Computers & Operations Research 74 (2016) 127–142

Contents lists available at ScienceDirect

Computers & Operations Research

journal homepage: www.elsevier.com/locate/caor

Social structure optimization in team formation

Alireza Farasat[*], Alexander G. Nikolaev

Department of Industrial and Systems Engineering, University at Buffalo (SUNY), Buffalo, NY, USA

ARTICLE INFO

Article history:
Received 23 April 2015
Received in revised form
24 April 2016
Accepted 26 April 2016
Available online 28 April 2016

Keywords:
Team formation problem
Social network analysis
Combinatorial optimization
Discrete optimization applications
Lin–Kernighan heuristic

ABSTRACT

This paper presents a mathematical framework for treating the Team Formation Problem explicitly incorporating Social Structure (TFP-SS), the formulation of which relies on modern social network analysis theories and metrics. While recent research qualitatively establishes the dependence of team performance on team social structure, the presented framework introduces models that quantitatively exploit such dependence. Given a pool of individuals, the TFP-SS objective is to assign them to teams so as to achieve an optimal structure of individual attributes and social relations within the teams. The paper explores TFP-SS instances with measures based on such network structures as edges, full dyads, triplets, k-stars, etc., in undirected and directed networks. For an NP-Hard instance of TFP-SS, an integer program is presented, followed by a powerful Lin–Kernighan-TFP (LK-TFP) heuristic that performs variable-depth neighborhood search. The idea of such λ-opt sequential search was first employed by Lin and Kernighan, and refined by Helsgaun, for successfully treating large Traveling Salesman Problem instances but has seen limited use in other applications. This paper describes LK-TFP as a tree search procedure and discusses the reasons of its effectiveness. Computational results for small, medium and large TFP-SS instances are reported using LK-TFP and compared with those of an exact algorithm (CPLEX) and a Standard Genetic Algorithm (SGA). Finally, the insights generated by the presented framework and directions for future research are discussed.

1. Introduction

The success of a project as well as the productivity of a whole organization often depends on the effectiveness and efficiency of work of participating teams [3]. The challenge of assembling successful teams can be addressed by formulating a problem of grouping individuals or assigning them to (sub)sets so as to optimize some outcome-related objectives [57]. Team Formation Problem (TFP) has received attention from the operations research community over the past years [17,26,23,56]. However, despite the common understanding that the social structure between members of the same team plays an important role in the team's output, such consideration has not been explicitly taken into account in mathematical modeling, primarily due to the lack of quantitative means to do so [36,59].

This paper addresses the challenge of developing a mathematical framework for incorporating social structure measures into TFP. It identifies the means to quantify social structure by assessing the impact of each individual's local network on their work-related outcome. For example, such outcome can be the amount of goods produced, the number of errors committed (self-reported or observed), it can be some job satisfaction indicator, the frequency of conflicts at workplace, etc. Rooted in social science theories, the presented framework allows one to build models for TFP explicitly incorporating Social Structure (TFP-SS). The class of TFP-SS models sheds light on team building strategies and also advances the emerging quantitative research of social theories and team outcomes [40,15].

The presented framework elucidates the connection between work environment, social network theories and measurable team outcomes: see Fig. 1. Social network theories motivate the use of graph-based constructs, called network structures, for representing social relations: such network structures include edges, full dyads, k-stars, and (un)directed triplets, among others. Theories of social exchange, structural holes, homophily, reciprocity, transitivity and network evolution described later in Section 3 support the design of interpretable network structure measures as functions of network structures in TFP-SS (e.g., the number of transitive triplets in a given graph). The resulting models are useful for both descriptive and prescriptive purposes. Given historical work-related outcome data for differently structured teams, the researcher can quantify the impact of each theory on team performance, by estimating the weight of each respective network structure measure in a model of the outcome. Then, by adjusting team roster decision variables, the outcome can be driven in the

[*] Corresponding author.
E-mail addresses: afarasat@buffalo.edu (A. Farasat), anikolae@buffalo.edu (A.G. Nikolaev).

http://dx.doi.org/10.1016/j.cor.2016.04.028
0305-0548/© 2016 Elsevier Ltd. All rights reserved.

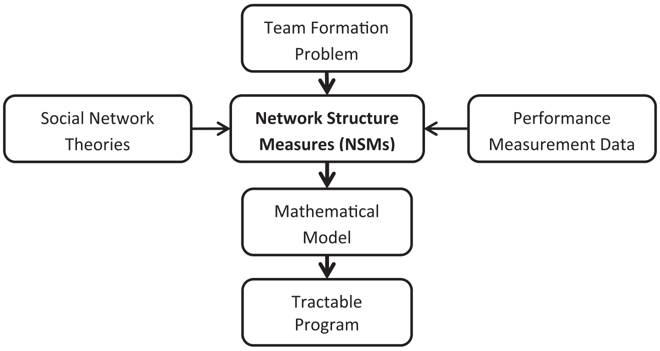


Fig. 1. The proposed framework for the TFP.

desired direction.

This optimization-permitting ability together with the reliance of the TFP-SS models on social theories distinguishes the presented framework from the existing network clustering, community detection and clique problems literature [50,44,51]. Importantly, the presented framework allows one to more closely control individual team members' local networks, which play a big role in information transmission, according to the structural holes theory. This paper also presents an extremely efficient Lin–Kernighan TFP (LK-TFP) algorithm for solving TFP-SS, based on variable depth first neighborhood search.

To summarize, this paper presents a framework for formulating and solving team formation problems that employ information provided by the local social network of individuals. This framework addresses problems at a meeting point of social science and operations research that have significant practical appeal. The efforts in building and treating models for TFP-SS lead to the creation of a methodological toolbox that quantifies social and behaviorial aspects of working in teams, particularly in professional nursing, rescue, police operations, sport teams and academic research. The paper's key contributions are three-fold.

(1) It motivates and justifies the use of mathematical programming and optimization techniques in the area of social science, where most problems have been previously qualitatively addressed by observations, experiments and basic statistical methods.
(2) It presents a prescriptive, quantitative approach to a real-world application of social network analysis, as opposed to the existing descriptive studies. It also introduces a framework to operations research, computer and social scientists for modeling more complex problems in the area of social science from an operations research perspective.
(3) It identifies the relation between established social science findings and team outcomes. It presents explicit, rigorous functions of social structures to evaluate the outcomes. It also describes how social structures and individual attributes can be incorporated into mathematical models of the outcome regardless of the network type (e.g., directed, undirected, weighted or unweighed).
(4) Designing and testing methods for solving the TFP where the optimal social structure is sought within the teams. The paper presents both an exact method and an efficient heuristic exploiting the Lin–Kernighan-inspired variable depth neighborhood search.

The rest of the paper is organized as follows. Section 2 offers a review on existing models based on Social Network Analysis (SNA) and motivates a call for prescriptive models in this field. Section 3 provides an overview of social network theories and defines relevant network structure measures that quantify social structure. Section 4 discusses the relation between work-related outcomes and social structure measures. Section 5 gives a formal statement of a special-case non-trivial instance of TFP-SS and studies this instance in greater detail. Section 6 presents LK-TFP algorithm. Section 7 reports experimental results on TFP-SS instances of varied sizes with undirected and directed networks. Section 8 concludes the paper and discusses future research directions.

2. Emerging prescriptive research in SNA

The science of SNA encompasses a set of techniques for modeling network-based systems (see Wasserman and Faust [55] for SNA motivation and position statement). These techniques range from studying centrality measures [13] to building complex probabilistic models describing network structure and formation [4,46,5]. More recently, the domain of SNA has attracted the attention of exact science professionals whose expertise allowed for advances in modeling interactions between agents [18,42,53].

The main deficiency of the existing SNA tools is that they mostly offer descriptive insights [41], rather than prescriptive capabilities. The dearth of models that could allow a decision-maker to optimally change/influence a social network structure accentuates the difficulty in handling such tasks, and at the same time, calls for filling this gap. The existing works in the area of optimization and prediction are notable [54,37,12], however, they have focused on small, highly constrained tasks as opposed to introducing broad classes of problems and general methodologies for addressing them. Of such prescriptive efforts, the models for finding subsets of influential individuals in networks are the most studied [33,28,6].

There exist models that incorporate such graph-based measures as network diameter, density, and centrality, into TFP. However, again, most of these studies are descriptive and focus on impacts of social relations, expressed by SNA measures, on team performance [10,40,1,15]. Existing prescriptive models considering a team's social network use little information captured in the social network structure. Basic SNA concepts such as closeness, diameter, and minimum spanning tree have been employed in identifying a team of experts so as to minimize intra-team communication costs [36,20,52], and in some cases, individual member costs [32,31].

In 2003, in a study of 816 organization founding teams, Ruef et al. showed that homophily and network constraints are the key factors defining team composition [48]. In a more recent study of 2349 open-source software (OSS) development teams, Hahn et al. reported positive correlations between the developers' decisions to join project teams, the collaborative ties with project initiators and the perceived status of other (non-initiator) members [29].

Zhu et al. investigated the impacts of personal and dyadic motives on team formation [60]. They used Exponential Random Graph modeling to find that individuals first get interested in a project due to personal motives such as self-interest, mutual interest, collective action and coordination cost. The typical secondary considerations include dyad-based considerations explained by the social theories of homophily, swift trust, social exchange and co-evolution.

TFPs have been formulated in multiple fields of practice and attacked by researchers in diverse disciplines. For instance, the problem of assigning students of a class to different projects can be framed as a TFP [22]. Kim et al. presented a Parallel Balanced Team Formation (PBTF) problem and employed MapReduce to solve PBTF variations considering the diversity of team members' skills including task-handling and communication abilities [34]. Agrawal et al. presented *MaxTeam* and *MaxPartition* to model problems in a similar setting [2]. An integer programming approach was

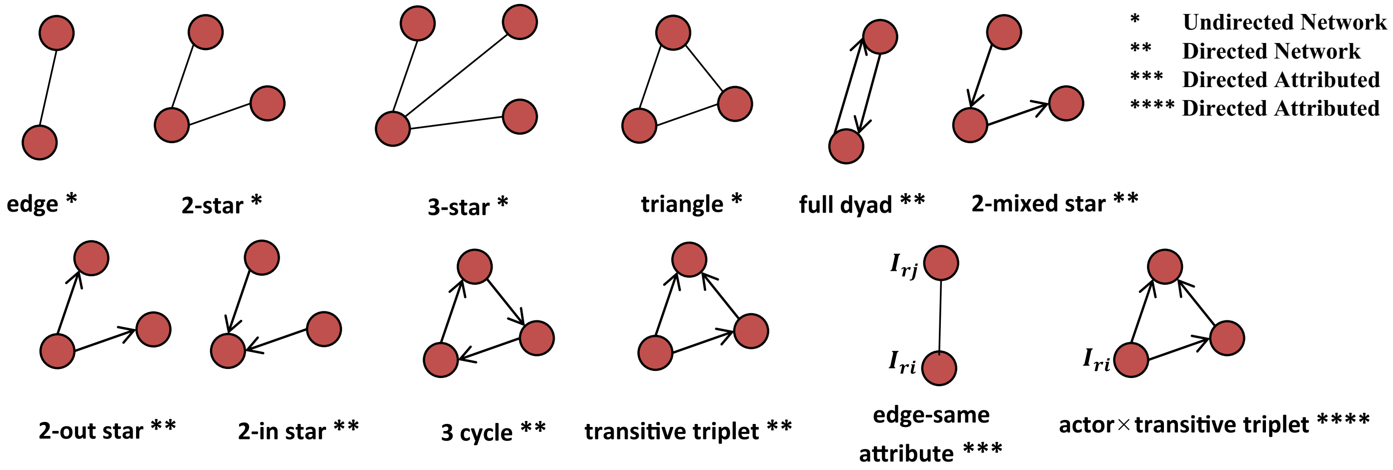


Fig. 2. Examples of basic network structures for undirected, directed and attributed networks; labels of the form (∗) indicate network types. The two bottom-right network structures consider nodal attributes in undirected and directed networks. For instance, the "edge-same" and "actor ×transitive triplet" structures capture the simultaneous impact of the social relationships (represented by the edge and the transitive triplet structures, respectively) and nodal attributes on tie formation. In an attributed graph, binary variable $I_{ri} = 1(0)$ if node i has (does not have) r-th attribute (see [53]).

developed to form student teams for case studies in the integrated core courses at Indiana University [19].

Another example of solving TFP for real-world applications is grouping the employees of Eli Lilly and Company (Lilly) as they participated in paid volunteer projects, in teams, to serve communities in impoverished countries. Volunteer teams are selected from a large pool of applicants, where a traditional selection process often fails to consider the employees' preferences and interpersonal relationships [39]. With the rapid growth of online social networks, grouping people into working teams becomes a new fruitful area for TFP applications. Awal and Bharadwaj adapted TFP to identify a set of experts from an expertise-focused social network that can collaborate effectively to accomplish a given task [7].

Given the qualitative evidence of network effects on team success, there is much value in conducting rigorous quantitative research on team formation. This paper makes the first effort to introduce a comprehensive framework for TFP, based on social network theories. The ability to quantify social network structure is the key to this effort.

3. Social science theories and network structure measures

This section motivates the formulation of a TFP that explicitly incorporates and quantifies the goodness of Social Structure within formed teams: this problem is referred to as TFP-SS. The section touches upon the existing social science theories and ways to quantify social network structure.

Network structure measures (NSM) are the key tools used in the presented framework to construct rigourous, closed-form functions of social structures in TFP. Earlier studies of the behavior of connected individuals sought for social theories that could explain network formation mechanisms [18,46,53]. These efforts resulted in the use of network structures, i.e., simple geometric constructs corresponding to the underlying social theories, in mathematical modeling. Prior to establishing how these social network theories can be useful for explaining outcomes in TFP, some definitions and notation are in order.

3.1. Definitions and notation used to quantify social structure

Let $\mathcal{G}(V, E)$ be a (un)directed graph with a set of nodes V, $|V| = N$, and a set of edges (arcs) E, represent a social network of individuals. With the notation v_i used for node i, $i = 1, \ldots, N$, set $e_{ij} = 1(0)$ if there exists (does not exist) an edge between nodes i and j. Let w_{ij} denote the weight of an edge, which indicates the strength of a social tie. Define $N_{\mathcal{G}}(v_i)$, $i = 1, \ldots, N$, as the local network of node v_i, i.e., the set of all the neighbors of (the nodes adjacent to) i in $\mathcal{G}$. Define M as the total number of teams and $X_j \subset \mathcal{G}$, $j = 1, \ldots, M$, as the network of the members of team j, with $|X_j| = n_j$ as the number of members in team j. Let $N_{X_j}(v_i)$, $i = 1, \ldots, N$, $j = 1, \ldots, M$, denote the local network, also known as the ego network (Everett and Borgatti 2005), of node i in team j. An individual represented by node i in $\mathcal{G}$ is said to belong to team j, $v_i \in X_j$, if node i is contained in X_j. In an attributed graph, set the binary variable $I_{ri} = 1(0)$, $r \in \mathcal{A}$, if node i has (does not have) the rth attribute (e.g., certain expertise or ability), where set $\mathcal{A} = 1, 2, \ldots, A$ contains the indices of the attributes relevant to a problem at hand.

According to earlier works in other SNA applications, the meaningful part of a social environment (climate) in a community is captured by the community's social network. Social scientists theoretically explore the connections between individuals in a social network, which in turn, can be expressed in a graph via basic network structures (see Fig. 2). Using the introduced notation, a full dyad (also known as reciprocal tie) involving nodes i and p is the structure where $e_{ip} = 1$ and $e_{pi} = 1$. Similarly, in a directed graph, a triplet of nodes i, j and k is a three-cycle whenever $e_{ip} = e_{pq} = e_{qi} = 1$; in an undirected graph, such triplet is simply called a triangle.

Network structure measures (NSMs) are functions of network structures capturing the tendencies highlighted by social theories; they can also be viewed as properties of social network graphs. For instance, the number of reciprocated ties measures the tendency for reciprocity in a community [53].

Let $\mathcal{F}_l(N_{X_j}(v_i))$, $l \in \mathcal{L}$, denote the lth NSM in the local network of node i in team j, where $\mathcal{L}$ is the set of indices of network structure types of interest to a researcher. For example, the number of edges in a local network of node v_i is found using the respective NSM as $\mathcal{F}_{edge}(N_{X_j}(v_i)) = \sum_{p \in X_j, p \neq i} e_{ip}$.

Social science theories and their respective NSMs are explored next to explain how one can interpret the observed NSM quantities in a team or an individual's local network.

3.2. Social science theories for interpreting team network structure

There are several theories related to social networks that may

Table 1
Social network theories, the corresponding measurable social structures, and their impact on particular aspects of team work.

Social network theory	Social structures	Impact on teams
Social exchange	Full dyad	Cooperation
Homophily	Ego-alter	Individual's attributes
Transitivity	Triangle, k-star, edge	Information sharing in team
Contagion	k-star	Leadership
Network evolution	k-star	Individual's attributes
Structural holes	Triangle	Personal performance

explain relations between the social structure within a team and the team's work-related outcome. Network theories rooted in social science elucidate the creation, maintenance, dissolution, and reconstitution of organizational networks [18], and also, interpret social structures from communication and individual attributes perspectives. The theories relevant in the team formation context include (1) social exchange, (2) homophily, (3) transitivity, (4) contagion, (5) network evolution, and (6) structural holes theories. Their connection with the network structures quantifiable by NSMs can be established as follows.

The *social exchange theory* states that the inclination to have a communication tie from individual A to individual B is predicated on the presence of a communication tie from individual B to individual A [18,60]. The main concern of exchange and dependence theories is the mutual relationships between pairs of network actors, called reciprocity. Since reciprocity facilitates information, knowledge and experience sharing between team members, it is an indicator of cooperation in the team. The number of full dyads in a local network of node v_i in team j can be expressed as $\mathcal{F}_{fulldyad}(N_{X_j}(v_i)) = \sum_{p\in X_j} e_{ip}e_{pi}$.

Homophily as a node level theory suggests that individuals with similar attributes are more likely to properly communicate with one another. This theory explicitly takes into account individuals' attributes such as gender, age, education, organization type, etc.; attributes such as professional skills, knowledge, intelligent, leadership skills, job satisfaction, problem solving skills, flexibility and motivation are vital factors in the team success [58,60]. Network structures pertaining to the homophily theory must incorporate individual attributes into models, with corresponding measures of the form $\mathcal{F}_{ego-alt_r}(N_{X_j}(v_i)) = \sum_{p\in X_j} e_{ip}e_{pi}I_{ri}I_{rp}$, $r\in\mathcal{A}$.

Transitivity is an important factor impacting team outcome due to the role of information flow and communication between team members. This theory stresses an inclination toward consistency in relationships within a community, and hence, expects better-functioning teams to exhibit higher levels of transitivity [18]. Different triplet type-based NSMs inform a researcher of different variations of transitivity [46]. As an example, the number of three-cycles in a weighted graph can be expressed as $\mathcal{F}_{wightedtriangle}(N_{X_j}(v_i)) = \sum_{p,q\in X_j,p\neq q\neg i} w_{ip}w_{qi}w_{pq}e_{ip}e_{pq}e_{qi}$.

The *contagion theory* focuses on the tendency to “follow the crowd” in social networks. Detected by the prominence of k-star structures, the tendency indicates the popularity of certain individuals in a network. In directed networks, k-in star and k-out star structures illustrate the level of popularity. The presence of these social structures implies that individuals with higher in-degree, or higher outdegree, are more attractive to individuals looking to form new ties [53]. In the team formation context, high contagion signals strong team core, team cohesiveness, but may also indicate over-reliance of a team on a single individual in performing tasks. Individuals with high degree of popularity also help to maintain an effective advice network within a team and facilitate the spread of information [14]. The number of k-stars can be computed as $\mathcal{F}_{k-star}(N_{X_j}(v_i)) = \sum_{p\in X_j,i\neq 1,\dots,k} e_{i1}e_{i2}\dots e_{ik}$ ($\forall\ k = 2, \dots, K$ and $K \le n_j - 1$).

The *network evolution theory* states that social networks are dynamic, which means that ties emerge and change over time [53,60]. These relational changes may be viewed as a function of the existing social structure in a network. This idea implies that all nodes in the network act to increase their personal utility, or “happiness”. The relevance of this theory for TFP is conceptual, since it motivates the consideration of local networks in team performance studies.

The *theory of structural holes* [47] argues that the shape of a local (ego-centered) network influences the amount of information that the ego-node receives. As a result, the ego is supported by more non-redundant information at any given time, which provides the ego with the capability of performing better or being perceived as the source of new ideas [14]. A network position where an ego benefits from the information flow within a team is called a structural hole [47]: the abundance of structural holes in a network can be assessed by counting the k-star and triangle structures in it.

Models based on the theories summarized in Table 1 have been previously used in other SNA applications, particularly in network formation studies. A large branch of SNA literature develops Exponential Random Graph Models [46]. Stochastic actor-based models introduced by Snijders et al. [53] are also widely used; they also successfully utilize network structures based on individuals' attributes. Snijders et al. [53] were the first to focus research attention on individuals' local networks. Following the same logic, this paper primarily considers TFP-SS instances where aggregate, additive utility of team members is maximized.

4. Expressing work-related measurements using NSMs

The objective of TFP is to optimize some aggregate measure of team performance. An integral component of a TFP-SS framework should relate a team outcome (i.e., performance) with measurable elements of the social structure, as discussed in Section 3. Social network theories motivate and justify the use of NSMs in quantifying social structure. The next step is to establish an explicit relation between NSMs and team performance (e.g., using available problem-specific data). As noted in Section 2, grouping students of a class into teams serves a good example of TFP. Importantly, the success of the teams may highly rely on the nature of social connections between students in each team. Therefore, quantifying the structure of social connections using the relevant NSMs (as illustrated with the Zackary Karate Club example in Section 7) has much value. Such NSM-based quantification can be exploited in the same way in which the less granular social network measures, including the network diameter and centrality measures, had been incorporated into the problem formulations in the earlier TFP literature [1,36].

According to the theory of structural holes, in many situations, the work-related outcome of each team can be represented as a function of the NSMs computed over the team members' local networks. Note that in general, relying exclusively on local networks in TFP-SS performance computations may be incorrect (e.g., this approach is not valid for evaluating consulting teams). However, with nursing, rescue, and police teams, the consideration of local networks can certainly be justified.

In most real-world applications, data of prior individual performance of team members is recorded and can be accessed. Therefore, performance function $\mathcal{P}(X_j) = \sum_{i=1}^{n_j}\mathcal{H}(N_{(X_j)}(v_i))$ can be approximated using any general parametric models and parameter estimation techniques.

There exists a variety of techniques that can be used to estimate

$\mathcal{H}$ from empirical data: regression, spline interpolation, neural networks, and machine learning, among others. Given the abundance of available literature on this topic (including the papers referenced above), this paper will not focus on such methods in detail. Note only that in certain situations, the dependencies between team member outcomes must be taken into account, in which case simple methods such as linear regression may not work [5], and more complicated estimation techniques must be explored.

Assume that each individual's outcome is recorded, and also, the information of their local network structure is available. For illustrative purposes, assume that $\mathcal{H}(X_j)$ can be expressed as a linear function of NSMs computed over each team member's local network (in team j). Although the linearity assumption may reduce the accuracy of the model, as discussed, it offers a simple way to aggregate social structures and estimate the overall work-related outcome for team j,

$$\mathcal{P}(X_j) = \sum_{i=1}^{n_j} \sum_{l \in \mathcal{L}} \theta_l \mathcal{F}_l(N_{X_j}(v_i)). \tag{1}$$

Recall that the NSMs allow one to focus on individual local networks. In (1), θ_l is the weight quantifying the contribution of the network structure, i.e., the strength of its corresponding theory, represented by a corresponding NSM: each such weight should be estimated using the available data. Thereafter, a TFP-SS instance can be formally stated.

5. TFP-SS formulation

Consider the problem of partitioning a group of N individuals embedded in a social network, represented by graph $\mathcal{G}(V, E)$, into M teams so as to achieve the best outcomes across all the teams. The TFP-SS encompasses a variety of models; to specify a problem instance, a researcher must consider the following modeling components.

Individual or team network: The objective is to optimize a criterion function over all teams considering individuals' local networks or the teams' networks. This distinction should be made based on the adopted outcome evaluation approach. As such, the team's network may be preferred for forming consultant teams; on the other hand, the theory of structural holes and network evolution theory support the idea of using individuals' local networks in problems where the team outcome is the sum or the averages of the individual member outcomes [60].

Set of NSMs: As a part of formulating TFP-SS, a researcher should select a set $\mathcal{L}$ of network structure types, corresponding to social network theories relevant in the problem setting. For example, individual attributes as well as the corresponding NSMs may be less important in certain applications.

Objective function: With the outcome of each team evaluated as in (1), a researcher should define a proper objective function. In measuring team outcome, one might be interested in the following objectives (among others):

(1) optimizing the average outcome across all teams,

$$\max_{X_j} \left\{ \frac{\sum_{j=1}^{M} \mathcal{P}(X_j)}{M} \right\},$$

(2) maximizing the outcome of the weakest team,

$$\max_{X_j} \{ \min_{j=1,\ldots,M} \mathcal{P}(X_j) \},$$

(3) minimizing the absolute deviation from the average outcome,

$$\min_{X_j} \left| \mathcal{P}(X_j) - \frac{\sum_{j=1}^{M} \mathcal{P}(X_j)}{M} \right|,$$

(4) minimizing the squared deviation from the average outcome,

$$\min_{X_j} \left[\mathcal{P}(X_j) - \frac{\sum_{j=1}^{M} \mathcal{P}(X_j)}{M} \right]^2.$$

These objective functions highlight the different aspects which can be important to the decision-maker formulating their team formation problem, and also, correspond to the different ways in which aggregate team performance measures can be expressed via individuals performance measures; depending on an application, one can modify these objectives or introduce alternative ones. For instance, Baker et al. present a thorough overview of the methods useful to construct well-balanced student teams [8]. In many applications, multiple criteria play a role in quantifying the goodness of a team. In this case, the TFP can be treated as a multi-objective optimization problem taking into account, e.g., costs, delivery times, social structure, etc. The proposed objective functions can then be extended to binary or categorical ones (see Bergey and King [11] for more details about discrete team performance measurement).

Network type: Networks with different types of edges, e.g., (un) directed and weighted, may lead to different models. For instance, 2-out stars are not even defined on undirected networks.

The goal of this paper is to present a general methodology for treating TFP-SS problems. In order to illustrate the application of the resulting framework, a special case instance of TFP-SS is considered in detail.

5.1. TFP-SS: a special case instance

The presented framework is designed to generate and treat TFP-SS models using NSMs. In order to investigate the tractability of the resulting models, this section first considers a non-trivial TFP-SS Special case instance (TFP-SSS). The TFP-SSS is defined on an undirected, unweighed graph, representing a social network of individuals with identical skills. The choice of network structure types included for modeling TFP-SSS is limited to edge, 2-star, 3-star and triangle, $\mathcal{L} = \{edge, 2-star, 3-star, triangle\}$ – these social structures are most common in network formation modeling.

In the ensuing computational study, the experiments with TFP-SS instances based on directed networks are also included, so as to demonstrate the comprehensiveness and flexibility of the presented framework: the instances with directed networks use full dyad, 2-in star, 2-out star, 3-cycle and transitive triplet NSMs.

As stated, TPS-SSS is a realistic problem relevant for assembling professional teams (e.g., nursing, rescue, police, security, sport, etc.), where a minimum level of expertise is uniformly required of all team members. Such teams usually complete tasks under stressful conditions, and the effectiveness of working within a team structure is more important in this case than small differences in individual qualification.

In TFP-SSS, the average outcome of teams is maximized based on the individuals' local networks, which is mathematically equivalent to maximizing the sum of the outcomes over all the teams,

$$\max \sum_{j=1}^{M} \mathcal{P}(X_j). \tag{2}$$

It should be noted that the local network of any given node

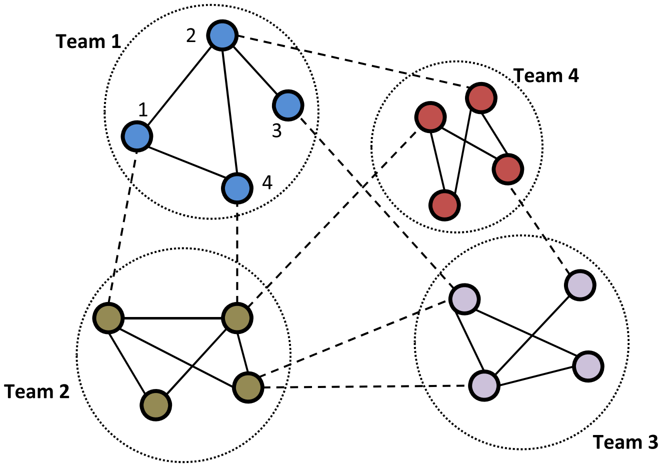


Fig. 3. Four teams in the given social network.

includes the node itself, its immediate neighbors and the links between them all. To visualize a particular instance of TFP-SSS, consider a social network of size 16 in Fig. 3.

Quantifying the outcomes of four teams of size four amounts to computing NSMs as illustrated in Table 2 for nodes in Team 1. The described instance of TFP-SSS is formally stated:

Instance: A graph $\mathcal{G}(V, E)$, $|V| = N$; $n_j \in Z^+$ for $1 \le j \le M$; a partition of disjoint sets $X_1, X_2, \ldots, X_M$, where $X_j \subseteq \mathcal{G}(V, E)$ for $1 \le j \le M$ and $\theta_l \in R$ for $l \in \mathcal{L}$.

Question: Is there a partition of V into M disjoint subsets $X_1 \cup X_2 \cup \ldots \cup X_M$, with $|X_j| = n_j$, such that $\sum_{j=1}^{M} \mathcal{P}(X_j)$ is maximized, where $P(X_j) = \sum_{i=1}^{n_j} \sum_{l \in \mathcal{L}} \theta_l \mathcal{F}_l(N_{X_j}(v_i))$ for $1 \le j \le M$ and $\sum_{j=1}^{M} n_j = N$?

The following theorem states that TFP-SSS is NP-hard.

Theorem 1. *TFP-SSS with M teams to be formed out of N individuals is NP-hard.*

Proof. The presented TFP-SS instance is NP-hard by polynomial time reduction from Partition into Triangles (see the Appendix). □

The number of all feasible solutions in TFP-SSS is

$$\Gamma(N, M) = \binom{N}{n_1}\binom{N - n_1}{n_2}\ldots\binom{N - \sum_{j=1}^{M-1} n_j}{n_M},$$

with $n_1 + n_2 + \cdots + n_M = N$. Quantity $\Gamma(N, M)$ shows how quickly the number of the feasible solutions grows, as N and M increase. Since TFP-SSS is NP-hard, one cannot expect to obtain an exact algorithm to solve it in polynomial time. However, for small size problems, an exact method can be designed to search for an optimal solution. The next section presents an Integer Programming (IP) formulation of TFP-SSS.

5.2. An IP formulation of TFP-SSS

Define a set of decision variables for TFP-SSS:

Table 2
Network structure measure values for nodes in Team 1.

Node	No. of edges	No. of 2-stars	No. of 3-stars	No. of triangles
1	3	3	0	1
2	4	3	1	1
3	1	0	0	0
4	3	3	0	1

$$y_{ij} = \begin{cases} 1 & \text{if node } i \text{ is assigned to team } j,\ i = 1, \ldots, N \text{ and } j = 1, \ldots, M, \\ 0 & \text{otherwise,} \end{cases}$$

where $\mathcal{I} = \{1, \ldots, N\}$ and $\mathcal{J} = \{1, \ldots, M\}$. The objective function of TFP-SSS is nonlinear since it includes the products of the decision variables,

$$\max \sum_{j=1}^{M} \sum_{i=1}^{N} \Bigg(\theta_1 \sum_{p=1}^{N} e_{ip} y_{ij} y_{pj} + \theta_2 \sum_{p=1}^{N} \sum_{q=1}^{N} e_{ip} e_{iq} y_{ij} y_{pj} y_{qj} + \theta_3 \sum_{o=1}^{N} \sum_{p=1}^{N} \sum_{q=1}^{N} e_{io} e_{ip} e_{iq} y_{ij} y_{oj} y_{pj} y_{qj} + \theta_4 \sum_{p=1}^{N} \sum_{q=1}^{N} e_{ip} e_{pq} e_{qi} y_{ij} y_{pj} y_{qj} \Bigg). \tag{3}$$

To linearize (3), variables w_{iopqj}, z_{ipqj} and x_{ipj} are introduced to replace the terms $y_{ij}y_{oj}y_{pj}y_{qj}$, $y_{ij}y_{pj}y_{qj}$, and $y_{ij}y_{pj}$, respectively. An integer programming formulation for the TFP-SSS is given,

$$\max \sum_{j=1}^{M} \sum_{i=1}^{N} \Bigg(\theta_1 \sum_{p=1}^{N} e_{ip} x_{ipj} + \theta_2 \sum_{p=1}^{N} \sum_{q=1}^{N} e_{ip} e_{iq} z_{ipqj} + \theta_3 \sum_{o=1}^{N} \sum_{p=1}^{N} \sum_{q=1}^{N} e_{io} e_{ip} e_{iq} w_{iopqj} + \theta_4 \sum_{p=1}^{N} \sum_{q=1}^{N} e_{ip} e_{pq} e_{qi} z_{ipqj} \Bigg), \tag{4}$$

st:

$$\sum_{i=1}^{N} y_{ij} = n_j \quad \forall\, j, \tag{5}$$

$$\sum_{j=1}^{M} y_{ij} = 1 \quad \forall\, i, \tag{6}$$

$$w_{iopqj} \ge y_{ij} + y_{oj} + y_{pj} + y_{qj} - 3 \quad \forall\, i, o, p, q, j,\ o \ne p \ne q, \tag{7}$$

$$w_{iopqj} \le \frac{y_{ij} + y_{oj} + y_{pj} + y_{qj}}{4} \quad \forall\, i, o, p, q, j,\ o \ne p \ne q, \tag{8}$$

$$z_{ipqj} \ge y_{ij} + y_{pj} + y_{qj} - 2 \quad \forall\, i, p, q, j,\ p \ne q, \tag{9}$$

$$z_{ipqj} \le \frac{y_{ij} + y_{pj} + y_{qj}}{3} \quad \forall\, i, p, q, j,\ p \ne q, \tag{10}$$

$$x_{ipj} \ge y_{ij} + y_{pj} - 1 \quad \forall\, i, p, j, \tag{11}$$

$$x_{ipj} \le \frac{y_{ij} + y_{pj}}{2} \quad \forall\, i, p, j,$$

$$y_{ij}, x_{ipj}, z_{ipqj}, w_{iopqj} \in \{0, 1\}. \tag{12}$$

where θ_l, $l \in \mathcal{L}$, and e_{ip} are known parameters and $o, p, q \in \mathcal{I}$. The constraints in (5) ensure that team j, $j = 1, \ldots, M$, has exactly n_j members (alternatively, these equality constraints can be replaced by upper bound constraints). The constraints in (6) ensure that every individual is assigned to exactly one team. The constraints in (7)–(12) are needed to linearize the model. The nonlinear term, $y_{ij}y_{oj}y_{pj}y_{qj}$, is replaced with the binary variable, w_{iopqj}, with the additional constraints guaranteeing that $w_{iopqj} = y_{ij}y_{oj}y_{pj}y_{qj}$: since w_{iopqj} is one if and only if $y_{ij} = y_{oj} = y_{pj} = y_{qj} = 1$ and zero otherwise, constraints (7) and (8) are used to satisfy such conditions. For example, suppose $y_{ij} = y_{oj} = y_{pj} = y_{qj} = 1$, then constraint (7) turns into $w_{iopqj} \geq 1$ and constraint (8) turns into $w_{iopqj} \leq 1$, ensuring that $w_{iopqj} = 1$. Moreover, if one of the variables takes on zero, then one has $w_{iopqj} \geq 0$ by (7) and $w_{iopqj} \leq \frac{3}{4}$ by (8), and hence, $w_{iopqj} = 0$. Similarly, constraints (9)–(12) guarantee that $z_{ipqj} = y_{ij}y_{pj}y_{qj}$ and $x_{ipj} = y_{ij}y_{pj}$. Finally, all the decision variables are set to be binary. Note that the formulations for the TFP-SS instances defined on directed networks can be constructed in a way similar to the presented TFP-SSS formulation.

Observe that while the model is linear/integer, the number of constraints in it quickly increases with the problem size. For N individuals to be assigned to M teams, the IP has $M(N^4 - 5N^3 + 9N^2 - 4N)$ variables and $2M(N^4 - 5N^3 + 9N^2 - 5N) + M + N$ constraints. For example with N=30 and M=5 (a small instance at first glance), the IP has 3,414,900 variables and 6,829,535 constraints; however, one can still expect that exact IP algorithms are able to find optima for small-size problems. Note that in most real-world applications, the number of individuals to be managed is more than 50 (e.g., think of a nursing department in a typical health care center). Hence, an efficient sub-optimal algorithm is in order for dealing with such instances as well as other versions of TFP-SS.

6. Variable depth neighborhood search algorithm for TFP-SS

This section presents an efficient algorithm for TFP-SS, called LK-TFP. The idea of variable depth neighborhood search is well recognized for its success in treating Traveling Salesman Problem (TSP). Proposed by Lin and Kernighan [38], the idea was revised and implemented in an exceptionally effective heuristic Lin–Kernighan–Helsgaun (LKH) for symmetric TSP by Helsgaun [30]. LKH algorithm performs λ-opt sequential moves, where in each step, λ links in the network representing the current solution are replaced by other λ links [30]. The variable depth neighborhood idea based on λ-opt search has not found success in applications beyond TSP and vehicle-routing problems [35,49]. Being similar to TSP in that the team assignments can be visualized as a Hamiltonian tour, TFP-SS appears to be a suitable problem for implementing the sequential move idea.

6.1. LK-TFP algorithm for TFP-SS

In LK-TFP, one feasible solution is within a λ-move from another if such a move improves the objective function value by exchanging λ individuals across different teams (see Fig. 4), similar to replacing λ links in TSP. The incremental improvements, i.e., multiple λ-moves, lead to a local optimum.

Updating the objective function of TFP-SSS is a computationally expensive task. A naive implementation re-computes the objective function using $\mathbf{O}(n^4)$ operations: this effort is spent mainly in recounting the most complex network structure measures in a graph (e.g., 3-stars in TFP-SSS).

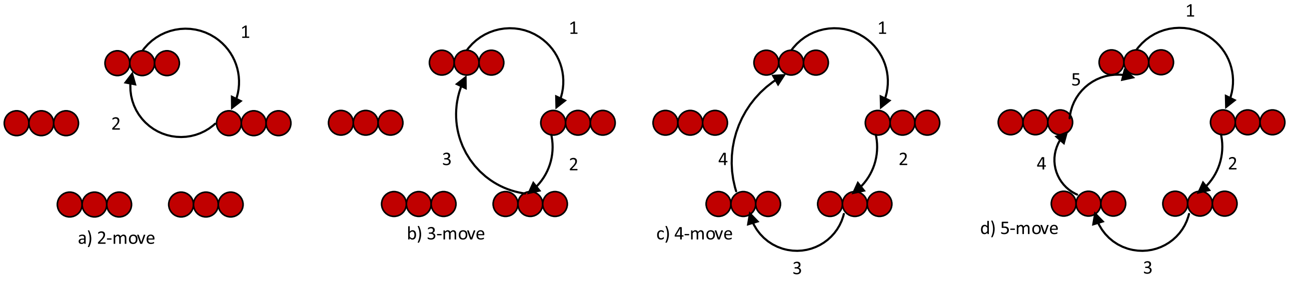
The idea of a branching algorithm for TFP-SS is to build a tree of solutions to be traversed (see Fig. 5). The solutions at the nodes of the tree are obtained by executing λ-moves. LK-TFP traverses the tree to arrive at such a transitive solution that improves the objective function, signifying the completion of a current sequential move. LK-TFP identifies good branches of the tree and avoids visiting too many non-improving solutions by cutting off the search space. To describe the tree traversal in detail, some terminology is required. Level $s \in S$ of the tree encompasses all the solutions reachable by a single s-move from the initial solution at the root of the tree (where a $s - move$ is a $\lambda - move$ with $\lambda = s$). We will use notation S to denote a set of search tree levels.

In the tree, the root node (a single node at level 1) represents a feasible solution of TFP-SS, which can be either a random initial solution or a current best known solution, necessarily feasible. The internal tree nodes (at level $s \in S$) represent solutions resulting from s-moves performed on the root solution. Importantly, each internal node also stores an incomplete, i.e., infeasible, solution. Leaf nodes (at the lowest possible tree level) are those where the algorithm must stop the search because of the pre-set branching rules (e.g., any team can have only one member participating in the same move). A path along two or more nodes represents a directed correspondence between two or more solutions.

Let ego_{sm} denote the individual that has been removed from a team, perturbing the solution at node $m \in J_s$ of level $s \in S$ of the tree, where $|J_s|$ and $|s|$ stand for the number of nodes at level s of the tree and the number of levels in the tree, respectively. An individual is called a friend of the ego if there exists a social (i.e., network) link between them. A set of ego_{sm}'s friends is denoted by F_{sm}, $m \in J_s$, $s \in S$. When ego_{sm} is moving from team j' and joining team j'' $(j'' \neq j')$, then one of team j'''s current members who is not friends with ego_{sm} must leave team j''. The candidates for leaving the team form a set denoted by L_{sm}.

The algorithm starts by branching from the root, selecting individuals one-by-one as an ego; essentially, each ego attempts to join their friends in other teams. If no such friends are found, two individuals within the smallest distance from the ego (with the distance measured as the length of a shortest path in the social network) are selected. Then, the leaving individuals are determined. For each leaving node in L_{sm}, a branch is added, pointing at a new node at the next level of the tree.

For instance, suppose that ego individual 1 is on team A, with its friend individuals 4 and 8 on teams B and C, respectively. As individual 1 is added to team B or C, with some individuals who are not friends with individual 1 leaving that team, a branch is added to the tree. Indeed, for each leaving individual, a corresponding branch is created in the search tree. As the potential leaving individuals join other teams (in turn), branches are added to the lower levels of the tree. Fig. 5 depicts the structure of the corresponding search tree. At level 0, individual 5 or 6 (7 or 9) should leave the team if individual 1 joins team B (C). Level 1 has four nodes in the tree corresponding to each scenario, depending on which individuals are leaving (being replaced). Assuming that individuals 3 and 7 are friends with individual 5, Level 2 is constructed by having individuals 1 and 2 leave team A and having



a) 2-move b) 3-move c) 4-move d) 5-move

Fig. 4. λ-move for λ=2, 3, 4, 5.

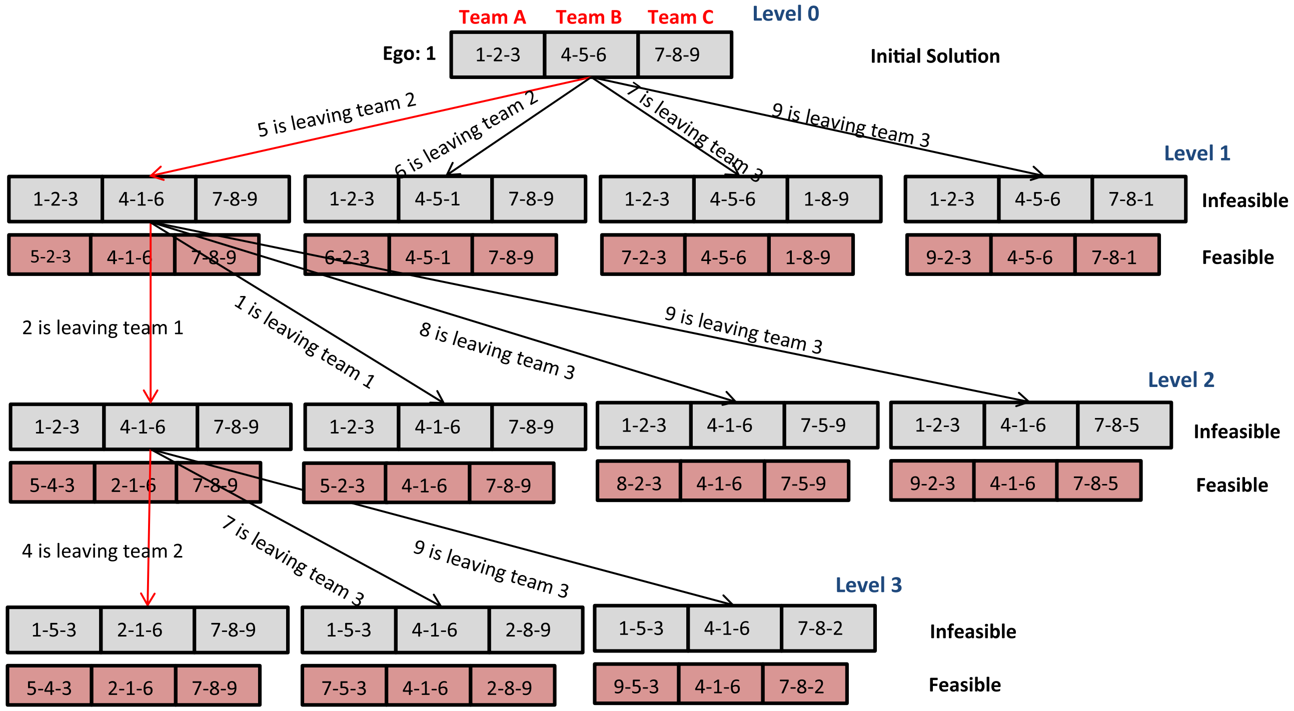


Fig. 5. General Search tree structure of LK-TFP. The algorithm sequentially traverses the infeasible solution space until one infeasible solution with an acceptable (improving) feasible counterpart is found. The red arrows show the depth first search (DFS) strategy. (For interpretation of the references to color in this figure caption, the reader is referred to the web version of this paper.)

individuals 8 and 9 leave team *C*. It should be noted that individual 1 has already left team *A*, and hence, the corresponding branch is fathomed (i.e., completing a 2-move).

At every tree node, e.g., with individual *m* as an ego at level *s*, the algorithm has two solutions: one feasible and one infeasible. An infeasible solution is produced by replacing a leaving individual with a copy of the ego (thus creating a duplicate of the latter). A feasible solution is produced by "completing" the thus-initiated infeasible move, i.e., having the leaving individual replace the ego in the ego's original team (once this is done, the ego no longer has any duplicate). Whenever an improving feasible solution is found, the algorithm proceeds to a search for another improving λ-move, with the new best solution placed at the root of a new tree.

During a tree traversal, for any non-improving feasible solution, the gain of the corresponding infeasible solution is computed as follows:

$$g_{sm} = f(\mathbf{Y}_{s,m}) - f(\mathbf{Y}_{s-1,\mathcal{P}_m}), \tag{13}$$

where g_{sm} is the gain (in the objective function) resulting from the current *s*-move at tree node *m*. Subscripts $\mathcal{P}_m$ and $(s-1)$ indicate the parent of node *m* and level $s-1$ in the tree. Define $\mathbf{Y}_{s,m}$ and $\mathbf{Y}_{s-1,\mathcal{P}_m}$ as $N \times M$ matrices representing the feasible solutions (i.e., the IP formulation of TFP-SSS) obtained at node *m* at level *s* and node *m*'s parent at level $(s-1)$, respectively, and $f(\mathbf{Y})$, in general, denotes the objective function value as in (2), and in the specific case of TFP-SSS, as in (3):

$$f(\mathbf{Y}) = \sum_{j=1}^{M}\sum_{i=1}^{N}\Bigg(\theta_1\sum_{p=1}^{N} e_{ip}y_{ij}y_{pj} + \theta_2\sum_{p=1}^{N}\sum_{q=1}^{N} e_{ip}e_{iq}y_{ij}y_{pj}y_{qj} + \theta_3\sum_{o=1}^{N}\sum_{p=1}^{N}\sum_{q=1}^{N} e_{io}e_{ip}e_{iq}y_{ij}y_{oj}y_{pj}y_{qj} + \theta_4\sum_{p=1}^{N}\sum_{q=1}^{N} e_{ip}e_{pq}e_{qi}y_{ij}y_{pj}y_{qj}\Bigg). \tag{14}$$

At each level of the tree, $g_{sm} = f(\mathbf{Y}_{s,m}) - f(\mathbf{Y}_{s-1,\mathcal{P}_m})$ is the gain resulting from the exchange, and $G_s = \sum_{k=0}^{s} g_{km}$, is the total gain obtained with an *s*-move. The algorithm continues to branch as long as the (continuously updated) value of G_s is positive; otherwise, the tree node and all its subsequent branches are fathomed. Whenever a solution tree is completely traversed with no improving feasible solution found, the algorithm stops.

In the computational experiments reported in this paper, LK-TFP was implemented with the depth first tree search strategy, thus benefitting from more efficient memory usage. The algorithm's key steps are outlined in the pseudo-codes for Algorithms 1 and 2.

Algorithm 1. LK-TFP algorithm for TFP-SS.

```
public void main()
    Initialize $X_j, j = 1, \dots, M$ such that $\bigcap_{j=1}^{M} X_j = \varnothing$ and
    $\bigcup_{j=1}^{M} X_j = V$;
    Calculate Objective Function $\sum_{j=1}^{M} \mathcal{P}(X_j)$;
    for (t = 1; $t \le T_{max}$; t++)
        depthNeighborhoodSearch(); /* executing a tree
        based search*/
        if (bestSolutionValue $\le$ currentSolutionValue)
            recordBestSolution();    /* recording the best
            solution*/
        end if
    end for
end main
```

Algorithm 2. Depth neighborhood search.

```
public type depthNeighborhoodSearch()
    while (unvisitedNode())
        for (int $s = 0$; $s \le |S|$; $s$++) /* level $s$ */
            for (int m = 0; $m \le |J_s|$; m++) /* node $m$ at level $i$ */
                do
                    find $e_{sm} = v_q$, $F_{sm}$ and $v_p \in L_{sm}$;
            exchange $v_q$ and $v_p$; /* exchanging the ego */
            replace $v_q$ with $v_p$; /* replacing the ego */
            $g_{sm} = f(\mathbf{Y}_{s,m}) - f(\mathbf{Y}_{s-1,\mathcal{P}_m})$;
            $G_s = \sum_k^s g_{kj}$; /* calculate gain*/
            depthFirstSearch(); /* Depth First Search */
          while ($G_s > 0$)
        end for
        updateVisitedNodes(); /* update the list of the
        visited nodes */
    end
end while
```

In the presented version, LK-TFP was found to be very efficient in solving TFP-SS instances. A discussion of the reasons of such performance follows.

6.2. Performance analysis of LK-TFP

LK-TFP is remarkably efficient in solving TFP-SS, similar to LKH for TSP, and deserves a discussion of the reasons of such performance. Algorithms using λ-opt search usually face a serious drawback. In order to provably find an optimum, n-opt should be applied with large n, which is computationally infeasible in nontrivial problem instances.

LK-TFP avoids this difficulty by employing an intelligent search strategy. At each step, the algorithm checks for the necessity of increasing the value of λ in λ-opt moves and it considers a growing set of potential exchanges. If these exchanges improve the objective function and provide a better feasible solution, they are always accepted and λ increases by one. During a tree search, increasing λ is equivalent to going deeper into (the lower levels of) the tree. Starting with a feasible solution, the algorithm repeatedly executes the exchanges guided by the incremental gain g_{sm}, until the whole tree is traversed and the algorithm stops, or an improving feasible solution is reached, in which case the algorithm is restarted.

Additionally, the sequential exchange criterion [30] in LK-TFP enables internal nodes, which would otherwise be missed due to a premature fathoming, to be visited. In fathoming a tree branch, the positive gain criterion plays an important role. Using G_s as a branch cutting criterion has a significant impact on the algorithm efficiency. The gain consideration prevents the algorithm from reaching too deep into the tree and enables more rapid fathoming when no improvement results from the inter-team exchanges. Indeed, G_s is the summation of the gains obtained by traversing the tree from the root to the corresponding leaf which completes an s-move.

At the first glance, this stopping criterion may appear too restrictive. However, as long as an improving exchange (requiring no more permutations than the set search depth) exists, it is guaranteed to be discovered. To illustrate this point, suppose that an improved solution i_* can be found by a sequential permutation from a (current) solution i with a computationally intensive 10-opt search. The 10-opt move leading from i to i_* may not be necessary, however, if it contains an improving move with fewer permutations (fewer than 10) leading to i_*, for example, a 5-opt move. That is, a 5-opt search may eventually find the 5-opt move leading to i_*, albeit starting from a solution different from i. Lin and Kernighan proved this by showing that for a sequence of numbers with a positive sum, there exists a cyclic permutation of these numbers such that every partial sum is positive [30].

Consider Fig. 6, eliciting the relation of gain g_{sm} to the partial sum G_s for a five-opt sequential move. Suppose a current explored subsequence is negative in gain (e.g., $G_2 = -2 < 0$) resulting from move 2; if it is a part of an improving larger sequence, then this subsequence will be found later by traversing the tree from another node (starting from node 5). It means that an improving sequential move will be found. Furthermore, there are some other rules which are useful to increase the search efficiency. Recently performed exchanges are stored in memory for a small number of iterations. Moreover, LK-TFP uses the two best friend rule similar to the 5-neighbor rule in the original LK algorithm, which restricts the search to the five nearest neighbors (i.e., cities).

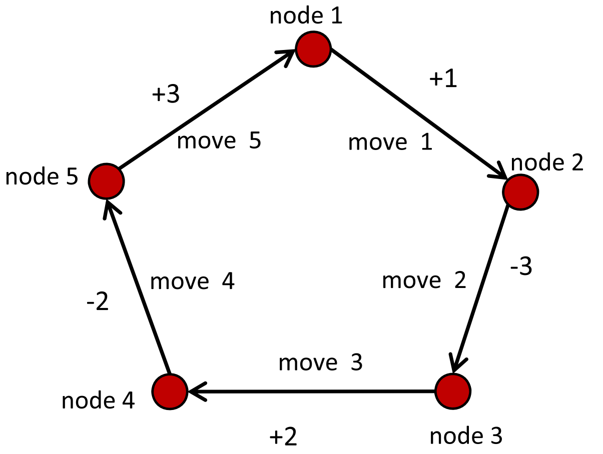


Fig. 6. An example of an improving sequence of moves starting at node 1.

The idea of this rule is to place best friends into same teams, where two friends are defined as best friends if they also have some other friend(s) in common. When leaving a team, an individual can be assigned to any team other than his current team. Now, our rule guides an individual towards the other teams which contain his best friends. If an individual has more than two best friends, in different teams, then one of these teams (to be explored first) is selected at random; with no best friends found, a team is chosen randomly among all the candidate teams. It is worth mentioning that one can think of other rules to choose teams, e.g., meeting the requirement of balanced teams in the student team formation problem.

This heuristic rule is based on expectations of which individuals are more probably teammates in the optimal solution. In other words, this rule treats the problem like a puzzle and attempts to complete it by placing right individuals in teams in each step. After each run, the best solution is placed in the root of tree and the search process continues. Results of LK-TFP for different size problems are reported in the next section.

7. Computational experiences with TFP-SS

This section explores the performance of LK-TFP on small-, medium- and large-sized TFP-SSS instances, including those incorporating directed networks. These problems are solved by both IP (using CPLEX) and LK-TFP (implemented in JAVA). Additionally, a Standard Genetic Algorithm (SGA) was implemented for additional benchmarking against a metaheuristic. The experiments were performed using a desktop with Intel(R) Core(TM)i5 3 GHz processor, 8 GB RAM and 64 bit operating system. LK-TFP was executed at least 30 times for each problem instance to explore the variability of the results over different runs, and thus, gauge the robustness of the algorithm. CPLEX was able to solve only small problems (with less than 20 nodes), with the memory limit being a key constraint.

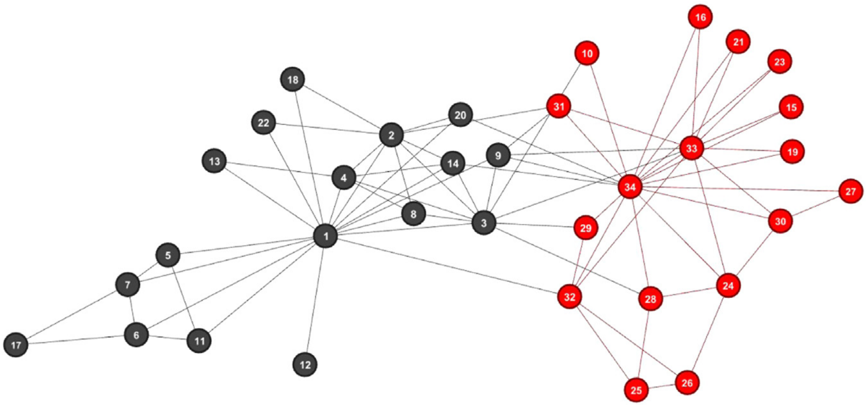


Fig. 7. Dividing ZKC network into two teams of size 17. (For interpretation of the references to color in this figure caption, the reader is referred to the web version of this paper.)

7.1. *TFP-SSS with undirected social networks*

A majority of existing social network datasets are undirected. TFP-SSS was solved with several real-world undirected networks including *Zachary Karate Club* (*ZKC*), *Florentine Families* (*FF*), *Kapferer Mine* (*KM*), *Taro Exchange* (*TE*), *Western Electric Employees* (*WEE*), *Thurman Office* (*TO*) and *Bernard* and *Killworth* (*BK*) (retrieved from http://vlado.fmf.uni-lj.si/pub/networks/data/) as well as with synthetic random networks. Without loss of generality, the weights θ_l, $l \in \mathcal{L}$, in (1) are set to be equal. The following list provides basic facts about the datasets used in this subsection:

(1) The ZKC network capturing a friendship network of 34 karate club members, were collected by observation at a US university over a 2-year period in the 1970s [27]. Fig. 7 depicts the ZKC network divided into two teams of size 17: node colors indicate team assignments.
(2) The FF network represents the social relations, including business ties and marriage alliances, among 16 Renaissance Florentine Families.
(3) The KM network connects 15 miners working on the surface in a mining operation in Zambia (then Northern Rhodesia). Fig. 8 presents the FF and KM networks divided into three and two teams, respectively: the nodes of the same color belong to the same team.
(4) The TE network represents the relations of gift-giving among 22 households in a Papuan village. Fig. 9(A) depicts the TE network divided into two teams.
(5) The WEE network captures the relations between 14 Western Electric (Hawthorne Plant) employees from the bank wiring room participating in horseplay. Using LK-TFP, the network is divided into 2 teams as shown in Fig. 9(B).
(6) The TO network is based on the informal interactions among 15 employees in an overseas office of a large international corporation. Fig. 10 shows an optimal assignment of TO network members into two teams.
(7) The BK network depicts interactions among 58 students living in a fraternity at a West Virginia college.

Table 3 summarizes the results of running LK-TFP and CPLEX across all the discussed networks.

Similarly, the results of both CPLEX and LK-TFP runs for the synthetically generated problems are presented in Table 4. The obtained results confirm that LK-TFP is effective and efficient in solving even large-scale TFP-SSS instances. As the results illustrate, IP techniques can handle only small problems (up to 16 individuals and five teams). For larger problems, more computational resources are required. For the instances with up to 30 individuals and four teams, CPLEX was not able to report any incumbent before getting out of memory. However, LK-TFP quickly identified optimal solutions for small problems. For medium- and large-sized problems, the runtime of LK-TFP is still remarkable. With large N and small M, LK-TFP does take longer to identify good solutions; naturally, the individuals' local networks get larger in such instances.

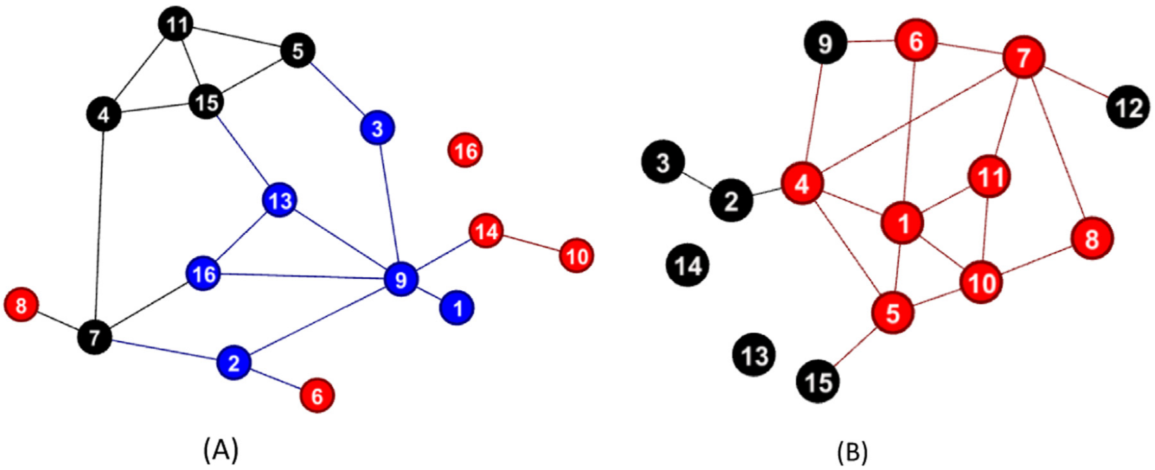


Fig. 8. Dividing (A) Florentine Families network into three teams and (B) Kapferer Mine network into two teams. (For interpretation of the references to color in this figure caption, the reader is referred to the web version of this paper.)

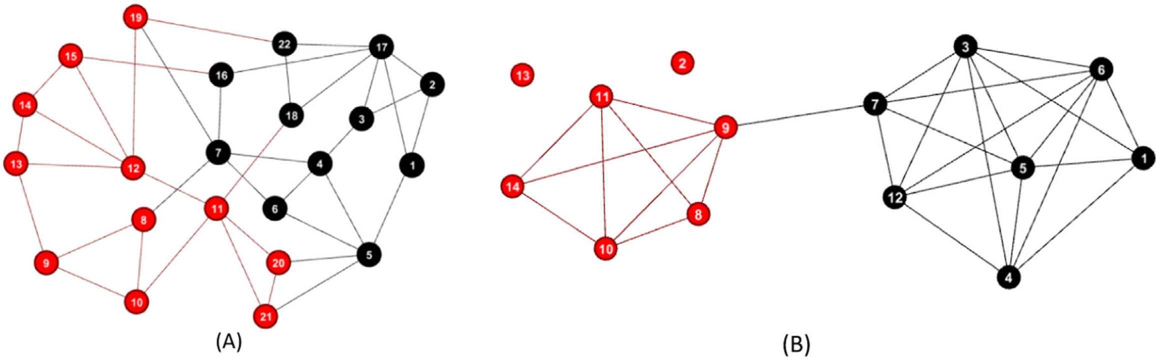


Fig. 9. Dividing (A) Taro Exchange network and (B) Western Electric Employees network into two teams.

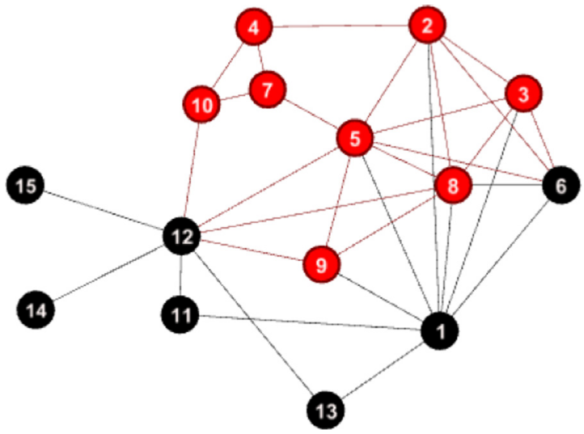


Fig. 10. Dividing Thurman Office network into two teams.

7.2. TFP-SS with directed social networks

There exist cases where the relationships between individuals in a social network are not necessarily bidirected. Incorporating the direction of relations into TFP-SS requires the use of other types of NSMs such as *full diad*, 2-out star, 3 cycle and *transitive triplet*. The directed TFP-SSS instances were created with *Dickson Bank Wiring* (*DBW*), *Thurman Organizational Chart* (*TOC*), *Succor Teams* (*ST*) (available on http://vlado.fmf.uni-lj.si/pub/networks/data/) and *Advogato* (*AD*) networks. The following list provides basic facts about the datasets used in this subsection:

(1) The DBW network is based on the data of 14 Western Electric employees helping each other with work. Fig. 11(A) shows how this directed network can be split into two teams.
(2) The TOC network represents formal relationships (organizational ties) between 15 employees of an international corporation. Based on this directed network, employees are assigned into two teams as depicted in Fig. 11(B).
(3) The ST network represents the relationship between 35 countries in terms of exporting succor players to each other. Fig. 12 shows how these countries can be divided into two subgroups (teams).
(4) The AD network is a sample of the trust network of Advogato with 500 users in an online community platform for software developers constructed in 1999 (KONECT: http://konect.uni-koblenz.de/networks/advogato).

Importantly, in the settings where many people are required to be grouped into working teams, e.g., in emergency situations or large projects, the scalability of computational methods becomes an issue. This section explores this issue with TFP-SS instances with directed networks. Table 5 summarizes the results of CPLEX and LK-TFP runs for the directed networks described above.

This trust network serves as a good testbed for exploring how one can adapt TFP-SS to find teams, the members of which prefer to be connected to trustworthy peers. Such formulation may be particularly useful when close cooperation is required within multiple teams in emergency situations with participating non-government organizations and multiple volunteers.

Table 3
Simulation results of LK-TFP and CPLEX runs across ZKC, FF, KM, TE, WEE, TO and BK networks (NA: not available).

Network	Teams	Optimum	LK-FTP solution			CPLEX solution	LK-TFP time (s)			CPLEX time (s)
			Min	Ave.	Max		Min	Ave.	Max	
ZKC	2	NA	872	874.8	877	NA	12	156.2	360	NA
ZKC	4	NA	103	123.75	141	NA	1	17.3	36	NA
ZKC	5	NA	218	230.7	239	NA	67	379.15	731	NA
FF	2	137	137	137	137	137	0.08	0.9	4	327
FF	3	96	96	96	96	96	1.3	2.95	8.5	511
FF	4	59	51	56.5	59	59	3.5	10.5	21	1630
KM	2	133	133	133	133	133	3.51	8.23	12.9	325
KM	3	72	65	71.6	72	72	2.15	6.4	18.5	616
TE	2	309	297	306.72	309	309	2.3	17.6	45	6048
TE	3	259	247	253.4	259	259	38.45	93.9	256.1	19,308
WEE	2	548	548	548	548	548	1.1	3.5	8.1	109
WEE	3	247	247	247	247	247	.9	11.12	23.5	352
TO	2	262	255	259.5	262	262	12.91	30.73	45.65	1244
TO	3	69	62	65.36	69	69	21.5	101.6	151.39	17,039
BK	2	NA	4035	4112.8	4150	4060	68	269.5	329.2	133,380
BK	3	NA	932	968.21	978	NA	214.5	516.3	1263.83	NA

Table 4
Simulation results of LK-TFP and CPLEX runs across the generated problems.

Candidates	Teams	Optimum	LK-FTP solution			CPLEX solution	LK-TFP time (s)			CPLEX time (s)
			Min	Ave.	Max		Min	Ave.	Max	
6	2	24	24	24	24	24	*X*	*X*	*X*	1
10	2	108	108	108	108	108	0.001	0.07	0.11	43
16	2	317	317	317	317	317	0.001	0.26	0.89	628
16	4	111	111	111	111	111	0.02	4.1	13	28,348
16	5	78	78	78	78	78	0.05	1.76	4	31,591
20	2	900	900	900	900	900	1	16.3	50	37,405
20	4	NA	240	247	254	240	1.2	14.03	28	192,674
20	5	NA	158	163.2	169	149	0.9	8	18	163,423
30	2	NA	2805	2841.9	2916	1523	61	578.65	1305	218,912
30	4	NA	451	474.7	555	NA	12	44	84	NA
30	5	NA	441	460.7	495	NA	16	39.59	85	NA
30	10	NA	93	98.2	101	NA	9	12.8	26	NA
40	2	NA	4723	4804.1	4943	NA	135	1244.74	2307	NA
40	10	NA	1253	1281.6	1323	NA	41	127.26	237	NA
50	5	NA	1722	1849.7	1931	NA	71	218.5	419	NA
50	10	NA	426	439.2	461	NA	15	39.48	64	NA

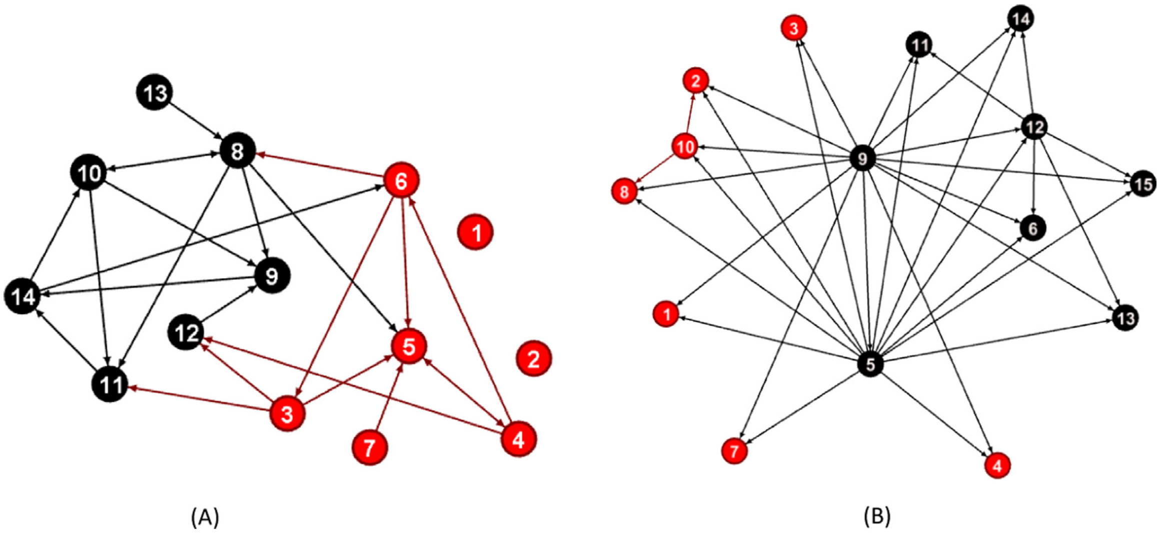


Fig. 11. Dividing (A) Dickson Bank Wiring and (B) Thurman Organizational Chart networks into two teams.

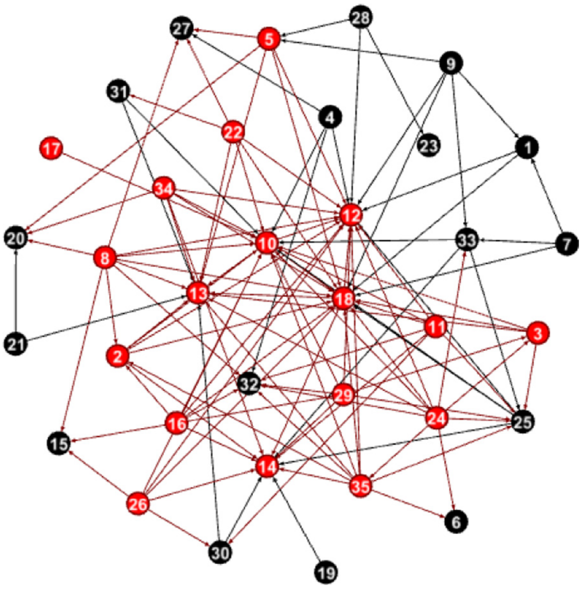


Fig. 12. Dividing Succor Teams network into two teams.

7.3. *Comparing LK-TFP with SGA*

Exact algorithms for NP-complete problems cannot be run in polynomial time. In order to perform benchmarking for LK-TFP, a metaheuristic – Standard Genetic Algorithm (SGA) – was implemented in JAVA and employed to solve the problem instances described in this section. In Genetic Algorithms (GAs), a number of operators such as crossover, mutation and selection are used for moving from one population of chromosomes (i.e., potential solutions) to a new population. Despite the fact that population-based algorithms (such as GAs) are potentially able to escape from local optima and utilize the benefits of exploration and exploitation strategies, they impose additional computation costs in search of (near-)optimal solutions [21]. In GAs, the operators and parameters have to be carefully tuned to improve the algorithm performance in each particular application.

In this study, a SGA with the following operators and parameter setting is implemented (for more details, see Chatterjee et al. [16]). Each chromosome in the SGA represents a feasible solution to TFP-SS. The selection operator is a combination of the tournament selection and elitism Chatterjee et al. [16], where the top five percent of the population are considered as elites in each iteration. We used a single cut-point crossover at a rate of 0.85 and a

Table 5
Simulation results of LK-TFP and CPLEX runs across BWD, TOC, ST and AD networks.

Network	Teams	Optimum	LK-FTP solution			CPLEX solution	LK-TFP time (s)			CPLEX time (s)
			Min	Ave.	Max		Min	Ave.	Max	
BWD	2	36	36	36	36	36	0.05	0.9	3.1	363
BWD	3	30	30	30	30	30	1.09	2.8	7.9	324
TOC	2	48	42	45.8	48	48	4.6	12.8	38	468
TOC	3	21	21	21	21	21	9	32.1	58.03	568
ST	2	761	749	754.9	761	761	239.8	580.15	789.4	42,757
ST	3	NA	345	350.1	368	NA	1208	1875.7	2349.8	NA
AD	5	NA	1820	1991.8	2191	NA	787	7846	21,940	NA
AD	10	NA	590	606.3	632	NA	896	11,918.5	32,101	NA
AD	20	NA	458	520.1	584	NA	453	9976.8	28,356	NA

standard mutation (which has been used for TSP) at a the rate of 0.4. The mutation and crossover operators need to keep the solution feasible. For instance, the mutation operator randomly selects two genes (i.e., two individuals in different teams) and switches them (similar to 2-opt in LK-TFP). The population size varies from 10 to 100 depending on the size of the corresponding social network. Further, we use simulation to randomly generate feasible solutions 100 times and compared the averages of objective function with those of LK-TFP

Table 6 summarizes the results of comparison between randomly generated solutions, the SGA and LK-TFP for synthetic undirected networks. The average and standard deviation of Random Solution is reported by taking the average over 100 randomly generated feasible solutions. Observe that objective function values obtained for random solutions are rather low, indicating that the use of optimization techniques for solving TFP-SS is well justified. It is also apparent that LK-TFP outperforms SGA in almost all the instances, while the SGA's run time is generally lower. Indeed, the SGA heuristic's run time depends on the population size and the stopping criterion (i.e., the maximum number of iterations); however, GAs tend to suffer from premature convergence.

The performance metrics of LK-TFP and SGA are compared with ZKC, FF, KM, TE, WEE, TO and BK networks in Table 7. In several cases, the SGA's performance is comparable with LK-TFP, however, LK-TFP performs consistently better overall. Simulation results obtained with LK-TFP, randomly generated solutions and the SGA runs across WD, TOC, ST and AD networks (directed networks) are summarized in Table 8.

8. Conclusion and discussion

This paper presents a mathematical framework which explicitly incorporates social structure in treating Team Formation Problem. The presented framework introduces models that quantitatively exploit the underlying network structure in team member communities. Importantly, this paper also opens broader research opportunities in the area of prescriptive SNA modeling. The presented framework sheds light on the relationship between social network theories and social structures, and discusses how to quantify social structure using information provided by the underlying graph. In order to assess team performance, network structure measures quantifying both social relations and individual attributes are given. The paper explores TFP-SS instances with measures based on network structures as edges, full dyads, triplets, k-stars, etc.

For a proven NP-Hard variation of TFP-SS, termed TFP-SSS, an integer programming formulation is presented for exact optimization. In order to tackle the problem instances of TFP-SSS, an efficient LK-TFP heuristic based on variable depth neighborhood search is developed for small-, medium- and large-sized instances with both real and randomly generated networks. The idea of λ-opt sequential search, introduced and developed by Lin, Kernighan and Helsgaun for solving large TSP instances, is also successfully applied to solve TFP-SS instances with undirected and directed networks. This paper describes the resulting LK-TFP heuristic as a tree search, and explains the roots of its efficiency, confirmed by computational results. The computational results demonstrate that LK-TFP returns high quality solutions and outperforms a well-tuned SGA metaheuristic.

Table 6
Simulation results obtained with LK-TFP, randomly generated solutions and SGA runs across the generated problem instances.

Candidates	Teams	Optimum	LK-FTP Solution			Random Solution		SGA	
			Min	Ave.	Max	Ave.	StD	Ave.	Time (s)
6	2	24	24	24	24	4.93	3.76	24	0
10	2	108	108	108	108	27.46	16.39	105.73	0.03
16	2	317	317	317	317	71.53	27.94	310.4	0.3
16	4	111	111	111	111	21.95	9.76	107.96	3.56
16	5	78	78	78	78	12.93	6.03	70.2	3.75
20	2	900	900	900	900	247.84	84.17	842.1	8.85
20	4	NA	240	247	254	60.14	18.38	220.3	7.3
20	5	NA	158	163.2	169	39.69	11.52	153.5	10.1
30	2	NA	2805	2841.9	2916	702.27	197.95	1740.5	38.85
30	4	NA	451	474.7	555	230.81	63.73	420.1	28.9
30	5	NA	441	460.7	495	107.92	34.09	360.3	22.25
30	10	NA	93	98.2	101	25.61	9.28	78.9	24.37
40	2	NA	4723	4804.1	4943	1414.58	301.74	3963.4	188.65
40	10	NA	1253	1281.6	1323	541.64	89.75	989.3	56.3
50	5	NA	1722	1849.7	1931	534.2	111.33	1123.8	63.4
50	10	NA	426	439.2	461	118.65	29.71	386	43.25

Table 7
Simulation results obtained with LK-TFP, randomly generated solutions and the SGA runs across ZKC, FF, KM, TE, WEE, TO and BK networks.

Network	Teams	Optimum	LK-FTP solution			Random solution		SGA	
			Min	Ave.	Max	Ave.	StD	Ave.	Time (s)
ZKC	2	NA	872	874.8	877	198.1	29.50	809.85	67.3
ZKC	4	NA	103	123.75	141	43.1	11.20	230.4	44.45
ZKC	5	NA	218	230.7	239	36.1	8.90	182.95	43.95
FF	2	137	137	137	137	25.3	9.42	131.8	15.25
FF	3	96	96	96	96	15.66	5.26	91.6	14.45
FF	4	59	51	56.5	59	10.66	5.26	50.85	14.45
KM	2	133	133	133	133	33.51	8.57	129.6	13.95
KM	3	72	65	71.6	72	13.96	10.39	55.9	14.75
TE	2	309	297	306.72	309	51.04	13.43	269.4	28.9
TE	3	259	247	253.4	259	41.8	9.77	215.8	45.1
WEE	2	548	548	548	548	80.1	29.40	504.9	24.8
WEE	3	247	247	247	247	46.14	15.64	240.9	29.5
TO	2	262	255	259.5	262	49.47	21.98	251.8	25.5
TO	3	69	62	65.36	69	20.37	7.77	64.1	24.85
BK	2	NA	4035	4112.8	4150	716.2	254.74	3885.75	79.2
BK	3	NA	932	968.21	978	304.6	104.80	836.15	71.35

Table 8
Simulation results obtained with LK-TFP, randomly generated solutions and the SGA runs across WD, TOC, ST and AD networks.

Network	Teams	Optimum	LK-FTP solution			Random solution		SGA	
			Min	Ave.	Max	Ave.	StD	Ave.	Time (s)
BWD	2	36	36	36	36	7.5	3.47	32.9	37.5
BWD	3	30	30	30	30	6.3	3.1	27.3	24.9
TOC	2	48	42	45.8	48	13.4	5.6	43.9	31.8
TOC	3	21	21	21	21	4.2	4.8	18.7	23.9
ST	2	761	749	754.9	761	119.4	20.36	749.1	86.1
ST	3	NA	345	350.1	368	75.2	17.8	336.4	52.6
AD	5	NA	1820	1991.8	2191	347.5	43.9	1618.2	1289.7
AD	10	NA	590	606.3	632	109.2	20.8	591.3	1326.5
AD	20	NA	458	520.1	584	54.3	18.8	359.4	1183.4

Touching upon the TSP ideas that are more general than the Lin–Kernighan inspired works, the stem-and-cycle (S&C) method is a family of heuristics that has proven to excel in solving very large TSP problems [11]. Bergey et al. explored how these heuristics can be represented within the common ejection chain (EC) framework that efficiently creates variable-depth neighborhoods for local search procedures; a λ-move in LK-TFP is just one type of an ejection and trial move in EC. Bergey et al. employed the state-of-the-art data structures to implement new TSP algorithms, increasing their efficiency [24]. Indeed, EC decomposes a very large neighborhood into a sequence of component neighborhood structures that can be evaluated in polynomial time [45]. The S&C method as well as GRASP [41], an efficient algorithm for community detection based on the modularity evaluation, can spark new developments in the TFP-SS heuristic design.

Observe also that TFP-SS instances can be interpreted as certain clique relaxation problems, by identifying the NSMs that can be used in TFP-SS to match clique relaxation-based formulations. As such, consider a TFP-SS instance, where only the number of edges is considered as the NSM and the objective is to maximize the minimum over all the teams' outcomes. This optimization problem can be interpreted as that of finding M network subsets such that each of them is a relaxed clique of the pre-defined minimal quality (e.g., an s-defective clique for a given fixed value of s) [9,43]. This observation signals that the LK-TFP framework can be useful for current and future efforts in the clique relaxation domain, where clustering algorithms have been dominant so far.

While this paper demonstrates the advantages of the presented framework to prescriptively implement SNA theories in TFP, some potential directions for further improvement exist. While the framework is able to generate a range of models for TFP based on social structures, the question of selecting the best model for a given application deserves more attention. Also, this paper avoided an extended discussion on estimating the function relating NSMs and observed team outcomes: this issue can be addressed in the future. Furthermore, LK-TFP can be further tested and implemented to address other, similar problems, e.g., clique relaxation problems. Since TFP-SS presents computationally challenging problems, other optimization algorithms can be designed for treating TFP-SS models; exact methods are of particular interest. Finally, the presented framework's ideas can be extended to problems beyond the team formation problem. Network clustering, information influence, community detection, and scheduling (e.g., of work shifts) problems are especially promising.

Acknowledgments

While contributing to this work, the second author was supported in part by the Academy of Finland, Grant Number 268078 (MineSocMed), and in part by the 2015 U.S. Air Force Summer Faculty Fellowship Program, sponsored by the Air Force Office of Scientific Research. This support is gratefully acknowledged. The authors acknowledge the anonymous referees whose remarks improved the quality of this paper significantly.

Appendix A

Theorem 1. *Consider an instance of TFP-SSS, with M teams to be formed out of N individuals:*

Instance: A graph $\mathcal{G}(V, E)$, $|\mathcal{G}| = N$; $n_j \in Z^+$ *for* $1 \le j \le M$; *a partition of disjoint sets* $X_1, X_2, \dots, X_M$, *where* $X_j \subseteq \mathcal{G}(V, E)$ *for* $1 \le j \le M$ *and* $\theta_l \in R$ *for* $l \in \mathcal{L}$.

Question: Is there a partition of V into M disjoint subsets $X_1 \cup X_2 \cup \dots \cup X_M$, *with* $|X_j| = n_j$, *such that* $\sum_{j=1}^{M} \mathcal{P}(X_j)$ *is maximized, where* $P(X_j) = \sum_{i=1}^{n_j} \sum_{l \in \mathcal{L}} \theta_l \mathcal{F}_l(N_{X_j}(v_i))$ *for* $1 \le j \le M$ *and* $\sum_{j=1}^{M} n_j = N$?

The presented problem is NP-Hard.

Proof. The proof proceeds by a polynomial-time reduction from Partition into Triangles. An arbitrary instance of Partition into Triangles is given.

Instance: A graph $\mathcal{G}(V, E)$, with $|\mathcal{G}| = 3q$, for a given fixed integer q.

Question: Can the vertices of $\mathcal{G}$ be partitioned into q disjoint sets $V_1, V_2, \dots, V_q$, each containing exactly 3 vertices, such that for each $V_i = \{u_i, v_i, w_i\}$, $1 \le i \le q$, three edges $\{u_i, v_i\}$, $\{u_i, w_i\}$, and $\{v_i, w_i\}$ all belong to E [25]?

Consider a particular instance of TFP-SSS with $N = 3q$ and $X_j = V_j$ for $1 \le j \le q$ and $M = q$. Set $\theta_l = 1$ for $l = \{triangle\}$ and $\theta_l = 0$ otherwise (i.e., use the number of triangles as the only NSM in the objective function of TFP-SSS. Finally, set $n_j = 3$ for $1 \le j \le q$. To demonstrate that there is a one-to-one correspondence between the described Partition into Triangles and TFP-SSS instances, suppose that X_j^*, $j = 1, \dots, q$ is the optimal solution to TFP-SSS. Therefore, one has that $|X_j^*| = 3$, and also, $\sum_{j=1}^{M} \mathcal{F}_{triangle}(N_{X_j}(v_i)) = q$ is the maximal value of the objective function. Observe that V_j^*, with $|V_j^*| = 3$, which assigns three nodes to partition j is equivalent to X_j^*. Note that these three nodes form a triangle. Suppose that $X_j^* \neq V_i^*$ for some i's and j's, with $1 \le i, j \le q$. Then, there exists at least one partition with 3 nodes $\{\overline{u_i}, \overline{v_i}, \overline{w_i}\} \in X_j^*$ such that $\{e_{\overline{u_i}}, e_{\overline{v_i}}, e_{\overline{w_i}}\} \notin E$. Therefore, one has $\sum_{j=1}^{M} \mathcal{F}_{triangle}(N_{X_j^*}(v_i)) < q$, which is a contradiction since there are q partitions and at least one of them does not have a triangle. Hence, $X_j^* = V_i^*$ is an optimal solution for both problems. This completes the proof. □

References

[1] Abbasi A, Altmann J. On the correlation between research performance and social network analysis measures applied to research collaboration networks. In: 44th Hawaii international conference on systems science (HICSS-44), January 4–7, Hawaii, USA; 2011.

[2] Agrawal R, Golshan B, Terzi E. Forming beneficial teams of students in massive online classes. In: Proceedings of the first ACM conference on Learning@ scale conference. ACM; Atlanta, Georgia, USA: 2014. p. 155–6.

[3] Agustín-Blas LE, Salcedo-Sanz S, Ortiz-García EG, Portilla-Figueras A, Pérez-Bellido ÁM, Jiménez-Fernández S. Team formation based on group technology: a hybrid grouping genetic algorithm approach. Comput Oper Res 2011;38 (2):484–95.

[4] Albert R, Barabási A-L. Statistical mechanics of complex networks. Rev Mod Phys 2002;74(1):47.

[5] Aral S, Muchnik L, Sundararajan A. Distinguishing influence-based contagion from homophily-driven diffusion in dynamic networks. Proc Natl Acad Sci 2009;106(51):21544–9.

[6] Arulselvan A, Commander CW, Elefteriadou L, Pardalos PM. Detecting critical nodes in sparse graphs. Comput Oper Res 2009;36(7):2193–200.

[7] Awal GK, Bharadwaj K. Team formation in social networks based on collective intelligence-an evolutionary approach. Appl Intell 2014;41(2):627–48.

[8] Baker K, Powell S, et al. Methods for assigning students to groups: a study of alternative objective functions. J Oper Res Soc 2002;53(4):397–404.

[9] Balasundaram B, Butenko S, Hicks IV. Clique relaxations in social network analysis: the maximum k-plex problem. Oper Res 2011;59(1):133–42.

[10] Balkundi P, Barsness Z, Michael JH. Unlocking the influence of leadership network structures on team conflict and viability. Small Group Res 2009;40 (3):301–22.

[11] Bergey P, King M. Team machine: a decision support system for team formation. Decis Sci J Innov Educ 2014;12(2):109–30.

[12] Bettinelli A, Liberti L, Raimondi F, Savourey D. The anonymous subgraph problem. Comput Oper Res 2013;40(4):973–9.

[13] Borgatti SP. Centrality and network flow. Soc Netw 2005;27(1):55–71.

[14] Borgatti SP, Halgin DS. On network theory. Organ Sci 2011;22(5):1168–81.

[15] Ceravolo DJ, Schwartz DG, Foltz-Ramos KM, Castner J. Strengthening communication to overcome lateral violence. J Nurs Manag 2012;20(5):599–606.

[16] Chatterjee S, Carrera C, Lynch LA. Genetic algorithms and traveling salesman problems. Eur J Oper Res 1996;93(3):490–510.

[17] Chen S-J, Lin L. Modeling team member characteristics for the formation of a multifunctional team in concurrent engineering. IEEE Trans Eng Manag 2004;51(2):111–24.

[18] Contractor NS, Wasserman S, Faust K. Testing multitheoretical, multilevel hypotheses about organizational networks: an analytic framework and empirical example. Acad Manag Rev 2006;31(3):681–703.

[19] Cutshall R, Gavirneni S, Schultz K. Indiana University's Kelley School of Business uses integer programming to form equitable, cohesive student teams. Interfaces 2007;37(3):265–76.

[20] Dorn C, Dustdar S. Composing near-optimal expert teams: a trade-off between skills and connectivity. In: On the move to meaningful internet systems: OTM 2010. Springer; Hersonissos, Crete, Greece: 2010. p. 472–89.

[21] Farasat A, Menhaj MB, Mansouri T, Moghadam MRS. Aro: a new model-free optimization algorithm inspired from asexual reproduction. Appl Soft Comput 2010;10(4):1284–92.

[22] Fitzpatrick E, Askin R, Goldberg J. Using student conative behaviors and technical skills to form effective project teams. In: 31st annual frontiers in education conference, 2001, vol. 3. IEEE; Reno, NV, USA: 2001. p. S2G–8.

[23] Fitzpatrick EL, Askin RG. Forming effective worker teams with multi-functional skill requirements. Comput Ind Eng 2005;48(3):593–608.

[24] Gamboa D, Rego C, Glover F. Data structures and ejection chains for solving large-scale traveling salesman problems. Eur J Oper Res 2005;160(1):154–71.

[25] Garey MR, Johnson DS. Computers and intractability, vol. 174. New York: Freeman; 1979.

[26] Gaston M, Simmons J, DesJardins M. Adapting network structures for efficient team formation. In: Proceedings of the AAAI 2004 fall symposium on artificial multi-agent learning; 2004.

[27] Girvan M, Newman ME. Community structure in social and biological networks. Proc Natl Acad Sci 2002;99(12):7821–6.

[28] Goyal A, Bonchi F, Lakshmanan LV. Learning influence probabilities in social networks. In: Proceedings of the third ACM international conference on Web search and data mining. ACM; New York City, New York, USA: 2010. p. 241–50.

[29] Hahn J, Moon JY, Zhang C. Emergence of new project teams from open source software developer networks: impact of prior collaboration ties. Inf Syst Res 2008;19(3):369–91.

[30] Helsgaun K. An effective implementation of the Lin–Kernighan traveling salesman heuristic. Eur J Oper Res 2000;126(1):106–30.

[31] Juang M-C, Huang C-C, Huang J-L. Efficient algorithms for team formation with a leader in social networks. J Supercomput 2013:1–17.

[32] Kargar M, An A, Zihayat M. Efficient bi-objective team formation in social networks. In: Machine learning and knowledge discovery in databases. Springer; Bristol, UK: 2012. p. 483–98.

[33] Kempe D, Kleinberg J, Tardos É. Maximizing the spread of influence through a social network. In: Proceedings of the ninth ACM SIGKDD international conference on knowledge discovery and data mining. ACM; Washington, DC, USA: 2003. p. 137–46.

[34] Kim BW, Kim JM, Lee WG, Shon JG. Parallel balanced team formation clustering based on mapreduce. In: Advances in computer science and ubiquitous computing. Springer; 2015. p. 671–5.

[35] Kothari R, Ghosh D. Insertion based Lin–Kernighan heuristic for single row facility layout. Comput Oper Res 2013;40(1):129–36.

[36] 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. ACM; Paris, France: 2009. p. 467–76.

[37] Leite AR, Borges AP, Carpes LM, Enembreck F. Improving the distributed constraint optimization using social network analysis. In: Advances in artificial intelligence – SBIA 2010. Springer; São Bernardo do Campo, Brazil: 2011. p. 243–52.

[38] Lin S, Kernighan BW. An effective heuristic algorithm for the traveling-salesman problem. Oper Res 1973;21(2):498–516.

[39] Mahar S, Winston W, Wright PD. Eli Lilly and company uses integer programming to form volunteer teams in impoverished countries. Interfaces 2013;43(3):268–84.

[40] Manser T. Teamwork and patient safety in dynamic domains of healthcare: a review of the literature. Acta Anaesthesiol. Scand. 2009;53(2):143–51.

[41] Nascimento MC, Pitsoulis L. Community detection by modularity maximization using grasp with path relinking. Comput Oper Res 2013;40(12):3121–31.

[42] Newman ME. Modularity and community structure in networks. Proc Natl Acad Sci 2006;103(23):8577–82.

[43] Pattillo J, Youssef N, Butenko S. On clique relaxation models in network analysis. Eur J Oper Res 2012.

[44] Pirim H, Ekşioğlu B, Perkins AD, Yüceer Ç. Clustering of high throughput gene expression data. Comput Oper Res 2012;39(12):3046–61.

[45] Rego C, Gamboa D, Glover F, Osterman C. Traveling salesman problem heuristics: leading methods, implementations and latest advances. Eur J Oper Res 2011;211(3):427–41.

[46] Robins G, Pattison P, Kalish Y, Lusher D. An introduction to exponential random graph (¡i¿ p¡/i¿¡ sup ¿*¡/sup ¿) models for social networks. Soc Netw 2007;29(2):173–91.

[47] Ronald B. Structural holes: the social structure of competition.Harvard: Cambridge; 1992.
[48] Ruef M, Aldrich HE, Carter NM. The structure of founding teams: homophily, strong ties, and isolation among us entrepreneurs. Am Soc Rev 2003:195–222.
[49] Salari M, Naji-Azimi Z. An integer programming-based local search for the covering salesman problem. Comput Oper Res 2012;39(11):2594–602.
[50] San Segundo P, Rodríguez-Losada D, Jiménez A. An exact bit-parallel algorithm for the maximum clique problem. Comput Oper Res 2011;38(2):571–81.
[51] San Segundo P, Tapia C. Relaxed approximate coloring in exact maximum clique search. Comput Oper Res 2014;44:185–92.
[52] Shi Z, Hao F. A strategy of multi-criteria decision-making task ranking in social-networks. J Supercomput 2013:1–16.
[53] Snijders TA, Van de Bunt GG, Steglich CE. Introduction to stochastic actor-based models for network dynamics. Soc Netw 2010;32(1):44–60.
[54] Squillante M. Decision making in social networks. Int J Intell Syst 2010;25 (3):255.
[55] Wasserman S, Faust K. Social network analysis: methods and applications, vol. 8. Cambridge University Press; 1994.
[56] Wi H, Oh S, Mun J, Jung M. A team formation model based on knowledge and collaboration. Expert Syst Appl 2009;36(5):9121–34.
[57] Zhang L, Zhang X. Multi-objective team formation optimization for new product development. Comput Ind Eng 2013;64(3):804–11.
[58] Zhang Z-x, Luk W, Arthur D, Wong T. Nursing competencies: personal characteristics contributing to effective nursing performance. J Adv Nurs 2001;33 (4):467–74.
[59] Zhong X, Huang Q, Davison RM, Yang X, Chen H. Empowering teams through social network ties. Int J Inf Manag 2012;32(3):209–20.
[60] Zhu M, Huang Y, Contractor NS. Motivations for self-assembling into project teams. Soc Netw 2013;35(2):251–64.