Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence (AAAI-17)

Profit-Driven Team Grouping in Social Networks

Shaojie Tang
Naveen Jindal School of Management
University of Texas at Dallas

Abstract

In this paper, we investigate the profit-driven team grouping problem in social networks. We consider a setting in which people possess different skills and compatibility among these individuals is captured by a social network. Here, we assume a collection of tasks, where each task requires a specific set of skills, and yields a different profit upon completion. Active and qualified individuals may collaborate with each other in the form of *teams* to accomplish a set of tasks. Our goal is to find a grouping method that maximizes the total profit of the tasks that these teams can complete. Any feasible grouping must satisfy the following three conditions: (i) each team possesses all skills required by the task, (ii) individuals within the same team are social compatible, and (iii) each individual is not overloaded. We refer to this as the TEAMGROUPING problem. Our work presents a detailed analysis of the computational complexity of the problem, and propose a LP-based approximation algorithm to tackle it and its variants. Although we focus on team grouping in this paper, our results apply to a broad range of optimization problems that can be formulated as a cover decomposition problem.

Introduction

In this paper, we consider the problem of grouping teams on a networked community of people with diverse skill sets. We consider a setting in which people possess different skills and compatibility among these individuals is captured by a social network. Here, we assume a collection of tasks, where each task requires a specific set of skills, and yields a different profit upon completion. Active and qualified individuals may collaborate with each other in the form of *teams* to accomplish a set of tasks. Our goal is to find a grouping method that maximizes the total profit of the tasks that these teams can complete. One relevant example is from the domain of online labor markets, such as Freelancer, Upwork , and Guru. In these online platforms, freelancers with various skills can be hired to work on different types of projects. Instead of just working independently, more and more freelancers are realizing that it is more beneficial to work as a team, together with other solo freelancers who have complementary skills (Golshan, Lappas, and Terzi 2014). This allows them to expand their talent pool and achieve better load balance. Nowadays many major platforms in this area such as Upwork has provided team-hiring services to their enterprise customers.

In this paper, we formalize the profit-driven team grouping problem as follows: we assume a set of m individuals $\mathcal{V}$ and a set of n skills $\mathcal{S}$. Each individual $u \in \mathcal{V}$, is represented by a subset of skills, i.e., $u \subseteq \mathcal{S}$; these are the skills that the individual possesses. We also assume a set of tasks $\mathcal{T}$, every task $t \in \mathcal{T}$ can also be represented by the set of skills that are required in order for the task to be completed (i.e., $t \subseteq \mathcal{S}$). Finally, every task t is associated with a profit λ_t, this could be the benefit that the completion of a task will yield for the platform. The team grouping problem is to group individuals to different teams and assign them to different tasks satisfying the following three conditions: (i) each team possesses all skills required by the task, (ii) individuals within the same team have high social compatibility, and (iii) each individual is not overloaded. Our goal is to maximize the sum of profits from all tasks that can be performed. We refer to this as the TEAMGROUPING problem. It was worth noting that the social compatibility among individuals can be interpreted in many ways. In this work, we model the *social compatibility* by means of a social network. One natural and popular option with respect to capturing the underlying social compatibility of a team is *connectivity*. This follows the approach of (Lappas, Liu, and Terzi 2009) and requires that each team forms a connected graph. Other options to measure social compatibility include the diameter of a team (Anagnostopoulos et al. 2012), i.e., the induced graph of any team in G must have small diameter. Fortunately, our results are not restricted to any specific measures of social compatibility, instead, we propose a general framework that works for any reasonable measure.

Contributions: To the best of our knowledge we are the first to define and study the TEAMGROUPING problem and its variants. We summarize our contributions as follows:

(1) We show that this problem is $1/\ln m$ hard to approximate, i.e., it is NP-Hard to find a solution with approximation ratio larger than $1/\ln m$.

(2) We propose a LP-based algorithm with approximation ratio $\max\{\mu/\Delta, \mu/2\sqrt{m}\}$ where Δ denotes the size of the largest minimal team and $1/\mu$ is the approximation ratio of MINCOSTTEAMSELECTION problem (defined formally in Definition 1). If there is no constraint on social compatibil-

ity, this ratio is equivalent to $\max\{1/n \ln n, 1/2\sqrt{m} \ln n\}$.

(3) We also consider two extensions of the basic model. In the first extension, we relax the assumption that each individual can only participate in one task by allowing individuals to have different load limits. In the second part, we consider the scenario when each task can only be performed by a fixed number of teams or times. We develop effective approximation algorithms to tackle both extensions.

(4) Although we restrict our attention to the profit-driven team grouping problem in this paper, our results apply to other applications such as lifetime maximization problem in wireless networks (Bagaria, Pananjady, and Vaze 2013), resource allocation and scheduling problems (Pananjady, Bagaria, and Vaze 2014), and supply chain management problems (Lu 2011). In this sense, this research contributes fundamentally to the development of approximate solutions for any problems that fall into the family of generalized cover decomposition problem.

Related Work

To the best of our knowledge we are the first to formulate and study the team grouping problem and its variants. However, our work is closely related to other team formation and cluster hiring problems. (Lappas, Liu, and Terzi 2009) introduce the minimum cost team formation problem. Given a set of skills that need to be covered and social network, their objective is to select a team of experts that can cover all required skills, while ensuring efficient communication between the team's members. There is a considerable amount of literature on this topic and its variants (Li and Shan 2012; Kargar, Zihayat, and An 2013; Dorn and Dustdar 2010; Gajewar and Sarma 2012; Kargar and An 2011; Li and Shan 2010; Sozio and Gionis 2010). In (Golshan, Lappas, and Terzi 2014), they study cluster hire problem, where the objective is to hire a profit-maximizing team of experts with the ability to complete multiple projects, subject to a fixed budget. Different from all the above works where they aim to select a best qualified team, our objective is to group individuals into multiple teams. It turns out that these two problems are closely related, this allows us to leverage existing techniques on team formation to solve our problem.

The other category of related work is maximum disjoint set cover problem (Bagaria, Pananjady, and Vaze 2013). Given a universe, and a set of subsets, the objective is to find as many set covers as possible such that all set covers are pairwise disjoint. Our problem can be considered as a generalized disjoint set cover problem in the sense that every task in our problem may have different requirement of coverage, capacity constraint, and profit, and any feasible set cover must satisfy both coverage requirement as well as social compatibility. In addition, the requirement of "disjoint" is also relaxed by allowing individuals to have different load limits in our problem. Therefore, this work contributes fundamentally to the generalized cover decomposition problem.

Problem Formulation

Individuals. Skills. Tasks. In this paper we will assume that there is a set of n skills $\mathcal{S}$, a set of m individuals $\mathcal{V}$ and a set of k tasks $\mathcal{T}$. Each individual $u \in \mathcal{V}$, is represented by a subset of skills, i.e., $u \subseteq \mathcal{S}$; these are the skills that the individual possesses. Similarly, every task $t \in \mathcal{T}$ can also be represented by the set of skills that are required in order for the task to be completed (i.e., $t \subseteq \mathcal{S}$). In addition, each task t is associated with a profit λ_t upon the completion of this task. We assume that each task has unlimited number of copies, i.e., the same task can be performed by multiple teams. Notice that this assumption may not always hold in real world, to this end, we also study the case where each task has a capacity constraint, i.e., task t can only be performed up to g_t times.

Load. Our basic model assumes that each individual can only participate in *one* task. In our extended model, we will relax this assumption by allowing individuals to have different load limits, i.e.,, each individual u can participate in up to f_u number of tasks.

Teams. In practice, the social compatibility among individuals play an important role in a team work. For example, low social compatibility or high coordination cost will degrade the efficiency of organizations (Coase 1937). We model the *social compatibility* by means of a social network $G = (\mathcal{V}, \mathcal{E})$. One natural and popular option with respect to capturing the underlying social compatibility of a team is *connectivity*. This follows the approach of (Lappas, Liu, and Terzi 2009) and requires that each team C forms a connected graph. It was worth noting that there exist many ways to quantify the social compatibility among individuals, other options include the *diameter constraint* (Anagnostopoulos et al. 2012), i.e., the longest shortest path among team members in G is no larger than a given threshold. Fortunately, our results are not restricted to any specific measures of social compatibility, instead, we propose a general framework that works for any measure of social compatibility that has been explicitly defined.

Problem Formulation. For a team of individuals $C \subseteq \mathcal{V}$, we say that team C has a skill s if there exist at least one individual $u \in C$, such that u has skill s, i.e., $s \in u$. For a task $t \in \mathcal{T}$, we say that team C covers t if C (as a team) has all the skills required for t. Clearly, a team of individuals may cover more than one tasks, but they can only participate in one of those tasks due to each individual's load limit[1]. We define the set of qualified teams for task t to be the set of distinct teams that is social compatible and covers task t. That is,

$$\mathcal{C}_t = \{C \subseteq \mathcal{V} | C \text{ is social compatible and covers } t\}$$

Let C_{ti} denote the i-th team in $\mathcal{C}_t$. A minimal qualified team of a task t is a qualified team of this task that is not a superset of any other qualified team. In the rest of this paper, we only consider minimal qualified teams and let $\mathbf{C} = \{\mathcal{C}_1, \cdots, \mathcal{C}_k\}$ denote the set of sets of minimal qualified teams for all tasks.

[1] As mentioned earlier,this assumption will be relaxed in the extended model.

The objective of this work is to find a most profitable way to group individuals into different teams, and assign one task to each team, such that (i) each team possesses all skills required by the corresponding task, (ii) all team members are social compatible, and (iii) each individual can only participate in one team. Given the above notation and constraint, we can now define TEAMGROUPING problem as follows:

P.1: *Maximize* $\sum_{C_{ti} \in \mathcal{C}_t \in \mathbf{C}} (x_{ti} \cdot \lambda_t)$
subject to:

$$\begin{cases} \sum_{u \in C_{ti} \in \mathcal{C}_t \in \mathbf{C}} x_{ti} \leq 1, \forall u \in \mathcal{V} \\ x_{ti} \in \{0, 1\}, \forall C_{ti} \in \mathcal{C}_t \in \mathbf{C} \end{cases}$$

In the above formulation, x_{ti} indicates whether team C_{ti} has been selected ($x_{ti} = 1$) or not ($x_{ti} = 0$), and the first constraint specifies the load limit on each individual. The following results show that we cannot hope to achieve an $\omega(1/\ln m)$ approximation ratio for this problem.

Theorem 1 *The* TEAMGROUPING *problem is* $1/\ln m$ *hard to approximate.*

Proof: For our proof, we will consider a simplified version of TEAMGROUPING problem with only one task, i.e., $k = 1$, and there is no constraint on social compatibility. We call this problem S-TEAMGROUPING. We next prove that the *maximum disjoint set cover cover problem* (DSCP) can be reduced to S-TEAMGROUPING. The formal definition of DSCP is as follows: Given a universe $\mathcal{U}$, and a set of subsets $\mathcal{X}$, find as many set covers as possible such that all set covers are pairwise disjoint. We wish to formulate an equivalent S-TEAMGROUPING with a set of skills $\mathcal{S}$ required the task, and a set of individuals $\mathcal{V}$. Let $\mathcal{S} = \mathcal{U}$ and $\mathcal{V} = \mathcal{X}$. Because there is only one task and no constraint on social compatibility, S-TEAMGROUPING is equivalent to grouping $\mathcal{V}$ into maximum number of disjoint teams each of which can cover all skills in $\mathcal{S}$. It was shown in (Bagaria, Pananjady, and Vaze 2013) that the DSCP is hard to achieve an $\omega(1/\ln m)$ approximation ratio unless $NP \subseteq DTIME(n^{O(\ln \ln m)})$, thus TEAMGROUPING, which is a general case of S-TEAMGROUPING, is also $1/\ln m$ hard to approximate. □

One immediate result from the above proof is that if there is only one task and no constraint on social compatibility, we can simply adopt the method proposed in (Bagaria, Pananjady, and Vaze 2013) to achieve $1/\ln m$ approximation ratio. In the following, we propose a LP-based approximation algorithm to tackle the general case.

LP-Based Approximation Algorithm

In this section, we give a $\max\{\mu/\Delta, \mu/2\sqrt{m}\}$-approximation algorithm for TEAMGROUPING, where $1/\mu$ is the approximation factor of the algorithm for the MINCOSTTEAMSELECTION problem, and $\Delta = \max_{C \in \mathcal{C}_t \in \mathbf{C}} |C|$, i.e., the size of the largest minimal team. The formal definition of MINCOSTTEAMSELECTION will be introduced in Definition 1.

LP Relaxation

Primal LP of P.1: *Maximize* $\sum_{C_{ti} \in \mathcal{C}_t \in \mathbf{C}} (x_{ti} \cdot \lambda_t)$
subject to:

$$\begin{cases} \sum_{u \in C_{ti} \in \mathcal{C}_t \in \mathbf{C}} x_{ti} \leq 1, \forall u \in \mathcal{V} \\ 0 \leq x_{ti} \leq 1, \forall C_{ti} \in \mathcal{C}_t \in \mathbf{C} \end{cases}$$

The above is the linear program (LP) relaxation of **P.1**. This LP has m constraints (excluding the trivial constraints $x_{ti} \geq 0, \forall C_{ti} \in \mathcal{C}_t \in \mathbf{C}$). However, since the number of variables $\sum_{t \in \mathcal{T}} |\mathcal{C}_t|$ could easily be exponential in the number of individuals, standard LP solvers can not solve this packing LP effectively.

To tackle this challenge, we adopt ellipsoid algorithm (Grötschel, Lovász, and Schrijver 1981) which is capable of solving certain LP problems where the number of constraints is exponential in polynomial time.

We refer to the above relaxed TEAMGROUPING problem as the primal LP. The dual to this primal LP associates a price $y(u)$ for each node $u \in \mathcal{V}$:

Dual LP of P.1: *Minimize* $\sum_{u \in \mathcal{V}} y(u)$
subject to:

$$\begin{cases} \sum_{u \in C_{ti}} y(u) \geq \lambda_t, \forall C_{ti} \in \mathcal{C}_t \in \mathbf{C} \\ y(u) \geq 0, \forall u \in \mathcal{V} \end{cases}$$

We leverage the ellipsoid method for exponential-sized LP with an (approximate) separation oracle to establish an approximation-preserving reduction from MINCOSTTEAMSELECTION, as defined in the following, to primal LP.

Definition 1 (MincostTeamSelection) *Assume that there is a set of skills $\mathcal{S}$ and individuals $\mathcal{V}$, each individual $u \in \mathcal{V}$ is associated with a cost and possesses a subset of skills. Find a team of individuals with minimum cost such that (1) all team members are social compatible, and (2) all skills in $\mathcal{S}$ can be covered.*

Depending on the definition of social compatibility, MINCOSTTEAMSELECTION has been intensively studied in the literature. In (Lappas, Liu, and Terzi 2009), they propose to use connectivity as a measure of social compatibility, that is, all team members must be connected in the social network. Under this context, the MINCOSTTEAMSELECTION problem can be reduced from *node weight group steiner tree* problem (Khandekar, Kortsarz, and Nutov 2012) which admits a performance ratio of $O(|\mathcal{E}|^{1/2} \ln |\mathcal{E}|)$ where $|\mathcal{E}|$ is the number of edges in the social network. It was worth noting that condition (1) can be replaced by other reasonable measurements on social compatibility among team members, for instance, some work (Anagnostopoulos et al. 2012) requires that a team must have bounded diameter. The following theorem is not restricted to any specific measure of social compatibility. Due to space constraints, the missing proofs are deferred to the full version.

Theorem 2 *If there is a polynomial $1/\mu$-approximation algorithm for* MINCOSTTEAMSELECTION*, then there exists a polynomial μ-approximation algorithm for* ***P.1****.*

Approximation Algorithm

Having described the LP relaxation, we now propose an approximation algorithms computing a group of teams from LP solutions. Our approach involves two algorithms as subroutines.

Candidate Solution I: Let $\mathcal{C}_t^H$ denote the subset of teams on task t corresponding to the separating hyper-planes found by the above separation oracle while running the ellipsoid algorithm, define $\mathbf{C}^H = \{\mathcal{C}_1^H, \cdots, \mathcal{C}_k^H\}$. In the first algorithm (Algorithm 1), we directly apply the deterministic rounding (Algorithm 4) to $\mathbf{C}^H$. We can prove that this algorithm achieves μ/Δ approximation ratio. For ease of presentation, we put the detailed description of our rounding technique in next section.

Algorithm 1 Candidate Grouping - I

1: Apply deterministic rounding (Algorithm 4) to $\mathbf{C}^H$ and output a group of teams.

Lemma 1 *Algorithm 1 achieves μ/Δ approximation ratio for* TEAMGROUPING.

Candidate Solution II: The framework of the second candidate solution (Algorithm 2) can be summarized as follows:

Step 1: For every task t, we first partition $\mathcal{C}_t^H$ to two disjoint subsets $\mathcal{C}_t^{H_1}$ and $\mathcal{C}_t^{H_2}$ such that: $\forall C \in \mathcal{C}_t^{H_1} : |C| \leq \sqrt{m}$ and $\forall C \in \mathcal{C}_t^{H_2} : |C| > \sqrt{m}$. That is, $\mathcal{C}_t^{H_1}$ (resp. $\mathcal{C}_t^{H_2}$) contains all teams with no more (resp. more) than $\sqrt{m}$ individuals. Let $\mathbf{C}^{H_1} = \{\mathcal{C}_1^{H_1}, \cdots, \mathcal{C}_k^{H_1}\}$ and $\mathbf{C}^{H_2} = \{\mathcal{C}_1^{H_2}, \cdots, \mathcal{C}_k^{H_2}\}$.

Step 2: Apply deterministic rounding (Algorithm 4) to $\mathbf{C}^{H_1}$ and output a group of teams $\widetilde{\mathcal{C}}$.

Step 3: Select a team, say $C_{t_{\max}}$, from $\mathbf{C}^{H_2}$ whose task $t_{\max}$ has the highest profit $\lambda_{t_{\max}}$.

Step 4: Compare $\widetilde{\mathcal{C}}$ and $\{C_{t_{\max}}\}$, choose the one with larger profit as the final output, i.e, the profit of the returned solution is $\max\{\sum_{C_{ti} \in \widetilde{\mathcal{C}}} \lambda_t, \lambda_{t_{\max}}\}$.

Algorithm 2 Candidate Grouping - II

1: Partition $\mathbf{C}^H$ into two subsets $\mathbf{C}^{H_1}$ and $\mathbf{C}^{H_2}$
2: Apply the deterministic rounding (Algorithm 4) to $\mathbf{C}^{H_1}$ and output $\widetilde{\mathcal{C}}$.
3: Select a team with the highest profit, say $C_{t_{\max}}$, from $\mathbf{C}^{H_2}$.
4: Compare $\widetilde{\mathcal{C}}$ and $\{C_{t_{\max}}\}$, return the one with larger profit.

We next prove that the approximation ratio of Algorithm 2 can be bounded by $\mu/2\sqrt{m}$.

Lemma 2 *Algorithm 2 achieves $\mu/2\sqrt{m}$ approximation ratio for* TEAMGROUPING.

Putting It All Together. Given solutions returned from Algorithm 1 and Algorithm 2, we simply choose the one with higher profit as our final output. We refer to this algorithm as Approx-TG (Algorithm 3). Lemma 1 and Lemma 2 together imply our main theorem.

Algorithm 3 Approx-TG

1: Compute two candidate solutions using Algorithm 1 and Algorithm 2.
2: Return the one with higher profit.

Theorem 3 *Approx-TG achieves $\max\{\mu/\Delta, \mu/2\sqrt{m}\}$ approximation ratio for* TEAMGROUPING.

Now consider a special case of TEAMGROUPING where there is no constraint on social compatibility. Under this setting, MINCOSTTEAMSELECTION problem as defined in Definition 1 is equivalent to classic *weighted set cover problem* (Chvatal 1979), which allows $\ln n$ approximation. In addition, we have $\Delta \leq n$, this is because the number of possible skills is at most n, if there is no constraint on social compatibility, any minimal qualified team contains at most n individuals. Then the following corollary holds by replacing μ using $1/\ln n$, and Δ using n in Theorem 3.

Corollary 4 *If there is no constraint on social compatibility, Approx-TG achieves $\max\{1/n \ln n, 1/2\sqrt{m} \ln n\}$ approximation ratio for* TEAMGROUPING.

It was worth noting that if $n \ll m$, i.e, the number of skills is much smaller than the number of individuals, the above approximation ratio can be further rewritten as $1/n \ln n$.

Consider another special case that uses connectivity to measure the social compatibility. As discussed earlier, under this setting, the MINCOSTTEAMSELECTION problem can be reduced from *node weight group steiner tree* problem (Khandekar, Kortsarz, and Nutov 2012) which admits a performance ratio of $O(|\mathcal{E}|^{1/2} \ln |\mathcal{E}|)$. Therefore, we have the following corollary.

Corollary 5 *If all teams are required to be connected, Approx-TG achieves*

$$\max\{1/O(|\mathcal{E}|^{1/2} \ln |\mathcal{E}|)\Delta), O(1/|\mathcal{E}|^{1/2} \ln |\mathcal{E}|)2\sqrt{m})\}$$

approximation ratio for TEAMGROUPING.

LP Rounding We next discuss how to round the factional solution of primal LP, this will be used as a subroutine in Algorithm 3. In the rest of our discussion, we say two teams are *adjacent* if they contain at least one common individual. We use $\mathcal{N}(C)$ to denote the adjacent teams of C. Let $\mathbf{C}^I$ denote the set of input teams, e.g., $\mathbf{C}^I$ refers to $\mathbf{C}^H$ (or $\mathbf{C}^{H_1}$ resp.) in Algorithm 1 (or Algorithm 2 resp.).

Our rounding method (Algorithm 4) can be described as follows:

Step 1: Sort all teams in $\mathbf{C}^I$ in non-decreasing order of their profit.

Step 2: Select the team $C_{ti} \in \mathbf{C}^I$ with the highest profit and add it to our final solution.

Step 3: Remove C_{ti} and $\mathcal{N}(C_{ti})$ from $\mathbf{C}^I$. This step ensures that no individual participates in multiple tasks.

Step 4: Goto Step 2 unless there are no teams left.

We next provide the approximation ratio of Algorithm 4.

Algorithm 4 Deterministic Rounding

1: Sort all teams in $\mathbf{C}^I$ in non-decreasing order of their profit.
2: **while** $\mathbf{C}^I \neq \emptyset$ **do**
3: Select the team with highest profit in $\mathbf{C}^I$, say C_{ti}
4: $\mathcal{C}^{DR} = \mathcal{C}^{DR} \cup \{C_{ti}\}$
5: $\mathbf{C}^I = \mathbf{C}^I \setminus \{C_{ti} \cup \mathcal{N}(C_{ti})\}$
6: **end while**
7: Return $\mathcal{C}^{DR}$

Lemma 3 *Let τ denote the size of the largest team in $\mathbf{C}^I$, Algorithm 4 achieves approximation ratio $1/\tau$.*

Extensions

Incorporating Heterogenous Load Limits

Our basic model assumes that each individual can only participate in *one* task. However, as mentioned earlier, different individuals may have different capabilities, i.e., , each individual u can participate in up to f_u number of tasks. In order to capture this scenario, we can simply create f_u copies of u with identical skill set, then all results developed previously can apply to the modified instance.

Theorem 6 *Approx-TG achieves $\max\{\mu/\Delta, \mu/2\sqrt{m}\}$ approximation ratio when individuals have heterogeneous load limits.*

Incorporating the Capacity Constraint of Each Task

Throughout this paper, we assume that each task can be performed unlimited number of times. However, this may not always hold in practice, take puzzle assembly as an example, this type of task can only be performed once. To this end, we add a group of additional constraints to the original problem: $\sum_{C_{ti} \in \mathcal{C}_t} x_{ti} \leq g_t, \forall t \in \mathcal{T}$ where g_t denotes the capacity of task $t \in \mathcal{T}$, i.e., task t can be performed up to g_t times.

P.2: *Maximize* $\sum_{C_{ti} \in \mathcal{C}_t \in \mathbf{C}} (x_{ti} \cdot \lambda_t)$
subject to:

$$\begin{cases} \sum_{u \in C_{ti} \in \mathcal{C}_t \in \mathbf{C}} x_{ti} \leq 1, \forall u \in \mathcal{V} \\ \sum_{C_{ti} \in \mathcal{C}_t} x_{ti} \leq g_t, \forall t \in \mathcal{T} \\ x_{ti} \in \{0,1\}, \forall C_{ti} \in \mathcal{C}_t \in \mathbf{C} \end{cases}$$

Similar to the one developed in the basic model, we propose a LP-Based Approximation Algorithm for **P.2**. The detailed description and analysis of our modified Approx-TG can be found in the full version.

Theorem 7 *The modified Approx-TG achieves $\max\{\mu/(\Delta+1), \mu/2(\sqrt{m}+1)\}$ approximation ratio for **P.2**.*

Performance Evaluation

In this section, we conduct an empirical evaluation of the proposed algorithms. All experiments were run on a machine with Intel Xeon 2.40GHz CPU and 64GB memory, running 64-bit RedHat Linux server. The goal of our experiments is multifold. First, we would like to evaluate the performance of the Team Grouping algorithms as measured by the total profit achieved. Second, we evaluate the extent to which the required number of skills and the connectivity constraint affect the quality of the solutions by measuring the cardinality of the teams produced by the algorithms. For extended version of the TEAMGROUPING problem, we evaluate the total profit achieved by the algorithms under different individual load limits and project capacity constraints.

Datasets. We evaluate the proposed algorithms on the real world benchmark dataset collected from *Upwork*. *Upwork* is a global freelancing platform where businesses and independent professionals connect and collaborate remotely. *Upwork* has 10 million registered freelancers and 4 million registered clients.

Social compatibility. We follow the definition of social compatibility introduced in (Lappas, Liu, and Terzi 2009): a group of individuals are social compatible as long as there exists a path between each pair of individuals in the team. Building a team of social compatible members can be done by solving an instance of the STEINER TREE problem as described in (Lappas, Liu, and Terzi 2009).

Algorithms. In addition to the algorithms we proposed, we also test with some straightforward heuristics that would be natural alternatives for solving the team grouping problem. The intuition behind these algorithms is to form a solution iteratively.

`Random`: A baseline algorithm that employs an iterative procedure. In each iteration, it selects a random project from the pool of projects $\mathcal{T}$, and then builds a team to cover the project with random selection. In order to fulfill the requirement of social compatibility, the algorithm needs to solve an instance of the STEINER TREE problem as described in (Lappas, Liu, and Terzi 2009) to ensure the connectivity of the team. The algorithm repeats this procedure until the individual pool is exhausted. For the extended version of the TG problem, it respects the capacity constraints of the tasks by moving on to the next task if the current task is already assigned to g_t teams.

`Greedy`: This is an algorithm that greedily picks the projects to be covered, one at a time, and finds the best set of individuals that can cover the selected project using the standard greedy approximation algorithm for set cover problem. In particular, it first ranks tasks w.r.t. profits and then selects teams for tasks using this order until the individual pool is exhausted. The project capacity constraints in extended version are taken care of in the same way as in `Random`. For each task t, the algorithm follows an iterative greedy procedure to build a team, adding at each step l the individual u that possesses the most yet uncovered required skills in t, an instance of the STEINER TREE problem needs to be solved to satisfy the social compatibility constraint.

`Greedy+`: This is an alternative version of `Greedy` which takes cost efficiency into account. It first ranks tasks w.r.t. *cost efficiency*, $\frac{\lambda_t}{|t|}$ (profit/number of skills required), and then builds teams for tasks using this order until individual load limit is exhausted. The rest of the procedure is the

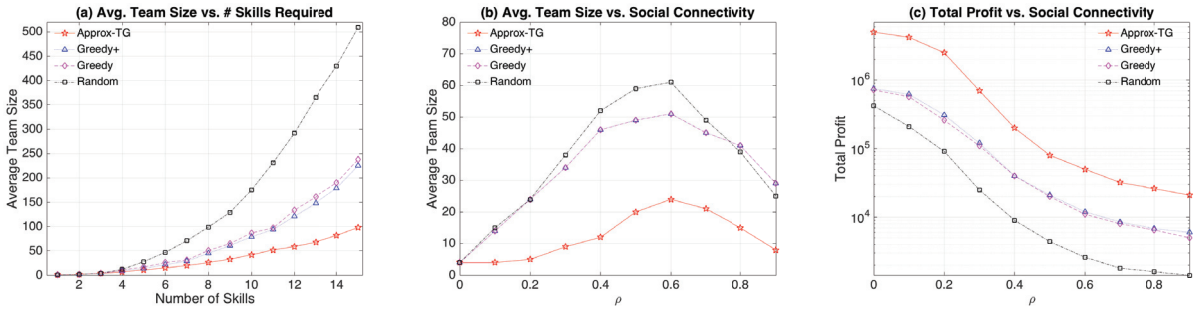

Figure 1: Performance of algorithms for basic Team Grouping problem.

same as in `Greedy`.

`Approx-TG`: An alias for Algorithm 3.

`Approx-TG+`: This is a modified version of `Approx-TG` to handle the capacity constraints of the tasks.

For `Random`, `Greedy` and `Greedy+` algorithms, we evaluate their output using Monte Carlo simulations with 10K runs and report average results in the following.

Evaluating Algorithms for Basic Team Grouping Problem

First, we focus on the evaluation of our algorithms for the basic TEAMGROUPING problem, i.e., `Random`, `Greedy`, `Greedy+` and `Approx-TG`. Recall that in the basic version of the problem, we have the individual load limit f_u set to 1, and the project capacity constraint g_t set to infinity (i.e., ∞). We report the total profit achieved by each algorithm, for increasing values of the social connectivity threshold, i.e., ρ ranging from 0 to 1. We also report the extent to which the required number of skills and the social connectivity threshold affect the average size of the teams produced by each algorithm.

Figure 1(c) shows the performance of different algorithms on the *Upwork* dataset, in terms of the dollar profit. The y-axis shows the profit achieved by each algorithm, and the x-axis shows the connectivity threshold (ρ) that was used to adjust the social compatibility constraint. As shown in Figure 1(c), we observe that, in all cases, `Approx-TG` outperforms all the other algorithms. We also notice that the total profit produced by each algorithm goes down as ρ increases. This is because as ρ increases all the edges with weight less than ρ are removed from the social graph, resulting in the disconnection of team members that were connected in original graph due to a sparse structure of the new graph. As a result, more individuals are selected as connecting nodes other than as functional nodes that actually utilize their skills on the projects, and the total profits achieved by the algorithms decrease accordingly.

Figure 1(a) illustrates how the average team sizes are affected by different project sizes (i.e., required number of skills). The y-axis shows the average team size produced by each algorithm, and the x-axis shows the required number of skills. In this set of experiments, we fix the connectivity threshold ρ to be 0. As shown in the figure, `Approx-TG` consistently produces small teams under all the skill set size values, and the average team size produced increases almost linearly with required number of skills. In contrast, `Random`, `Greedy` and `Greedy+` algorithms produce larger teams and the produced average team sizes increase exponentially as the required number of skills increases.

Figure 1(b) shows how the average team sizes are affected by different connectivity threshold (ρ) values. The y-axis shows the average team size produced by each algorithm, and the x-axis shows the value of ρ. In this set of experiments, we fix the required number of skills to be 3. In particular, projects that require exactly 3 skills are selected for this evaluation. We observe that as ρ increases, the average team sizes produced by all algorithms increase first and then decrease after ρ passes 0.6. It is reasonable that the average team sizes increase at the beginning because more nodes are required in the teams to fulfill the social compatibility requirement as ρ increases, since more edges are removed from the original graph. At the point ρ passes 0.6, the average team sizes are shown to be decreasing. The reason is that as a significant number of edges are removed from the graph, the graph is decomposed to a number of small components, resulting in a very sparse graph structure. Therefore, potential teams that can be built are restricted to these local small components, so as to ensure that the social compatibility constraint is satisfied in this case.

Conclusion

In this paper, we study the profit-driven team grouping problem. We aim to group individuals into different teams, and assign them to different tasks, such that the total profit of the tasks that can be performed is maximized. We consider three constraints when perform grouping, and present a LP-based approximation algorithm to tackle it. We also study several extensions of this problem. Although this paper studies team grouping problem, our results are general enough to tackle a broad range of problems that involve cover decomposition.

References

Anagnostopoulos, A.; Becchetti, L.; Castillo, C.; Gionis, A.; and Leonardi, S. 2012. Online team formation in social networks. In *Proceedings of the 21st international conference on World Wide Web*, 839–848. ACM.

Bagaria, V. K.; Pananjady, A.; and Vaze, R. 2013. Optimally approximating the coverage lifetime of wireless sensor networks. *arXiv preprint arXiv:1307.5230*.

Chvatal, V. 1979. A greedy heuristic for the set-covering problem. *Mathematics of operations research* 4(3):233–235.

Coase, R. H. 1937. The nature of the firm. *economica* 4(16):386–405.

Dorn, C., and Dustdar, S. 2010. Composing near-optimal expert teams: A trade-off between skills and connectivity. In *On the Move to Meaningful Internet Systems: OTM 2010*. Springer. 472–489.

Gajewar, A., and Sarma, A. D. 2012. Multi-skill collaborative teams based on densest subgraphs. In *SDM*, 165–176. SIAM.

Golshan, B.; Lappas, T.; and Terzi, E. 2014. Profit-maximizing cluster hires. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, 1196–1205. ACM.

Grötschel, M.; Lovász, L.; and Schrijver, A. 1981. The ellipsoid method and its consequences in combinatorial optimization. *Combinatorica* 1(2):169–197.

Kargar, M., and An, A. 2011. 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*, 985–994. ACM.

Kargar, M.; Zihayat, M.; and An, A. 2013. Finding affordable and collaborative teams from a network of experts. In *Proceedings of the SIAM International Conference on Data Mining (SDM)*, 587–595. SIAM.

Khandekar, R.; Kortsarz, G.; and Nutov, Z. 2012. Approximating fault-tolerant group-steiner problems. *Theoretical Computer Science* 416:55–64.

Lappas, T.; Liu, K.; and Terzi, E. 2009. Finding a team of experts in social networks. In *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, 467–476. ACM.

Li, C.-T., and Shan, M.-K. 2010. Team formation for generalized tasks in expertise social networks. In *Social Computing (SocialCom), 2010 IEEE Second International Conference on*, 9–16. IEEE.

Li, C.-T., and Shan, M.-K. 2012. Composing activity groups in social networks. In *Proceedings of the 21st ACM international conference on Information and knowledge management*, 2375–2378. ACM.

Lu, D. 2011. *Fundamentals of supply chain management*. Bookboon.

Pananjady, A.; Bagaria, V. K.; and Vaze, R. 2014. Maximizing utility among selfish users in social groups. In *Communications (NCC), 2014 Twentieth National Conference on*, 1–6. IEEE.

Sozio, M., and Gionis, A. 2010. The community-search problem and how to plan a successful cocktail party. In *Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining*, 939–948. ACM.