors) are connected together if they have worked together with at least two papers. Let p_{x_i} be the set of papers published by expert x_i . Same as [7], for each pair of two connected experts x_i and x_j , the communication cost between x_i and x_j is defined as $1 - \frac{p_{x_i} \cap p_{x_j}}{p_{x_i} \cup p_{x_j}}$.

The proposed three algorithms are implemented in Java and are executed on a PC with an Intel i7 2.93 GHz CPU and 4GB memory. For each experiment, 30 projects are randomly generated by the same method used in [7]. Because the problem addressed in the hierarchy establishment phase is NP-complete, algorithm Brute-Force and algorithm Opt are expected to of much higher execution time. To speed up algorithm Brute-Force and algorithm Opt, we pre-build all degree-constrained spanning trees for small scenarios (i.e., skill number smaller than or equal to 7 and degree constraint smaller than or equal to 6). Besides, the communication cost between any two experts is pre-computed as the prior works [7, 8]. We employ the following three performance metrics to evaluate the performance of the proposed algorithms.

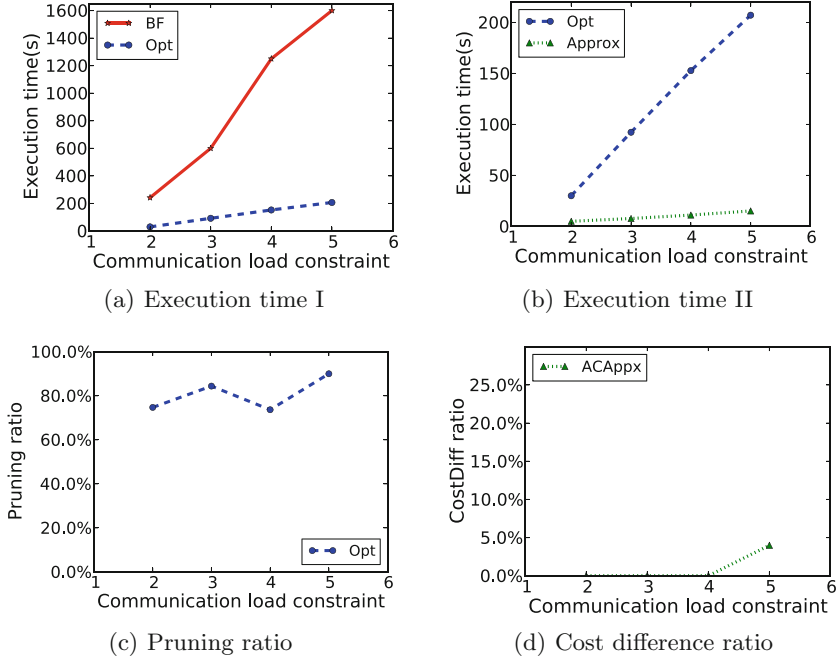


Fig. 3. Impact of the communication load constraint

- Query execution time: The query execution time is defined as the CPU time of each algorithm.
- Pruning ratio: Pruning ratio is used to measure the performance of the pruning strategy used in algorithm Opt. Let n and n_p be the number of teams which can handle the project and the number of teams that are pruned without sent to the hierarchy establishment phase. The pruning ratio is defined as $\frac{n_p}{n}$.
- Cost difference ratio: The cost difference ratio is used to measure the quality of the solutions found by algorithm Approx, and is defined as $\frac{c_{Approx} - c_{Opt}}{c_{Opt}}$ where c_{Opt} and c_{Approx} are the communication costs of the teams found by algorithm Opt and algorithm Approx, respectively.

We use ‘BF’, ‘Opt’ and ‘Approx’ to represent algorithm Brute-Force, algorithm Opt and algorithm Approx, respectively, in the following subsections.

4.2 Impact of the Communication Load Constraint

In this experiment, we measure the impact of the communication load constraints on the proposed algorithms by varying the constraint from 2 to 5. The number of the required skills for a project is set to 6. The experimental result is shown in Fig. 3. As observed in Fig. 3(a), algorithm Opt is more efficient than algorithm Brute-Force due to the effect of pruning. In addition, the speedup of

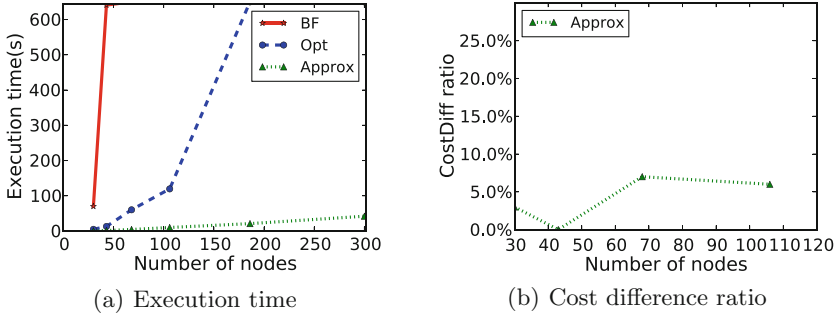


Fig. 4. Impact of the size of the social networks of experts

algorithm Opt over algorithm Brute-Force increases as the increase of the communication load constraint. As shown in Fig. 3(c), it is interesting that the pruning ratio is around 70%–90% as the communication load constraint increases from 2 to 5. The reason is that although the pruning ratio seems not of explicit relationship to the communication load constraint, the execution time of hierarchy establishment phase increases with the increase of the communication load constraint. On the other hand, with an approximation algorithm in the hierarchy establishment phase, it is obvious that algorithm Approx is much more scalable than algorithm Opt. The result depicted in Fig. 3(b) agrees with the intuition. Figure 3(d) reveals that the cost difference ratio is smaller than 6%, showing that the trade-off of algorithm Approx in the quality of solutions is acceptable.

4.3 Impact of the Size of Social Networks

We investigate in this experiment the scalability of these three algorithms by increasing the number of nodes in the social network from 30 to 300. The precise numbers of nodes are 30, 43, 68, 106, 186 and 300. The number of required skills for each project is fixed at 6; the communication load constraint is set to 3. We can see in Fig. 4(a) that algorithm Brute-Force needs over 10 min to solve the team formation problem in the social network with 43 nodes, and has an explosive increase in execution time when the number of nodes is larger than 50. Although being able to deal with the team formation problem in the social network with 150 nodes within 10 min, algorithm Opt is still not scalable enough to handle social networks with more nodes. Due to employing approximation in the hierarchy establishment phase, algorithm Approx is able to handle the team formation problem on the social network with 300 nodes within 60 seconds. Although algorithm Approx cannot find the best team, as shown in Fig. 4(b), the cost difference ratio is smaller than 10% when the number of nodes is smaller than or equal to 106. Experimental result shows that algorithm Approx is suitable for large social network with small trade-off in the quality of solutions.

5 Conclusion

In this paper, we proposed the team formation problem with the communication load constraint in expert social networks. To solve this problem, we presented a two-phase framework and developed three algorithms, namely algorithm Brute-Force, algorithm Opt and algorithm Approx. The experimental results showed that algorithm Opt is able to obtain optimal solutions efficiently by avoiding some teams being sent to the hierarchy establishment phase. Moreover, in the cases that nearly-optimal solutions are acceptable, algorithm Approx is much more scalable than algorithm Opt in large social networks at a cost of small increase in communication cost.

References

1. Anagnostopoulos, A., Becchetti, L., Castillo, C., Gionis, A., Leonardi, S.: Power in unity: forming teams in large-scale community systems. In: Proceedings of the 19th ACM International Conference on Information and Knowledge Management (2010)
2. Baykasoglu, A., Dereli, T., Das, S.: Project team selection using fuzzy optimization approach. *Cybern. Syst.* **38**(2), 155–185 (2007)
3. Chen, S.J., Lin, L.: Modeling team member characteristics for the formation of a multifunctional team in concurrent engineering. *IEEE Trans. Eng. Manage.* **51**(2), 111–124 (2004)
4. Fitzpatrick, E.L., Askin, R.G.: Forming effective worker teams with multifunctional skill requirements. *Comput. Ind. Eng.* **48**(3), 593–608 (2005)
5. Garey, M.R., Johnson, D.S.: *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W.H. Freeman & Co., San Francisco (1979)
6. Gaston, M.E., Simmons, J., des Jardins, M.: Adapting network structure for efficient team formation. In: Proceedings of the AAAI 2004 Fall Symposium on Artificial Multi-Agent Learning (2004)
7. Juang, M.C., Huang, C.C., Huang, J.L.: Efficient algorithms for team formation with a leader in social networks. *J. Supercomput.* **66**, 721–737 (2013)
8. Kargar, M., An, A.: Discovering top-k teams of experts with/without a leader in social networks. In: Proceedings of the 20th ACM International Conference on Information and Knowledge Management (2011)
9. 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 (2009)
10. Singh, S., Srivastava, R., Kumar, V., Agarwal, S.: An approximate algorithm for degree constraint minimum spanning tree. In: International Conference on Computer and Communication Technology (2010)
11. Zakarian, A., Kusiak, A.: Forming teams: an analytical approach. *IIE Trans.* **31**(1), 85–97 (1999)