July 2017, Volume 4, Number 2 (pp. 63–79)
http://www.jcomsec.org

Journal of Computing and Security

Capability-Based Team Formation in Researchers Social Networks

Somayeh Tavakoli [a] Afsaneh Fatemi [a,*]

[a] *Department of Software Engineering, University of Isfahan, Isfahan, Iran.*

ARTICLE INFO.

Article history:
Received: 19 March 2018
Revised: 06 July 2018
Accepted: 07 August 2018
Published Online: 02 December 2018

Keywords:
Social Network, Team Formation, Graph Pruning, Betweenness Centrality, Skill Spreading.

ABSTRACT

Teamwork and cooperative working are increasingly needed due to the growing complexity of scientific problems and highly specialized research projects. Although many studies have been conducted on the subject of team formation in social networks, the challenge of forming a team of experts at optimum cost while maintaining maximum skill fulfillment persists. The main objective in team formation problem is to assemble a team of experts to satisfy all of the technical skill requirements of a given project while minimizing the communication and personnel cost. Electing a good leader for the team leads to better organizing and management of the entire team and also helps to lower the communication cost. Therefore, in this study three algorithms are proposed to identify a team of experts and a leader. These algorithms select the best leader and a team with the minimum cost by pruning the communication graph, identifying the effective nodes and choosing candidates for leadership according to various criteria. Moreover, a new combinational cost function is defined based on the linear combination of the objectives to minimize the personnel and communication costs. The results of experiments on a DBLP data set reveals that these algorithms are faster and more effective compared to other algorithms. The obtained results are due to the omission of excessive nodes according to the skills of experts and the project's requirements, along with choosing the leader based on appropriate criteria.

1 Introduction

Social networks are widely used all around our lives. A social network is a structure consisted of social entities which are mainly persons or organizations and are connected to each other based on one or more dependencies or social ties [1]. These networks might be used to connect people who are colleagues, classmates, friends, or families, and also the people who are forming a team to complete a project. Facebook, MySpace and Twitter are a few general examples, while LinkedIn and Xing are more specialized networks. The members which are connected via social networks generate a positive side effect; hence they can help each other in accomplishing the tasks associated with a project and increase the chance of success [2].

In recent years, teamwork has emerged as one of the key factors of efficiency and work satisfaction in workplace, and the performance of the teams has gained attention and importance. It is proven that the performance of an individual in a team is not only dependent on his/her own skills, but also on the relationship with other group members. All in all, this

* Corresponding author.
Email addresses: s_tavakoli67@yahoo.com (S. Tavakoli), a_fatemi@eng.ui.ac.ir (A. Fatemi)
https://dx.doi.org/10.22108/jcs.2018.110888
ISSN: 2322-4460

has led to more effort being directed toward optimizing the combination of teams. Teams are usually formed from persons with different characters, behaviors and backgrounds who share their complimenting skills and information to accomplish a task or to fulfill a goal. Teams are formed when the expertise and skills of several people are required to obtain a specific goal. In fact, when more than a single person is needed to solve a problem, the communication and cooperation among experts plays a more important role [3–5].

Many of real projects usually require a set of skills provided by team members to be completed successfully. Team working, such that the team of experts possess this set of skills is inevitable toward this goal. The problem of forming a team is to find a team of experts who not only have the required skills, but also can cooperate effectively so that the associated communication and personnel cost is at minimum. Much effort has taken place to study team formation in social networks. However, the challenge of finding a team of experts which has the minimum communication cost while maintaining the required skills is still unresolved. Due to the popularity of social networks and the increasing number of their users, the processing time of existing algorithms in team formation area is prohibitively long [6–10].

In this study, three branch and bound algorithms named SkillSpread_TF, Density_TF and ComPersCost_TF, are introduced to do the team formation based on the members' skills through shrinking the communication graph. The proposed algorithms detect the best leader and the team with minimum cost by omission of excessive nodes and reduction of search space. To evaluate the proposed methods, some experiments are conducted on a real DBLP data set. The performances of proposed algorithms are compared to those of algorithms proposed by Kargar et al. [11, 12] and Juang et al. [13]. The results reveal significantly faster and more efficient performance by proposed algorithms.

In this paper, a literature review is provided in Section 2 after the introduction. Section 3 shortly explains the modeling of team formation with a leader. The proposed methods to obtain the specified objectives are discussed in Section 4, and Section 5 summarizes the results of executing the proposed algorithms in comparison with prior studies. Lastly, Section 6 concludes and suggests some ideas for future work.

2 Related Work

The problem of team formation is one of the important challenges in all organizations. This importance has provoked a lot of research effort. Branch and Bound, Simulated Annealing and genetic algorithm [4–6, 14, 15] are some instances of these efforts. However, none of them has considered the cooperation network among people. The problem of forming a team of experts in the social networks was first introduced by Lappas et al. [10]. Regardless of how the experts' information and their cooperation network are extracted, several methods have been proposed for team formation based on the cooperation network of experts. It is the common element of all these solutions to consider team formation as an optimization problem, although various methods have been devised to solve this problem.

Team formation in social networks can be categorized by two distinct approaches; team formation with leader and team formation without leader. Lappas and his colleagues [10] proposed a new method to calculate the communication cost in social networks while forming a team without a leader. In this method, the communication cost is calculated by means of calculating the longest minimum path between each two nodes of the graph and edges of the minimum spanning tree. In addition, they proved that team formation is an NP-hard problem. Three approximate algorithms named Rarest First, CoverSteiner and EnhancedSteiner were proposed in the study.

Other researchers have worked on various versions of this problem and proposed several algorithms to solve the problem of team formation without a leader. Kargar and An [12] proposed a new function named sum of distances to calculate the communication cost of a team of experts in social networks. In this work, assembling a team of experts under a function which minimized the sum of distances proved to be an NP-hard problem. Hence, an approximate algorithm with an approximation ratio equal to two is provided to solve it.

In addition, Kargar et al. [11] inserted personnel cost into the social network graph as one of the success factors. In this study, each expert is assigned a weight which shows the cost of providing service by that expert. The objective is to find a team of experts which covers all of the required skills while minimizing the personnel and communication costs. In order to solve this problem, a new cost function was defined based on the linear combination of communication and personnel costs and then proved that it is an NP-hard problem. Moreover, an approximate algorithm with an approximation ratio equal to two, and three exploration algorithms were proposed to find optimum solutions. The experiments conducted on two sets of DBLP and IMDB data asserted the effectiveness and scalability of the algorithms.

Zihayat et al. [9] took team formation problem as a multi objective optimization problem and inserted the level of expertise into the equation in addition to considering communication and personnel costs. A two-step method is proposed to solve the problem in this study. In first step, an initial population of an approximate estimation of the effective backup teams is generated. In second step, a local search procedure is applied to each of the solutions in step one to discover more optimized solutions. Experimental results show a good performance by the proposed algorithms compared to others in terms of execution time and response quality.

Li et al. [16] extended the problem of team formation without a leader by linking each skill to a number of experts. In this study, two new methods were proposed by considering the effectiveness of the formed team and also the effectiveness of the team formation process. The methods were based on EnhancedSteiner algorithm originally introduced by Lappas et al. [10]. EnhancedSteiner algorithm was originally developed to tackle general tasks. In addition, the skill node is selected based on the adjacency structure of the skill nodes instead of randomly selecting them according to the original EnhancedSteiner algorithm. Eventually a group based method is proposed to accomplish general tasks.

Rangapuram et al. [17] also considered the team formation in a real setting with a new formula named GDSP (Generalized Densest Subgraph Problem) based on the densest sub-graph. This formula models many pre-requisites of finding a team in a geographical concept or a social concept etc. The prerequisites include: (1) including of a group of given experts (2) meeting team cost limitations (3) finding the team location. The proposed formula leads to generate a generalized version of the densest subgraph with cardinality limitation problem which is an NP-hard problem and has many applications in analyzing social networks. In this paper, a new method is proposed to solve GDSP. Experimental results show that the proposed method is useful in a wider range of team formation problems and generates a more coherent and compressed team with a high quality.

The problem of forming a team of experts with a leader was introduced for the first time by Kargar and An [12] and they introduced a function named leadership-distance to measure the communication cost of a team of experts. In this work a precise polynomial algorithm is introduced to find a team of experts with a leader with the minimum leadership-distance. In this algorithm, each expert is introduced as a candidate for leadership and then the experts with the minimum distance are selected as members of that team. Eventually after evaluating all of the experts the best leader and team are determined.

Juang et al. [13] also worked on team formation with a leader who is responsible for coordination and project management. In this work, two effective algorithms are proposed to identify a team of experts with a leader. These algorithms reduce the search space aiming at accelerating the discovery of the best leader and team. They use the information regarding Betweenness Centrality of the experts and choose the best leadership candidate with the minimum leadership-distance by spreading the skills of each expert to the neighboring experts. Experimental results on a real set of DBLP data shows that the proposed algorithms are faster than previous algorithms introduced in [12].

Tavakoli and Fatemi [18] approached the coalition formation from a social communication graph pruning view point. Team formation problem is modeled as a multi-objective optimization problem. The main goal was minimizing the communication cost.

Dey et al. [19] used the network's small-world characteristics to for a team of players meeting the best performance and the best belongingness properties. The players are the network nodes and their interactions form the edges. They defined a new quantitative measure for the personal performance measurement. This measure is used for selecting the team members from a pool of players. In other words, the quality of a player and his/her team belongingness is quantized based on social networking. The proposed approach is only applicable for the networks with small-world property.

A large part of the body of literature is made of the supervised team formation approaches. Flores-Parra et al. [20] proposed an unsupervised approach, based on the social network analysis. They compared the results of their unsupervised method with the results from the real world supervised ones, and reported the pros and cons of each.

Most of the existing team formation methods, assume that the social members are fully cooperative and want to join the team for task accomplishment. Wang et al. [21] used the crowdsourcing idea in forming teams. They supposed that both the requester and the worker are selfish and they should negotiate to form the working team. Similarly, Tang [22] proposed a profit-driven team formation approach to maximize the overall profit of doing tasks by the team. He presented a perfect formalism for the problem and its variants, and used an approximation algorithm to solve the problem.

Despite many of the existing methods for working group formation, Huang at al. [23] paid attention to team organizational structure beside its ability in satisfying the skill requirements in task performance. They defined the grouped team formation problem and introduced the grouped organization structure inside team, where the cooperation between team members and leaders are considered specially.

The social science considers the team formation problem as an important issue. The research literature in social science can help to solve the problem in social networks. In one of the best researches Gompers et al. [24] studied the role of homophily in team formation using the real data of MBA students. Education, gender, working experience, and ethnicity are studied, and the results showed that all of these factors increase the probability of working as teammates. Due to the results, men tend more to work with teammates with similar level of education and working experience. In addition, ethnicity similarity can promote a weak team to an intermediate one, but has no impression on the efficiency of powerful teams. The ideas of this research and the similar social studies would be used to improve the solutions of team formation problem in special social networks.

3 A Model for Team Formation

In team formation problem with a leader [12, 18], a set of n experts $X = x_1, x_2, \ldots, x_n$ and a set of m skills $S = s_1, s_2, \ldots, s_m$ are assumed. Each expert is linked to a number of skills, hence, the set of skills xi can be described by $S(x_i) \subseteq S$. The expert x_i has the skill s_i, if s_i is a member of the set of skills associated with $x - i(s_i \in S(x_i))$. In addition, a subset of experts $X' \subseteq X$ have skill s_j, at least one of the experts in X' has skill s_j which can be shown by $X(s_j) = (x_i | s_j \in S(x_i))$. The project P requires a set of skills to be completed which are shown as $P = (s_1, s_2, s_p) \subseteq S$. The skill s_i is required for project P, if $s_i \in P$. Project P can be completed by a subset of experts X', if for each required skill of the project there is an expert who possesses that skill $(s_j \in P, \exists x_i \in X', s_j \in S(x_i))$.

The experts in a social network are described by weighted directionless graph $G(X, E)$. Each node of the graph represents an expert $x \in X$, while E is a set of edges between nodes. If two experts x_i and x_j are collaborating with each other, edge $e_{ij} \in E$ exists. Weight e_{ij} shows the communication cost between the two experts x_i and x_j. If two experts collaborate frequently, the communication cost (weight of the edge) would be low for them. A low weight for the communication edge means that the two experts can easily connect to each other and interact. For instance, if two writers in a writer collaboration network share many publications, the weight on their associated communication edge would be low. The distance between two experts x_i and x_j is shown as $D(x_i, x_j)$ which includes the sum of edge's weights of the shortest path between the two experts x_i and x_j. If the two experts are not connected *i.e.* there is no path between them, a very large value is assumed as their distance.

Definition 1 (Leadership Distance). A project with required skills and a team of experts is given. Team T has one expert as its leader $L \in X$. The leader of the team needs to have at least one of the required skills by the project to be considered one of the members. The leadership distance of team T with leader L is defined as the sum of the least distances between the leader and team members (each of the people with required skills of the project):

$$L\_Dist = \sum_{i=1}^{P} D(x_{s_i}.L) \tag{1}$$

where, $D(x_{s_i}, L)$ is the least distance between the leader and the expert with skill s_i.

Definition 2 (Personnel Cost of the Team). Graph G in which the nodes represent experts is given. Each of the experts is linked to a set of skills. For each of the skills linked to the experts the weight representing the cost of service for that skill is assigned which is shown by $PC(x_{s_i})$. In literature [8, 11], each of the experts is assigned a weight which shows the service cost. It is the cost that he/she receives to complete the project. In other words, the service cost of all skills of each expert is assumed equal. In real life, the cost of each skill is different. This study assigns a different cost for each skill of the experts. Given a project with required skills and a team of experts to complete the project, the personnel cost is defined as below:

$$P\_Cost(T) = \sum_{i=1}^{P} PC(x_{s_i}) \tag{2}$$

In which $PC(x_{s_i})$ stands for the personnel cost of expert x for skill s_i and p denotes for the number of required skills of project.

Due to the fact that each expert might undertake more than one skill in the project, the payment to each expert x is $\sum_{i=1}^{k} PC(x_{s_i})$, in which k stands for the number of skills of expert in the project.

The main objective of this study is to minimize the communication and personnel costs at the same time. This is a bi-objective optimization problem. One of the common solutions for bi-objective optimization problems is converting the problem into single objective problem by combining two objective functions in s single function [11]. Therefore, a combinational cost function is defined in this research to combine communication and personnel costs.

Definition 3 (The Team Formation Problem). Assuming a project P with some required skills, graph G of social network that includes a set of experts X, and the trade-off parameter λ between communication and personnel costs, the team formation problem consists of finding a team of experts from graph G for project P such that all of the required skills of the project are covered by the members, such that the defined combinational cost is at the minimum. The combinational cost is defined as follows.

Given a team of experts from graph G for a project and considering the trade-off between communication and personnel costs, the combinational cost of the team can be defined as below:

$$\begin{aligned} Combined\_Cost(T) = &(1-\lambda) * P_Cost(T) \\ &+ \lambda * CC\_LD(T) \end{aligned} \quad (3)$$

Parameter $\lambda (0 \leq \lambda \leq 1)$ annotates the trade-off between communication and personnel cost of the team which is taken from the user.

Since the values of $CC\_LD(T)$ and $P\_Cost(T)$ might be in different scales, they have to be normalized such that both of them fall into the same span before being used in the formula. $P\_Cost(T)$ includes the service providing cost of p experts and $CC\_LD(T)$ stands for the communication cost of p experts (including sum of the minimum distances from the leader to other experts possessing required skills of the project). Since both costs have a similar scale, there is no need for normalization of costs to be used in the formula.

Assumption 1. *In this study, we assume that each required skill of the project P only needs one expert to be completed and each expert may be assigned three skills in the project at maximum. On the other hand, the leadership cost is assumed similar for all of experts and so it is not considered in personnel cost calculations. In other words, if the leader of the team does not have any of the required skills of the project, there will be no extra cost added to personnel cost of the team. Furthermore, service cost is assumed different for each skill of experts. Parameter λ shows the trade-off between communication and personnel cost and is received as an input from the decision maker.*

4 Proposed Methods to Minimize the Team Cost

In this study, some branch and bound algorithms are proposed named SkillSpread_TF and Density_TF in order to minimize the communication cost. In addition, ComPersCost_TF algorithm is proposed to minimize the communication and personnel cost of the team by solving a multi-objective optimization problem.

The social networks' communication graph is very large, and most of the prior related works traverse the entire state space. In this traversal, a large number of visited nodes has no role in formation of the team while imposes longer processing time for the algorithms. The main objective pursued by the proposed algorithm is to make the size of communication graph smaller, which results in decreasing the state space. All in all, a faster process of finding the best leader and team will be executed.

4.1 SkillSpread_TF Algorithm

In this method, given a project with defined required skills, at first step, the shortest path between experts which have at least one of the required skills to other experts is calculated. This is done by Dijkstra algorithm. The Dijkstra algorithm is one of the graph traversal methods which solves the shortest path for positive-weighted. Eventually, this algorithm generates the shortest path tree and provides the shortest path from the origin to all of the nodes of the graph. Algorithm 1 shows the SkillSpread_TF.

Choosing experts which have at least one of the required skills, leads to pruning the people who do not have the required skills at no cost. Sometimes there are experts without any of the required skills, who might be chosen as the leader of the team due to being located in a proper location of communication graph, which can result in a lower final team cost. Hence, while searching for the shortest path between nodes, the Betweenness Centrality of the experts on the traversed paths is also calculated (lines 2-6). In fact, the Betweenness Centrality is utilized as a criterion to identify experts who do not have the required skills of the project but are prone to be chosen as the leadership candidates.

Definition 4 (Betweenness Centrality). Given the graph G(X,E), the Betweenness Centrality of each expert $x \in X$ is shown by $BC(x)$ and is defined as below [13, 25]:

$$BS(x_i) = \sum_{i=1}^{p} \frac{\alpha_{st}(x_i)}{\alpha_{st}}, i \neq s \text{ and } i \neq t \quad (4)$$

In this equation, α_{st} is the total number of paths between two experts x_s and x_t, while $\alpha_{st}(x_i)$ stands for the total number of paths crossing through expert x.

Betweenness centrality is the number of times a node is located between other nodes in a shortest path. In other words, when an expert is located in a short path

Algorithm 1 SkillSpread_TF Algorithm for Team Formation

INPUT: Social Network: G(X,E), A Project P = s_1, s_2, . . . , s_p
OUTPUT: bestTeam, bestLeader, bestLeaderDistance

1: Initialize betweenness centrality (BC) of all experts to 0
2: Set $\acute{X}$ to Experts(X) have skills of project and set all_x = ø
3: **for** each expert x ∈ $\acute{X}$ **do**
4: Add x to all_x
5: run Dijkstra algorithm, and compute all-pair shortest distances Dist(x,t) ∀ x ∈ $\acute{X}$, t ∈ X
6: and BC of nodes on the path from x to t and compute Diameter and Radius of G
7: **end for**
8: compute BC_Threshold = $\frac{MaxBC+MinBC}{2}$
9: select x ∈ X with the BC(x) > BC_Threshold and Add x to x_bc
10: **for** each expert x ∈ x_bc and x ∈ ¬ $\acute{X}$ **do**
11: Add x to all_x
12: run Dijkstra algorithm, and compute all-pair shortest distances Dist(x,t) ∀ x ∈ $\acute{X}$, t ∈ X
13: and Diameter and Radius of G
14: **end for**...s_i, set the skill distance SD(s_i) = $\frac{(Diameter+Radius)/2}{|X(s_i)|}$
15: **for** s_i, set the skill distance SD(s_i) = $\frac{(Diameter+Radius)/2}{|X(s_i)|}$ **do**
16: **for** each expert x ∈ $\acute{X}$ **do**
17: **for** each s ∈ S(x_i) where s ∈ S(P) **do**
18: **for** each neighbor of x where Dist(x,neighbor) < SD(s) and neighbor ∈ all_x **do**
19: spread skill to neighbor, and Add skill to RS(neighbor)
20: **end for**
21: **end for**
22: **end for**
23: bestLeaderDistance = ∞, bestTeam = ø,and bestLeader = ø
24: **for** each expert x ∈ RS in the descending order of the size of the received skill(RS) set **do**
25: compute lower bound Lower_SD for x with respect to project P
26: leader = x, leaderDistance = Lower_SD,and team = ø
27: **if** (leaderDistance > bestLeaderDistance **then**
28: x cannot be the best leader, skip x
29: **end if**
30: **for** each skill s ∈ P ∩ RS(x) first, then s ∈ P ∩ ¬RS(x) **do**
31: select expert v with the smallest distance to x
32: leaderDistance = leaderDistance + dist(x, v), and Add v to team
33: **if** (leaderDistance ≥ bestLeaderDistance **then**
34: x cannot be the best leader, skip x
35: **end if**
36: **end for**
37: bestLeader = leader, bestLeaderDistance = leaderDistance, and bestTeam = team
38: **end for**
39: **end for**
40: return bestTeam, bestLeader, bestLeaderDistance

between other experts, his/her betweenness centrality increases. An expert with a high BC is located in many short paths and has a shorter distance from other nodes compared to an expert with a lower BC. This expert with a high BC has a shorter leadership distance from other experts and is a good candidate for leadership.

In the next step, the shortest path from the experts with a BC value of equal or higher than the threshold to other experts is calculated (lines 7-12). If an expert has a BC value lower than the threshold, he/she will be totally pruned and ignored in next steps.

Definition 5 (Betweenness Centrality threshold value:). Given the calculated Betweenness Centrality of the experts in previous steps, the threshold is defined as below:

$$BC\_Threshold = \frac{MaxBC + MinBC}{2} \tag{5}$$

In this definition, MaxBC is the highest and MinBC is the lowest BC among experts.

Moreover, the radius and diameter of the graph is calculated while searching for the shortest path between experts (lines 5 and 11).

As stated earlier, the distance of two nodes is the shortest path between them. If the largest distance of node i from other nodes of the graph is called "eccentricity", the largest eccentricity of the graph nodes is the diameter and the smallest one is radius of the graph.

After completion of previous steps, the skill distance is calculated for all of the required skills of the project (line 13). Skill distance is used as a criterion of experts' skillspread.

Definition 6 (Skill distance:). The skill distance is calculated by the radius and diameter values obtained in previous steps as below [13]:

$$SD(s_i) = \frac{Diameter + Radius/2}{|X(s_i)|} \tag{6}$$

In this equation $|x(sj)|$ is the total number of experts who have skill sj.

Upon calculation of the skill distance, the skills of experts who possess at least one of the required skills, is spread in the relevant skill distance. It's noteworthy that only the skills which are among the required skills of the project are spread. Furthermore, only those experts will receive the skill who already have at least one of the required skills or their BC is higher than the threshold value. Each expert will store the received skills including those spread by the neighboring experts (lines 14-20).

After fully spreading the skills, the team formation process starts. Juang et al. [13] and Kargar and An [12] consider all of the experts as leadership candidates in team formation process. In the end, the best team will be chosen based on required skills of the project. In this study, in order to decrease the search space, only those experts are considered as leadership candidates who have at least one of the required skills of the project or receive it from their neighboring experts.

After complete spread of skills, the experts are sorted based on their received skills in a decreasing order (line 22). Afterwards, the Lower_SD is calculated for each of the experts by taking the set of skills required by project and also the set of skills received by experts (line 23). Lower_SD is formed by sum of distances which exist in the set of required skills of a given project but are not included in the received skills and also skill sets of each expert. Upper bound of the leadership distance can be calculated for each expert by using Lower_SD. Doing so allows for pruning some experts without choosing a team for them which results in lowering the total calculation cost.

After sorting, each of the experts is chosen sequentially as the leadership candidate and their respective Lower_SD is compared to the best leadership distance which is set to infinity by default. If the Lower_SD turns out to be greater than the best leadership distance, then team formation for that candidate is waived (lines 25-27). In other words, the selected expert cannot be considered as a possible leader due to the high value of leadership distance. If the Lower_SD is lower than the best leadership distance, the team formation process starts. For each of the required skills of the project, experts with the shortest distance to the leader are chosen as team members. While choosing team members for the project, the experts with required skills of the project that are in the list of received skills for leadership candidacy are chosen as team members at first. Otherwise, experts with the minimum cost to the leader are chosen and team members are added consequently (lines 28-34).

Upon completion of the team members based on the required skills of the project, if the leadership distance (including sum of the leadership distances to each member of the team) is greater than the best leadership distance, team formation for the selected leadership candidate ends (lines 31-33). In case the team formation process for the leadership candidate ends, the chosen candidate, the relevant team and the leadership distance will be considered as the best cases of their parameters (line 35). This procedure is repeated for all of experts until the best leader, team and leadership distance is determined (line 37).

Take Figure 1 as an example. Each node represents an expert, the letters next to them represent the skill set of each expert and the weight on the edges shows the communication cost between the two experts. Assuming the project requires skills (d, s, w), the minimum path trees for experts with at least one of the required skills of the project are generated (experts A, B, D, F, G) and BCs of the experts located on the short paths between them are calculated. In this example the BC of experts is equal $((A, 0), (B, 8), (C, 0), (D, 12), (E, 0), (F, 8), (G, 0))$ values. The experts with higher BC are located on more short paths and hence have a better chance of maintaining a lower communication cost to form the team. Therefore, the short path of experts whose BC is greater than the BC_Threshold are also calculated. The threshold value is calculated to identify the important nodes. In this example the corresponding value of the threshold is 6 ((0+12)/2=6). Since the Betweenness centrality of experts C & E is lower than the threshold value, they are pruned. Moreover, the diameter and radius of the graph is calculated while

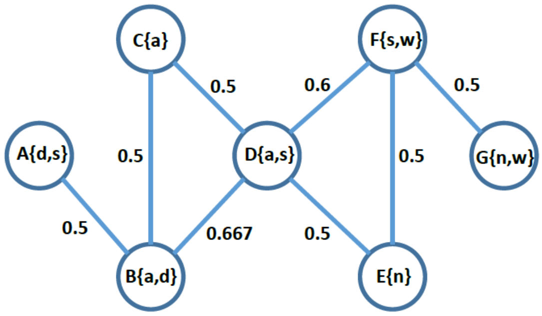


Figure 1. A Sample Social Network of Experts to Explain SkillSpread_TF

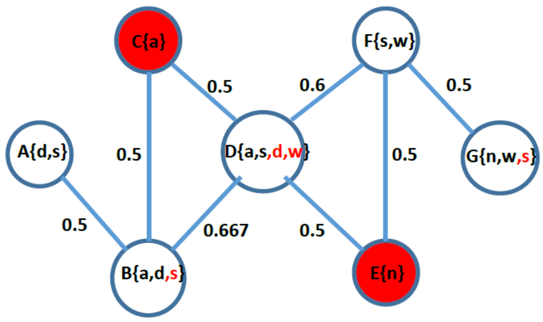


Figure 2. Skill spread after running SkillSpread_TF

calculating the short paths between experts. The diameter and radius in this example are 2.267 and 1.167 respectively.

In next step, the skill distance is calculated per all of the required skills of the project (d,s,w) by using the obtained diameter and radius of previous steps which are 0.8585, 0.5723 and 0.8585 respectively. The experts who possess at least one of the required skills of the project will spread their skills to their neighboring experts within the pertaining skill distance in next steps. For instance, expert A will spread skills d and s only to expert B who has a distance lower than 0.8585 and 0.5723. Final result of skill spread is shown in Figure 2.

In this example, experts C and E were pruned in previous step and did not receive any skill from their neighbors due to not possessing any of the required skills of the project and also their lower BC value than the threshold.

After spreading skills, the experts are sorted in descending order based on the number of their received skills (D, A, B, F, G) and the Lower_SD of them is calculated based on the required skills of the project. For instance, the Lower_SD values for the sorted experts are 0.8585, 0.8585, 0.8585, 0.8585 respectively. Afterwards, each expert is chosen as a leadership candidate and the experts with the shortest distance with respect to the required skills of the project are chosen as team members. Eventually the best leader and team will be chosen. In this example, expert D is selected as the leader and experts (B, D, F) are the final team members with a leadership distance of 1.267.

4.2 Density_TF Algorithm

This method is similar to SkillSpread_TF with some differences in the leader election process. In this method the leader is chosen by applying the density criterion to select leadership candidates which will be explained shortly in the following.

Algorithm 2 shows the Density_TF in pseudo-code form. In this method while searching for the short paths from experts with at least one of the required skills to others, the density of experts who are located on these paths is also calculated (lines 1-6). In fact the density is used as a criterion to identify effective nodes that may minimize the leadership distance if chosen as leadership candidates. In addition, at the team formation step the density is used to sort experts and selecting them as leadership candidates. Therefore, the density values of experts who possess at least one of the required skills of the project are calculated.

Definition 7 (Density:). Given the communication graph G(X,E), the density of each expert $x \in X$ is shown by Density(x_i) and is defined as below [26]:

$$Density(x_i) = \frac{assoc(x_i)}{assoc(G)} \quad (7)$$

In this equation, assoc(x_i) shows the total number of connections for each expert and assoc(G) represents the total connections of the graph G.

After completing the previous step, the shortest paths from experts whose density is equal or greater than the threshold value to other experts are calculated (lines 8-12). If an expert has a density value lower than the threshold, he/she will be totally pruned and ignored in following steps.

Definition 8 (Density threshold value:). Given the calculated density of the experts in previous steps, in this study the threshold is defined as below:

$$Density_T hreshold = \frac{MaxDensity + MinDensity}{2} \quad (8)$$

MaxDensity and MinDensity stand for maximum and minimum density of experts calculated in previous steps respectively.

The team formation process starts at this step. In this process, the experts who have at least one of the project required skills or maintain a density lower than the threshold are sorted in descending order

Algorithm 2 Density_TF Algorithm for Team Formation

INPUT: Social Network: G(X,E), A Project P = $s_1, s_2, \ldots, s_p$
OUTPUT: bestTeam, bestLeader, bestLeaderDistance

1: Set Ẋ to Experts(X) have skills of project and set all_x = ∅
2: **for** each expert x ∈ Ẋ **do**
3: Add x to all_x
4: compute Density(x) = $\frac{assoc(x)}{assoc(G)}$
5: run Dijkstra algorithm, and compute all-pair shortest distances Dist(x,t) ∀ x ∈ Ẋ, t ∈ X
6: and Density of nodes on the path from x to t
7: **end for**
8: compute Density_Threshold = $\frac{MaxDensity+MinDensity}{2}$
9: select x ∈ X with the Density(x) > Density_Threshold and Add x to x_den
10: **for** each expert x ∈ x_den and x ∈ ¬Ẋ **do**
11: Add x to all_x
12: run Dijkstra algorithm, and compute all-pair shortest distances Dist(x,t) ∀ x ∈ Ẋ, t ∈ X
13: **end for**
14: bestLeaderDistance = ∞
15: bestTeam = ∅
16: bestLeader = ∅
17: **for** each expert x ∈ all_x in the descending order of the size of Density **do**
18: leader = x
19: leaderDistance = 0
20: team = ∅
21: **for** each skill s ∈ P **do**
22: select expert v with the smallest distance to x
23: leaderDistance = leaderDistance + dist(x, v)
24: add v to team
25: **if** (leaderDistance ≥ bestLeaderDistance **then**
26: x cannot be the best leader, skip x
27: **end if**
28: **end for**
29: bestLeader = leader
30: bestLeaderDistance = leaderDistance
31: bestTeam = team
32: **end for**
33: return bestTeam, bestLeader, bestLeaderDistance

with respect to their density values. Afterwards, each expert is chosen as a candidate for leadership and the team formation process is executed according to the first proposed method. This procedure is repeated for all of the experts until the best leader, team and leadership distance is determined (lines 16-32).

4.3 ComPersCost_TF Algorithm

The pseudo-code of ComPersCost_TF algorithm is shown in Algorithm 3. In this method, after defining a project with its required skills and also the tradeoff parameter λ between the communication and personnel costs, the shortest path from experts with at least one of the required skills of the project to others will be calculated (lines 2-6). Choosing experts which have at least one of the project required skills, leads to cost-free pruning the people who do not have the required skills. Similar to the first method, betweenness centrality criterion (definition 4) is used in this process as well. In the next step, the shortest path from the experts with a BC value of equal or higher than the threshold (definition 5) to other experts is calculated (lines 7-12). If an expert has a BC value lower than the threshold, he/she will be totally pruned and ignored in following steps.

Following on, the skills of experts with at least one of the required skills of the project are spread to all other experts having the same condition or a BC value lower than threshold. Only those skills are spread which are included in the list of required skills of the project. Moreover, the combinational cost is also calculated during spreading of the skills. If the neighboring expert already has a skill or receives it from another expert, the repeated skill is replaced by an existing one only if has a lower combinational cost. Again, if both of combinational costs are equal, the skill with lower cost will be selected (lines 13-26).

While spreading the skills, each expert stores the set of received skills and the associated combinational cost function with their respective communication and personnel costs. After completion of spreading the skills, experts have received all of the required skills of the project and their associated costs from their neighbors.

After completion of skill spreading, all of the experts who received skills from their neighbors in previous steps are sorted based on sum of the combinational cost of all of their skills in descending order. After sorting experts, the first one with the least cost is chosen as the leader and then experts who are in his/her list of received skills are chosen as the final team members. In other words, the experts with the lowest cost in comparison to the leader who have received skills from him/her in the spreading step and are stored in the corresponding list of received skills are chosen as team members.

5 Evaluation

In this section, the proposed algorithms are evaluated according the performance. All of the proposed algorithms and the ones that were taken into account to make comparisons were implemented in C# language. The experiments in this section were conducted by a quad core 2.8 GHz processor with 16 GB of RAM and Windows 10 operating system. Each experiment was run several times and the average of results was reported.

5.1 The Datasets

Due to lack of a single standard data set, in this study a DBLP data set was extracted to evaluate the performance of proposed methods. In this data set, each of the writers who have published at least one paper are assumed as experts. If more than one writer was involved in publishing the paper, an edge would be assumed between them and a default value is assumed as the weight on that edge to show the communication cost between them. If two experts have multiple collaborative projects, the communication cost between them is decreased and for each time of collaboration between them, one unit is deducted from the weight of their communication edge.

After extracting the experts' names and creating the communication edges between them (according to the collaboration of experts in publishing papers), each of the experts is assigned a set of skills (between one to ten), chosen from more than 100 existing skills in the network. As mentioned earlier, on the contrary to previous works, in this study each skill of the expert is assigned a distinct cost. As a result, a random number between 50 and 200 was assigned to each skill of the experts as the service cost of that specific skill.

The generated graph from DBLP data set included 5003 experts, 23957 edges and 37331 skills.

5.2 Evaluation Criteria

In order to evaluate the performance of proposed algorithms, these criteria were considered:

- Execution time of the algorithm: the total execution time of the team formation process which included the routing time among experts to find the shortest path.
- Communication cost
- Personnel cost
- Pruning rate of experts: If there are a total of n experts and np stands for the number of them who are pruned, the pruning rate can be defined as n_p/n. The pruning rate of the experts is evaluated at two points; one before starting the team formation process and the other while the process is ongoing.

5.3 Evaluation and Analysis of Performance for SkillSpread_TF and Density_TF Methods

In order to show the performance improvement in proposed methods (Density_TF and SkillSpread_TF) these methods were compared to the algorithms proposed by Kargar et al. [12] and Juang et al. [13] named BCPruning and SSPruninig.

Since Density_TF and SkillSpread_TF algorithms and the other algorithms in this comparison, all use the leadership distance function and eventually detect the best leader based on the minimum communication cost, the communication cost and team cardinality are assumed equal for all algorithms. Therefore, the comparison of these two criteria is waived in this section.

First scenario: The impact of the number of required skills of the project

In this scenario the effect of the number of required skills of the project is studied. The number of required skills of the project is increased from 4 to 10 with increments of 2. In these experiments the number of experts is assumed constant at 1500 people.

Figure 3 shows that the execution time of all algorithms gets longer by increasing the number of re-

Algorithm 3 ComPersCost_TF Algorithm for Team Formation

INPUT: Social Network: G(X,E), A Project P = s_1, s_2, . . . , s_p, trade-off λ between Communication and Personnel Cost

OUTPUT: best Team, Leader, LeaderDistance, PersonnelCost, CombinedCost

```
Initialize betweenness centrality (BC) of all experts to 0
Set X́ to Experts(X) have skills of project and set all_x = ∅
for each expert x ∈ X́ do
    Add x to all_x
    run Dijkstra algorithm, and compute all-pair shortest distances Dist(x,t) ∀ x ∈ X́, t ∈ X
    and BC of nodes on the path from x to t
end for
compute BC_Threshold = (MaxBC+MinBC)/2
select x ∈ X with the BC(x) > BC_Threshold and Add x to x_bc
for each expert x ∈ x_bc and x ∈ ¬X́ do
    Add x to all_x
    run Dijkstra algorithm, and compute all-pair shortest distances Dist(x,t) ∀ x ∈ X́, t ∈ X
end for
for each expert x ∈ X́ do
    for each s ∈ S(x_i) where s ∈ S(P) do
        for each neighbor of x neighbor ∈ all_x do
            if (s ∈ ¬S(neighbor) or Combined_Cost(s) < Combined_Cost(s ∈ S(neighbor)) then
                Spread skill to neighbor with Cost
                Add skill to RS(neighbor) with Cost
            else
                if (s ∈ S(neighbor) and Combined_Cost(s) = Combined_Cost(s ∈ S(neighbor)) then
                    respect to λ , PC(xs) and L_Dist(x) Spread skill to neighbor or Skip
                end if
            end if
        end for
    end for
    select first expert x ∈ RS in the ascending order of the sum of the Combined_Cost and set to
    bestLeader = x, bestTeam = team(x)
    bestLeaderDistance = L_Dist(x)
    bestPersonnelCost = P_Cost(x)
    bestCombinedCost = Combined_Cost(x)
end for
return bestTeam, bestLeader, bestLeaderDistance, bestPersonnelCost, bestCombinedCost
```

quired skills of the project. This observation is justifiable by noting that increasing the number of skills, translates into a longer time consumption to form a team for each leadership candidate.

SkillSpread_TF and Density_TF algorithms are more effective in reducing the execution time when the number of required skills of the project grows due to pruning the experts. The underlying reason for this improvement is pruning experts which results in reducing the search space and consequently a lower calculation cost. As shown in Figure 3, the Density_TF and SkillSpread_TF algorithms have almost similar time consumption for projects with a low number of required skills. However, at high numbers of required skills, the Density_TF takes less execution time than SkillSpread_TF due to lower calculation complexity.

As discussed earlier, the underlying reason of improved performance of Density_TF and Skill-Spread_TF algorithms is pruning the communication graph and omission of some experts in the routing step and also the method of choosing the initial leader in team formation which all in all results in a higher speed of team formation in comparison to other algorithms.

In skill spreading step, as opposed to SSPruning algorithm in which all of the expert's skills are spread to all of neighbors who have a lower distance from the associated distance of those skills, the SkillSpread_TF only spreads the skills which are in the set of required skills of the project. Moreover, only the experts who have at least one of required skills of the project in addition to maintaining a lower distance than the associated skill distance or those whose BC is lower

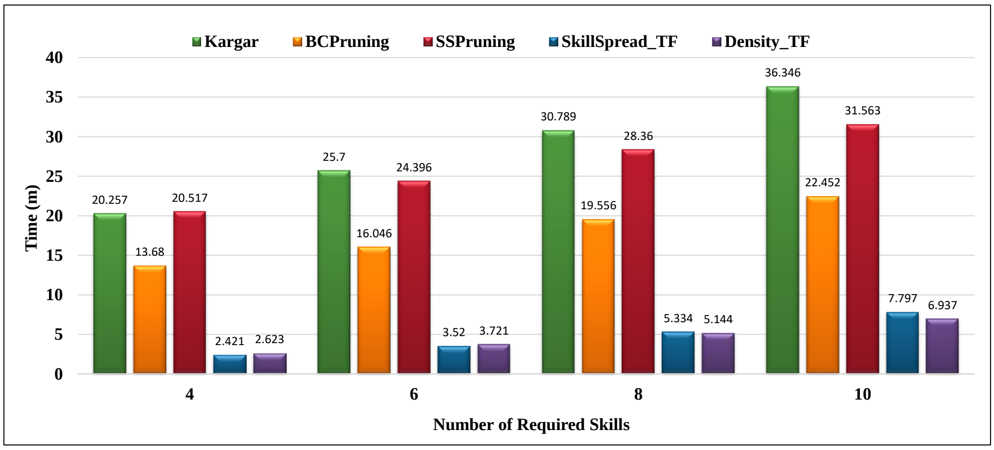


Figure 3. The Impact of Number of Required Skills on Execution Time for Single Objective Algorithms

than the threshold are prone to receive the skills. This reduces the calculation cost compared to the SSPruning algorithm.

The SkillSpread_TF algorithm calculates the lower bound of the leadership distance for each expert by taking the required skills of the project into account and ignores some of the experts for leadership candidacy. On the other hand, by sorting experts based on the number of their received skills in SkillSpread_TF algorithm, and sorting them based on density in Density_TF there is a good chance that individuals with lower leadership distance are chosen as leadership candidates. As a result, most of the leadership candidates are pruned while comparing possible team member's leadership distance with the best leadership distance and the process terminates for them.

On the contrary, Kargar algorithm considers all of the experts as leadership candidates and then selects a team for them by picking the experts with the lowest leadership distance values. In the end, after evaluating all of experts and forming teams for them the best leader and team are selected. This process has significant calculation cost and takes a long execution time.

The SSPruning algorithm is capable of omitting some of the experts at no calculation cost for team member selection by calculating the lower bound for leadership distance. Moreover, the sorting of experts is based on the number of received skills in skill spreading step which in combination with selecting the leadership candidates in a descending order facilitate ignoring some of the experts by comparing their leadership distance to the best leadership distance. Therefore the execution time is improved compared to Kargar's algorithm.

The BCPruning algorithm also sorts the experts based on their betweenness centrality in a descending order and chooses each one as the leadership candidate. The experts are pruned by comparing their leadership distance to the best distance at team member selection stage which results in a lower execution time than Kargar and SSPruning.

Second Scenario: The impact of number of experts

In this scenario, the impact of the number of experts on the performance of algorithms is analyzed. In these experiments the number of required skills of the project is assumed constant at ten. As illustrated in Figure 4, the execution time of the algorithms increases by adding experts. The reason lies in the fact that this increase leads to a greater number of leadership candidates and also the number of people who possess those specific skills. SkillSpread_TF and Density_TF express a more optimized response by increasing the number of experts. The experimental results show that the execution times of the proposed algorithms are significantly improved compared to other algorithms and they prove to be more efficient.

Third scenario: Analyzing the pruning rate of experts

In this study, the pruning rate is analyzed in two steps. The first step is before starting the team formation (step1) in which experts are pruned at no extra calculation cost with respect to pruning at team formation step. The second step refers to pruning at the time of team formation (step2).

The experimental results are shown inFigure 5. In these experiments the number of experts is assumed constant at 750 while there are 10 required skills for the project.

BCPruning and SSPruning algorithms are capable of pruning 97% of experts at team formation step which results in a shorter execution time compared

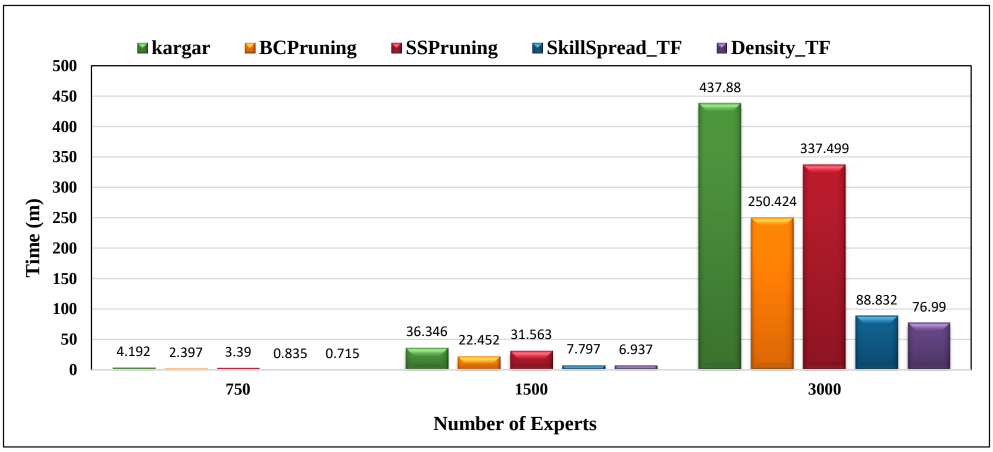


Figure 4. The Impact of Number of Experts on the Execution Time for Single Objective Algorithms.

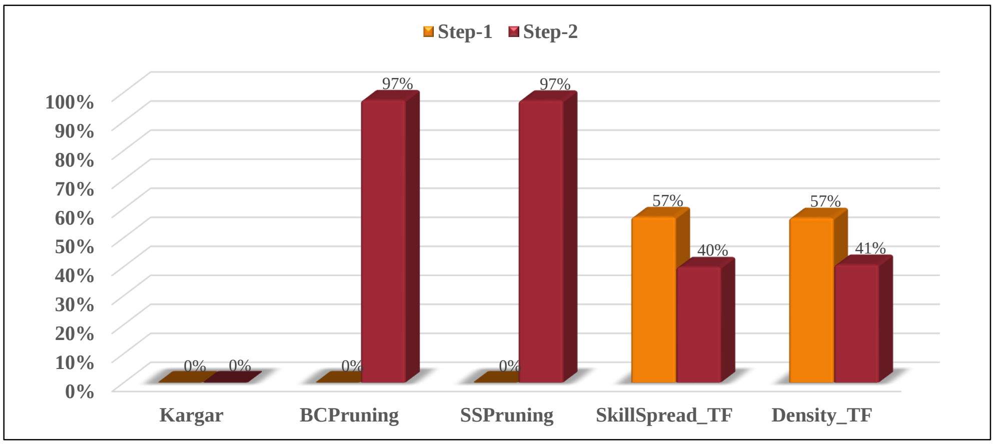


Figure 5. The Pruning Rate of Experts in Single Objective Algorithms

to Kargar algorithm. However, the execution time of SkillSpread_TF and Density_TF algorithms due to pruning 57% of experts before team formation process even starts, and also pruning 40% of them while choosing the team members, is significantly lower than other methods.

5.4 Performance Analysis and Comparison of the Three Proposed Methods

In this section, the proposed algorithms are compared to each other to demonstrate their improvement trends.

First scenario: The impact of the number of required skills of the project

In this scenario the effect of the number of required skills of the project is studied. The number of required skills of the project is increased from 4 to 10 with increments of 2. In these experiments the number of experts is assumed constant at 1500 people.

The experimental results are shown in Figure 6. As shown in this figure, the execution time of ComPersCost_TF algorithm is faster and more efficient compared to other methods.

The underlying reason of this improvement is that the associated cost is spread simultaneously with skills of the experts. Hence, on the contrary to SkillSpread_TF and Density_TF algorithms, in which all of the experts who possess at least one of the required skills of the project or maintain a BC or density higher than the threshold values are considered as leadership candidates, this algorithm takes the first leadership candidate with the lowest combinational cost as the best leader just after completion of skill spreading. In fact, pruning experts before team formation process results in lower calculation cost and execution time for the algorithm.

Second Scenario: The impact of number of experts

In this scenario, the impact of number of experts on

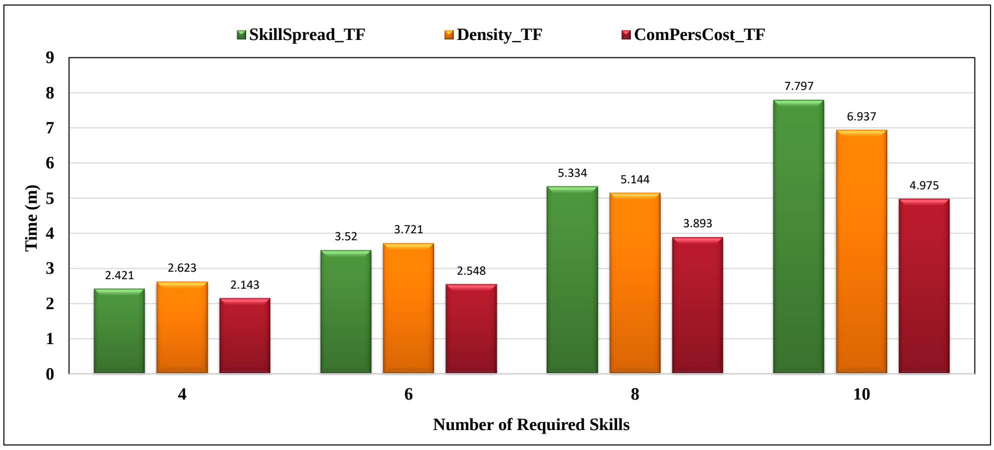


Figure 6. The Impact of the Number of Required Skills on the Execution Time of Proposed Algorithms

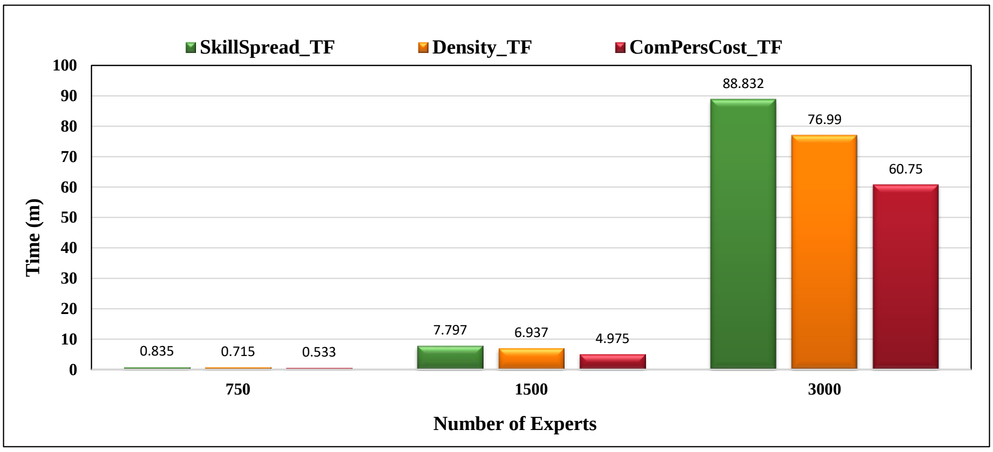


Figure 7. The Impact of Number of Experts on the Execution Time of Proposed Algorithms

the performance of different algorithms is studied. In these experiments the number of required skills of the project is assumed constant at ten. The results are shown in Figure 7. The execution time increases when the number of experts grows as illustrated in Figure 7. ComPersCost_TF exhibits a faster and more efficient performance in this case compared to SkillSpread_TF and Density_TF algorithms and expresses more scalability with regards to the number of experts.

Third scenario: Analyzing the pruning rate of experts

In these experiments the number of experts is assumed constant at 750 while there are 10 required skills for the project. The experimental results are shown in Figure 8.

6 Conclusions and Future Work

In this study three algorithms were proposed named SkillSpread_TF, Density_TF and ComPersCost_TF, to reduce the size of communication graph, in order to improve the team formation with a leader given the social networks' structure. In addition to minimizing the combinational cost (communication and personnel costs) while choosing the best leader and associated team using the proposed methods, a faster execution time was obtained by reducing the search space.

Due to lack of a single standard data set, in this study a DBLP data set was extracted to evaluate the performance of proposed methods. To evaluate the efficiency of proposed methods, the algorithm execution time, communication cost, personnel cost and pruning rate of experts are taken into account. The proposed algorithms were compared to the algorithms introduced by Kargar et al. [11, 12] and Juang et al. [13]. The conducted experiments on a DBLP data set proved that the proposed algorithms not only

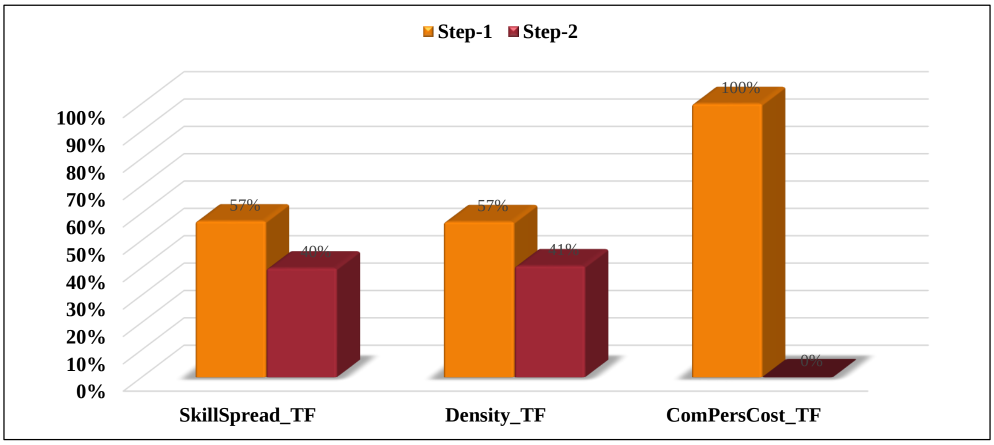


Figure 8. The Pruning Rate of Experts in Proposed Algorithms

minimized the defined cost (either communication cost or the combinational cost) but also had higher efficiency than other methods. Furthermore, the early pruning of communication graph and reducing the search space while considering multiple criteria for choosing the leader, reduced the execution time of the proposed algorithms significantly in comparison to other methods.

In order to continue this study, a number of other criteria can be considered for best leader election and also several other objectives can be defined to measure the performance. An uncertain number of experts for each required skill of the project is one of the possible variations of team formation problem. Given the leadership cost of experts is assumed constant in this study, making this parameter a variable calls for new research and different approaches to solve the problem. Lastly, assigning various budgets to one objective and optimizing some other objectives can be another approach to continue research on this topic.

Acknowledgements

This research was supported by Department of Computer Engineering, University of Isfahan. The authors thank Dr. Bahman Zamani and Dr. Marjan Kaedi for their assistance to improve the manuscript.

References

[1] Waleed M. Al-Adrousy, Hesham A. Ali, and Taher T. Hamza. A recommender system for team formation in MANET. *Journal of King Saud University - Computer and Information Sciences*, 27(2):147 – 159, 2015. ISSN 1319-1578. doi:https://doi.org/10.1016/j.jksuci.2014.06.014.

[2] Meenal Chhabra, Sanmay Das, and Boleslaw Szymanski. Team Formation in Social Networks. In Erol Gelenbe and Ricardo Lent, editors, *Computer and Information Sciences III*, pages 291–299, London, 2013. Springer London. ISBN 978-1-4471-4594-3. doi:10.1007/978-1-4471-4594-3_30.

[3] Jian Wang and Jia Zhang. A win–win team formation problem based on the negotiation. *Engineering Applications of Artificial Intelligence*, 44:137 – 152, 2015. ISSN 0952-1976. doi:10.1016/j.engappai.2015.06.001.

[4] F. Farhadi, M. Sorkhi, S. Hashemi, and A. Hamzeh. An Effective Expert Team Formation in Social Networks Based on Skill Grading. In *2011 IEEE 11th International Conference on Data Mining Workshops*, pages 366–372, Dec 2011. doi:10.1109/ICDMW.2011.28.

[5] Hyeongon Wi, Seungjin Oh, Jungtae Mun, and Mooyoung Jung. A team formation model based on knowledge and collaboration. *Expert Systems with Applications*, 36(5):9121 – 9134, 2009. ISSN 0957-4174. doi:10.1016/j.eswa.2008.12.031.

[6] Gaganmeet Kaur Awal and K. K. Bharadwaj. Team formation in social networks based on collective intelligence – an evolutionary approach. *Applied Intelligence*, 41(2):627–648, Sep 2014. ISSN 1573-7497. doi:10.1007/s10489-014-0528-y.

[7] Aris Anagnostopoulos, Luca Becchetti, Carlos Castillo, Aristides Gionis, and Stefano Leonardi. Online Team Formation in Social Networks. In *Proceedings of the 21st International Conference on World Wide Web*, WWW '12, pages 839–848, New York, NY, USA, 2012. ACM. ISBN 978-1-4503-1229-5. doi:10.1145/2187836.2187950.

[8] Mehdi Kargar, Morteza Zihayat, and Aijun An. Affordable and collaborative team formation in an expert network. Technical Report CSE-2013-01, Department of Computer Science and Engineering, York University, Technical Report CSE-

2013, January 2013.

[9] M. Zihayat, M. Kargar, and A. An. Two-Phase Pareto Set Discovery for Team Formation in Social Networks. In *2014 IEEE/WIC/ACM International Joint Conferences on Web Intelligence (WI) and Intelligent Agent Technologies (IAT)*, volume 2, pages 304–311, Aug 2014. doi:10.1109/WI-IAT.2014.112.

[10] Theodoros Lappas, Kun Liu, and Evimaria Terzi. Finding a Team of Experts in Social Networks. In *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '09, pages 467–476. ACM, 2009. ISBN 978-1-60558-495-9. doi:10.1145/1557019.1557074.

[11] Mehdi Kargar, Aijun An, and Morteza Zihayat. Efficient Bi-objective Team Formation in Social Networks. In Peter A. Flach, Tijl De Bie, and Nello Cristianini, editors, *Machine Learning and Knowledge Discovery in Databases*, pages 483–498, Berlin, Heidelberg, 2012. Springer Berlin Heidelberg. ISBN 978-3-642-33486-3. doi:10.1007/978-3-642-33486-3_31.

[12] Mehdi Kargar and Aijun An. 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*, CIKM '11, pages 985–994. ACM, 2011. ISBN 978-1-4503-0717-8. doi:10.1145/2063576.2063718.

[13] Ming-Chin Juang, Chen-Che Huang, and Jiun-Long Huang. Efficient algorithms for team formation with a leader in social networks. *The Journal of Supercomputing*, 66(2):721–737, Nov 2013. ISSN 1573-0484. doi:10.1007/s11227-013-0907-x.

[14] Adil Baykasoglu, Turkay Dereli, and Sena Das. PROJECT TEAM SELECTION USING FUZZY OPTIMIZATION APPROACH. *Cybernetics and Systems*, 38(2):155–185, 2007. doi:10.1080/01969720601139041.

[15] ARMEN Zzkarian and ANDREW Kusiak. Forming teams: an analytical approach. *IIE Transactions*, 31(1):85–97, Jan 1999. ISSN 1573-9724. doi:10.1023/A:1007580823003.

[16] Cheng-Te Li, Man-Kwan Shan, and Shou-De Lin. On team formation with expertise query in collaborative social networks. *Knowledge and Information Systems*, 42(2):441–463, Feb 2015. ISSN 0219-3116. doi:10.1007/s10115-013-0695-x.

[17] Syama Sundar Rangapuram, Thomas Bühler, and Matthias Hein. Towards Realistic Team Formation in Social Networks Based on Densest Subgraphs. In *Proceedings of the 22Nd International Conference on World Wide Web*, WWW '13, pages 1077–1088. ACM, 2013. ISBN 978-1-4503-2035-1. doi:10.1145/2488388.2488482.

[18] S. Tavakoli and A. Fatemi. Improving team-formation in social networks with communication graph pruning. In *Proceedings of the 21th CSICC national Conference*, pages 362–367, 2016.

[19] Paramita Dey, Maitreyee Ganguly, and Sarbani Roy. Network centrality based team formation: A case study on T-20 cricket. *Applied Computing and Informatics*, 13(2):161 – 168, 2017. ISSN 2210-8327. doi:https://doi.org/10.1016/j.aci.2016.11.001.

[20] Josue Miguel Flores-Parra, Manuel Castanon-Puga, Luis-Guillermo Martinez, Ricardo Rosales-Cisneros, and Carelia Gaxiola-Pacheco. Towards Recommendation System for Work-group Formation Using Social Network Analysis Approach. In *Proceedings of The World Congress on Engineering and Computer Science 2017*, pages 508–513, 2017.

[21] W. Wang, J. Jiang, B. An, Y. Jiang, and B. Chen. Toward Efficient Team Formation for Crowdsourcing in Noncooperative Social Networks. *IEEE Transactions on Cybernetics*, 47 (12):4208–4222, Dec 2017. ISSN 2168-2267. doi:10.1109/TCYB.2016.2602498.

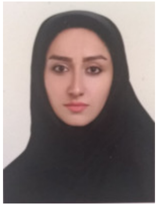
[22] Shaojie Tang. Profit-Driven Team Grouping in Social Networks. In *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence*, pages 45–51, 2017. URL `https://aaai.org/ocs/index.php/AAAI/AAAI17/paper/view/14791`.

[23] Jianbin Huang, Ze Lv, Yu Zhou, He Li, Heli Sun, and Xiaolin Jia. Forming Grouped Teams with Efficient Collaboration in Social Networks. *The Computer Journal*, 60(11):1545–1560, 2017. doi:10.1093/comjnl/bxw088.

[24] Paul A Gompers, Kevin Huang, and Sophie Q Wang. Homophily in Entrepreneurial Team Formation. Working Paper 23459, National Bureau of Economic Research, May 2017.

[25] Francesco Buccafurri, Gianluca Lax, Serena Nicolazzo, Antonino Nocera, and Domenico Ursino. Measuring Betweenness Centrality in Social Internetworking Scenarios. In Yan Tang Demey and Hervé Panetto, editors, *On the Move to Meaningful Internet Systems: OTM 2013 Workshops*, pages 666–673. Springer Berlin Heidelberg, 2013. ISBN 978-3-642-41033-8. doi:10.1007/978-3-642-41033-8_84.

[26] Carla Sofia Pereira and António Lucas Soares. Improving the quality of collaboration requirements for information management through social networks analysis. *International Journal of Information Management*, 27(2):86 – 103, 2007. ISSN 0268-4012. doi:https://doi.org/10.1016/j.ijinfomgt.2006.10.003.



Somayeh Tavakoli received her B.Sc. in computer engineering from the Jahad Higher Education Institute of Isfahan, Isfahan, Iran, in 2012 and her M.Sc. in Computer Software Engineering from the University of Isfahan, Isfahan, Iran, in 2016. Her research interest is in the area of Social Networks' challenges.

Afsaneh Fatemi received her B.Sc. from the Isfahan University of Technology, Isfahan, Iran, in 1995, and her M.Sc. and Ph.D. from the University of Isfahan, Isfahan, Iran, in 2001 and 2012, all in Computer Software Engineering. Dr. Fatemi joined the Faculty of Computer Engineering in 2013, as an assistant professor. Her main research interests lie in Complex Networks, Multi-Agents Systems, and Emergency-Response Environments' Modeling. She has collaborated with the of Model Driven Software Engineering research group (MDSE) and the Big Data research group at the University of Isfahan from 2014 and 2017, respectively.