Submitted to
manuscript 2015

Profit-Driven Team Grouping in Social Networks

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

Jing Yuan
Department of Computer Science and Engineering, The University of North Texas

Tao Li, Yao Wang
Center of Intelligent Decision-making and Machine Learning, School of Management, Xi'an Jiaotong University

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 the compatibility between these individuals is captured by a social network. Moreover, there is a collection of tasks, where each task requires a specific set of skills and yields a profit upon completion. Individuals may collaborate with each other as *teams* to accomplish a set of tasks. We aim to find a group of teams to maximize the total profit of the tasks that they can complete. Any feasible grouping must satisfy the following conditions: (i) each team possesses all the skills required by the task assigned to it, (ii) individuals belonging to the same team are socially compatible, and (iii) no individual is overloaded. We refer to this as the TeamGrouping problem. We analyze the computational complexity of this problem and then propose a linear program-based approximation algorithm to address it and its variants. Although we focus on team grouping, our results apply to a broad range of optimization problems that can be formulated as cover decomposition problems.

Key words: approximation algorithm; team formation; cover decomposition

1. Introduction

In this paper, we address the team grouping problem in a networked community of people with diverse skill sets. We consider a setting where people possess different skills and the compatibility between these individuals is captured by a social network. We assume a collection of tasks where each task requires a specific set of skills and yields a profit upon completion. Individuals may collaborate with each other as *teams* to accomplish a set of tasks. We aim to find a grouping method that maximizes the total profit of the tasks they can complete. Relevant examples are available in the domain of online labor

markets, such as Freelancer (`www.freelancer.com`), Upwork (`www.upwork.com`), and Guru (`www.guru.com`). In these online platforms, freelancers with various skills can be hired to work on different types of projects. Instead of working purely independently, a growing number of freelancers are realizing the benefit of working as a team, with fellow freelancers who have complementary skills (Golshan et al. 2014). This allows them to expand their talent pool and better balance their workload. Many major platforms in this area, such as Upwork, provide team hiring services to their enterprise customers.

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 possessed by this individual, that is, $u \subseteq S$; these are the skills that the individual possesses. There is a set of tasks $\mathcal{T}$, and every task $t \in \mathcal{T}$ can also be represented by the set of skills required by this task (i.e., $t \subseteq \mathcal{S}$). Finally, every task t will yield a profit λ_t, which is the benefit that the completion of the task will yield for the platform. The team grouping problem (labeled TeamGrouping) is to group individuals into different teams and assign a task to each team in a manner that satisfies the following conditions: (i) each team possesses all the skills required by the task, (ii) individuals within the same team have high social compatibility, and (iii) no individual is overloaded. Our goal is to maximize the sum of profits from all the tasks that can be performed by these teams. Social compatibility between individuals can be interpreted in many ways. In this work, we model *social compatibility* by means of a social network in which the nodes represent individuals and an edge connecting two nodes denotes a social connection between the corresponding individuals. One popular indicator of social compatibility is *connectivity* (Lappas et al. 2009); therefore, each team must form a connected graph. Another important indicator of social compatibility is *diameter*, for example, according to (Anagnostopoulos et al. 2012), the induced graph of each team should have a small diameter. However, our results are not restricted to any specific indicator of social compatibility. Instead, we propose a general framework in which a socially compatible team is a subset of nodes of the graph for which the induced subgraph has some desirable property.

We next present a toy example of our problem. Assume there are three IT projects requiring different skills: the first task will yield profit $\lambda_1 = \$50$ and requires skills $t_1 = \{\text{HTML, MySQL, JavaScript, PHP}\}$, the second task will yield profit $\lambda_2 = \$10$ and requires skills $t_2 = \{\text{JavaScript, HTML}\}$, and the last task will yield profit $\lambda_3 = \$5$ and requires

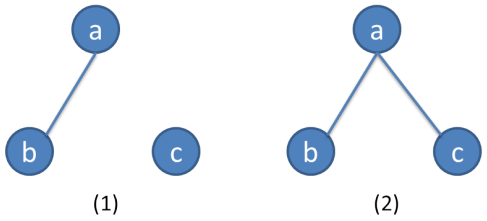

Figure 1 Two social networks.

skills $t_3 = \{\text{PHP}\}$. In addition, there are three individuals $\{a, b, c\}$ with the following skills: $a = \{\text{HTML, MySQL}\}$, $b = \{\text{JavaScript}\}$, and $c = \{\text{HTML, PHP}\}$. In our basic formulation, each individual can participate in only one team, and all team members must be connected. We consider the social networks illustrated in Fig. 1. The most profitable grouping approach in Fig. 1 (1) is to assign team $\{a, b\}$ to t_2, and team $\{c\}$ to t_3, which yields \$15 in profit. This is because a and b are connected while c is isolated. For the social network in Fig. 1 (2) by contrast, since the induced graph of all three individuals is connected, the most profitable grouping approach is to assign team $\{a, b, c\}$ to t_1, which yields \$50 in profit.

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:

- We show that this problem is $1/\ln m$-hard to approximate; that is, it is NP-hard to find a solution with approximation ratio larger than $1/\ln m$.
- We propose a linear program (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 the MincostTeamSelection problem (Definition 1). If there is no constraint on social compatibility, then this ratio reduces to $\max\{\ln n/n, \ln n/2\sqrt{m}\}$.
- We consider two extensions of the basic model. In the first extension, we consider a scenario where each task can only be performed a given number of times at most. We develop a $\max\{\mu/(\Delta+1), \mu/2(\sqrt{m}+1)\}$-approximate algorithm for this extension. In the second extension, we relax the assumption that each person can participate in only one task by allowing individuals to have different load limits. We develop a $\max\{\mu/(4\Delta), \mu/(8\sqrt{f_{\max} m})\}$-approximate algorithm for this extension, where $f_{\max}$ represents the largest number of tasks an individual can participate in.
- Although we focus on TeamGrouping, our results apply to other applications, such as the lifetime maximization problem in wireless networks (Bagaria et al. 2013), resource allocation and scheduling problems (Pananjady et al. 2014), and supply chain management

problems (Lu 2011). In this sense, this research contributes fundamentally to any problems that fall into the family of generalized cover decomposition problem.

The remainder of this paper is organized as follows. In Section 2, we review the literature on team formation and disjoint set cover. We introduce the formulation of our problem in Section 3. In Section 4, we present our LP-based approximation algorithms. We conduct extensive experiments in Section 5. The two extensions of the basic model are studied in Section 6. We summarize this study in Section 7. Most notations used in this paper are summarized in Table 1.

Table 1 Symbol table.

Notation	Meaning
n, m, k	Number of skills, individuals, tasks
Δ	Size of the largest minimal team
$\mathcal{C}$	Ground set of teams
$\mathcal{C}_t \in \mathcal{C}$	Set of teams covering task t
$C_{ti} \in \mathcal{C}_t$	ith team in $\mathcal{C}_t$
$1/\mu$	Approximation ratio of the MincostTeamSelection problem
x^*	(Approximate) Solution of primal LP
$\mathcal{N}(C)$	C's adjacent teams from $\mathcal{C}^I$
$\mathcal{C}(x^*)$	$\mathcal{C}(x^*) = \{C_{ti} \mid x^*_{ti} > 0\}$
$\mathcal{C}(x^*)_t$	$\mathcal{C}^H_t = \mathcal{C}(x^*) \cap \mathcal{C}_t$

2. 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 hiring problems. Lappas et al. (2009) introduced the minimum cost team formation problem. Given a set of skills to be covered and a social network, the objective is to select a team of experts that can cover all required skills, while ensuring efficient communication between team members. There is a considerable amount of literature on this topic and its variants (Kargar et al. 2013, Dorn and Dustdar 2010, Gajewar and Sarma 2012, Kargar and An 2011, Li and Shan 2010, Sozio and Gionis 2010). Golshan et al. (2014) studied the

cluster hiring problem, where the objective is to hire a profit-maximizing team of experts who can complete multiple projects within a fixed budget. The aforementioned studies aim to select a single team. By contrast, our objective is to group individuals into multiple teams. Nevertheless, our problem is closely related to the team formation problem, and we use their solution as a key component of our solution.

Another category of related work is the *maximum disjoint set cover* (DSCP) problem (Bagaria et al. 2013). Given a universe, and a set of subsets, the objective of this problem is to find as many set covers as possible such that all set covers are pairwise disjoint. Our problem can be considered a generalization of DSCP because every task in our problem may have different coverage requirements, capacity constraints, and profits. Moreover, every feasible set cover (team) in our problem must satisfy both coverage requirement and social compatibility. In addition, the requirement of "disjointness" is relaxed in our problem by allowing individuals to have different load limits.

3. Problem Formulation

Individuals. Skills. Tasks. Consider 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 possessed by this individual; that is, $u \subseteq \mathcal{S}$; these are the skills that the individual possesses. Every task $t \in \mathcal{T}$ can also be represented by the set of skills needed to complete the task (i.e., $t \subseteq \mathcal{S}$). In addition, each task $t \in \mathcal{T}$ has a profit λ_t. We assume that each task has an unlimited number of copies; that is, the same task can be performed by an unlimited number of teams. We relax this assumption in Section 6 by imposing a capacity constraint on each task.

Load. Our basic model assumes that each individual can participate in only *one* task. In Section 6, we relax this assumption by allowing individuals to have different load limits.

Teams. In practice, social compatibility between individuals plays an important role in teamwork. For example, low social compatibility or high coordination costs might degrade the organizational efficiency (Coase 1937). We model *social compatibility* by means of a social network $G = (\mathcal{V}, \mathcal{E})$, where the nodes in $\mathcal{V}$ represent individuals and an edge in $\mathcal{E}$ connecting two nodes denotes the social connection between the corresponding individuals. *Connectivity* is a widely known concept that captures the underlying social compatibility of a team. This follows the approach of Lappas et al. (2009) and requires that each team

form a connected graph. Another popular indicator of social compatibility is *diameter* (Anagnostopoulos et al. 2012); that is, the longest shortest path between team members in a social network is no longer than a given threshold. Nonetheless, our results are not restricted to any specific notations of social compatibility.

Problem Formulation. For a team of individuals $C \subseteq \mathcal{V}$, C is deemed to have skill s if there exists at least one individual $u \in C$ such that u has skill s, that is, $s \in u$. For a task $t \in \mathcal{T}$, team C is deemed to cover t if C (as a team) has all the skills required by t. A team of individuals may cover more than one task, but each individual can only participate in one of those tasks [1]. We define the set of qualified teams for a task $t \in \mathcal{T}$ to be the set of socially compatible teams covering t. That is,

$$\mathcal{C}_t = \{C \subseteq \mathcal{V} \mid C \text{ is socially compatible } \wedge C \text{ covers } t\}.$$

A minimal team for a task is a qualified team for this task that is not a superset of any other qualified team. In the rest of this paper, we only consider minimal teams. Let $\mathcal{C} = \cup_{t \in \mathcal{T}} \mathcal{C}_t$. The objective of the TeamGrouping problem is to select a group of teams from $\mathcal{C}$ such that each individual participates in only one team. We formally define the TeamGrouping problem in **P.1**. For each $t \in \mathcal{T}$ and $i \in \{0, 1, \cdots, |\mathcal{C}_t|\}$, let C_{ti} denote the ith team in $\mathcal{C}_t$. Let x_{ti} be an indicator of whether team C_{ti} is selected ($x_{ti} = 1$) or not ($x_{ti} = 0$).

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

subject to:

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

The first constraint specifies that each individual participates in at most one team. Recall that $|\mathcal{V}| = m$, the following results show that we cannot hope to achieve an approximation ratio of $\omega(1/\ln m)$ for this problem.

Theorem 1. *Let $m = |\mathcal{V}|$. **P.1** is $1/\ln m$-hard to approximate.*

Proof: For this proof, we consider a simplified version of **P.1**. There is only one task, that is, $k = 1$, and there is no constraint on social compatibility. We call this problem s-TeamGrouping. We next prove that the DSCP can be reduced to s-TeamGrouping.

[1] As mentioned earlier,this assumption will be relaxed in Section 6.

The formal definition of DSCP is as follows: Given a universe $\mathcal{U}$ and a set of subsets $\mathcal{X}$, the goal is to 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 to do 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 the maximum number of disjoint teams such that each team can cover all skills in $\mathcal{S}$. According to Bagaria et al. (2013), it is hard to achieve an approximation ratio of $\omega(1/\ln m)$ unless $NP \subseteq DTIME(n^{O(\ln \ln m)})$. Thus, **P.1**, which is a generalization of s-TeamGrouping, is also $1/\ln m$-hard to approximate. □

Bagaria et al. (2013) developed an $1/\ln m$-approximate algorithm for DSCP. For the special case of our problem where there is only one task and no constraint on social compatibility, we can simply adopt their method to achieve an approximation ratio of $1/\ln m$. In the following, we propose an LP-based approximation algorithm to address the general case.

4. LP-Based Approximation Algorithms

In this section, we give a $\max\{\mu/\Delta, \mu/2\sqrt{m}\}$-approximation algorithm for **P.1**, where $1/\mu$ is the approximation factor of the algorithm for the MincostTeamSelection problem, which is formally defined in Definition 1, and $\Delta := \max_{C\in\mathcal{C}} |C|$ is the size of the largest minimal team. Our algorithm consists of two phases: we first solve the LP relaxation of the original problem to obtain a fractional solution (Section 4.1) and then use this fractional solution to compute a group of teams (Section 4.2).

4.1. LP Relaxation

We first present the LP relaxation of **P.1**.

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

subject to:

$$\begin{cases} \sum_{C_{ti}\in\mathcal{C}:C_{ti}\ni u} x_{ti} \le 1, \forall u \in \mathcal{V} \\ x_{ti} \ge 0, \forall C_{ti} \in \mathcal{C} \end{cases}$$

This LP has m constraints (excluding the trivial constraints $x_{ti} \ge 0$). However, its number of variables is $\sum_{t\in\mathcal{T}} |\mathcal{C}_t|$, which can easily be exponential in the number of individuals. Hence, standard LP solvers cannot solve this packing LP effectively.

To address this challenge, we rely on the ellipsoid algorithm (Grötschel et al. 1981) and the dual problem (**Dual LP of P.1**). On a high level, we use the ellipsoid method to test

whether a given non-degenerate convex set S is empty or not. Here, S represents the feasibility region of the dual problem. This method starts with an ellipsoid that is guaranteed to contain S. In each iteration, it determines whether the center of the current ellipsoid is in S. If the answer is "yes,", then S is nonempty, which indicates that the current solution is feasible. In this case, the method tries a smaller ellipsoid that decreases the objective function. Otherwise, the method finds a violated constraint through an (approximate) separation oracle and tries a smaller ellipsoid whose center satisfies that constraint. Geometrically, we take a hyperplane through the center of the original ellipsoid such that S is contained in one of the two half-ellipsoids. We take the smallest ellipsoid completely containing this half-ellipsoid, whose volume is substantially smaller than the volume of the previous ellipsoid. This process iterates until the volume of the bounding ellipsoid is sufficiently small, in which case S is considered empty; that is, we cannot find a feasible solution with a smaller objective. This process takes a polynomial number of iterations for solving linear problems. We do not require an explicit description of LP to make this method work; we only need a polynomial-time (approximate) separation oracle to examine whether a point lies in S or not and, in the latter case, return a separating hyperplane.

Here, we formally introduce our algorithm. We refer to the LP relaxation of **P.1** as the primal LP. We next present **Dual LP of P.1**, the dual to the primal LP. In the dual problem, we assign a price $y(u)$ to each node $u \in \mathcal{V}$.

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

subject to:

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

We leverage the ellipsoid method for exponential-sized LP with an (approximate) separation oracle to solve this problem. In particular, in each iteration of the ellipsoid method, we solve the MincostTeamSelection problem approximately to obtain a polynomial-time approximate separation oracle to check the feasibility of the current solution.

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

MincostTeamSelection has been intensively studied in the literature, using various indicators of social compatibility. For example, if there is no requirement of social compatibility, then MincostTeamSelection reduces to the classical *weighted set cover problem* (Chvatal 1979), which admits an $O(\log n)$-factor approximation. (Lappas et al. 2009) proposed the use of connectivity as a measure of social compatibility; that is, all team members must be connected in a social network. In this context, the MincostTeamSelection problem can be reduced from *node weight group steiner tree* problem (Khandekar et al. 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. As stated by Anagnostopoulos et al. (2012), a team must have a bounded diameter. We next present the main theorem of this section. This theorem is not restricted to any specific indicator of social compatibility.

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

Proof: Let $\mathcal{A}$ be a $1/\mu$-approximation algorithm for MincostTeamSelection. We use $\mathcal{A}$ as an approximate separation oracle to examine whether the current solution to the dual problem is feasible or not. Let $S(L)$ denote the set of $y \in \mathbb{R}_+^{\mathcal{V}}$ satisfying that

$$\sum_{u\in\mathcal{V}} y(u) \le L,$$

$$\sum_{u\in C_{ti}} y(u) \ge \lambda_t, \forall C_{ti} \in \mathcal{C}.$$

We implement binary search to find the smallest value of L for which $S(L)$ is nonempty. For a given L, the method first checks the inequality $\sum_{u\in\mathcal{V}} y(u) \le L$. Then, it runs algorithm $\mathcal{A}$, using $y(u)$ as the price function to select the cheapest group $C_t \in \mathcal{C}_t$ for each task $t \in \mathcal{T}$. Suppose $\mathcal{A}$ is an exact algorithm, that is, $\mu = 1$. If for all t, $\sum_{u\in C_t} y(u) \ge \lambda_t$, then $y \in S(L)$. If there exists some t such that $\sum_{u\in C_t} y(u) < \lambda_t$, then $y \notin S(L)$ and C_t is a separating hyperplane. However, for general $\mu \le 1$, $C_t \in \mathcal{C}_t$ might not be the cheapest team for task $t \in \mathcal{T}$. Hence, $S(L)$ might actually be empty even if $\forall t, \sum_{u\in C_t} y(u) \ge \lambda_t$. Nonetheless, even for this general case, $\frac{1}{\mu}\cdot y \in S(\frac{1}{\mu}\cdot L)$. Let L^* be the minimum value of L for which the algorithm decides $S(L)$ is nonempty. We can conclude that $S(\frac{1}{\mu}\cdot L^*)$ is nonempty and $S(L^*-\epsilon)$ is empty, where ϵ is the precision of the algorithm. That is, the value of the dual

LP and thus the value of the primal LP belong to $[L^* - \epsilon, \frac{1}{\mu} \cdot L^*]$. Therefore, by finding a solution of value $L^* - \epsilon$ for the primal LP, we achieve an approximation ratio of μ against the optimal solution.

Here, we explain how to compute such a solution using only teams corresponding to the separating hyperplanes found by the separation oracle. Let $\mathcal{C}_t^H$ denote the subset of teams in $\mathcal{C}_t$ for which the dual constraint is violated in the implementation of the ellipsoid algorithm on $S(L^* - \epsilon)$. Then, $\sum_{t=1}^{k} |\mathcal{C}_t^H|$ is polynomial. Let $\mathcal{C}^H = \cup_{t \in \mathcal{T}} \mathcal{C}_t^H$, and consider the restricted dual LP.

$$\begin{aligned}
&\textit{Minimize } \textstyle\sum_{u \in \mathcal{V}} y(u) \\
&\textbf{subject to:} \\
&\qquad \begin{cases} \sum_{u \in C_{ti}} y(u) \geq \lambda_t, \forall C_{ti} \in \mathcal{C}^H \\ y(u) \geq 0, \forall u \in \mathcal{V}. \end{cases}
\end{aligned}$$

The value of the optimal solution to the above restricted dual LP is also at least L^*. Thus, we solve the following restricted primal LP of polynomial size, which is the dual of the restricted dual LP:

$$\begin{aligned}
&\textit{Maximize } \textstyle\sum_{C_{ti} \in \mathcal{C}^H} (x_{ti} \cdot \lambda_t) \\
&\textbf{subject to:} \\
&\qquad \begin{cases} \sum_{C_{ti} \in \mathcal{C}^H : u \ni C_{ti}} x_{ti} \leq 1, \forall u \in \mathcal{V} \\ x_{ti} \geq 0, \forall C_{ti} \in \mathcal{C}^H. \end{cases}
\end{aligned}$$

The value of the optimal solution of this restricted LP is at least L^*, which is a μ-approximation to the original primal LP. □

4.2. Approximation Algorithm

Before presenting our algorithm, we present a deterministic rounding method that converts any feasible solution of **Primal LP of P.1** to a feasible solution of **P.1**. Later, we use this rounding method as an essential subroutine to build our final algorithm.

4.2.1. LP Rounding Given any feasible solution $x^* = \{x_{ij}^* \mid C_{ti} \in \mathcal{C}\}$ of **Primal LP of P.1**, let $\mathcal{C}(x^*) = \{C_{ti} \mid x_{ti}^* > 0\}$ denote the set of all teams whose fractional value in x^* is positive. Two teams are considered *adjacent* if they contain at least one common individual. $\mathcal{N}(C, \mathcal{C}^I)$ denotes the set of all adjacent teams of C from a set of input teams $\mathcal{C}^I \subseteq \mathcal{C}(x^*)$, that is, $\mathcal{N}(C, \mathcal{C}^I) = \{C' \in \mathcal{C}^I \mid C' \neq C \wedge C \cap C' \neq \emptyset\}$. For simplicity, we use $\mathcal{N}(C)$ to denote $\mathcal{N}(C, \mathcal{C}^I)$ when it is clear from the context.

Our deterministic rounding method (Algorithm 1) takes a set of teams $\mathcal{C}^I \subseteq \mathcal{C}(x^*)$ as input.

Step 1: Select the team that has the highest profit from $\mathcal{C}^I$ (e.g., C_{ti},).

Step 2: Add C_{ti} to $\mathcal{C}^{DR}$ and remove $C_{ti} \cup \mathcal{N}(C_{ti})$ from $\mathcal{C}^I$. This step ensures that no individual participates in multiple tasks. Go to Step 1 unless there are no teams left. Output $\mathcal{C}^{DR}$.

Algorithm 1 Deterministic Rounding

Input: $\mathcal{C}^I \subseteq \mathcal{C}(x^*)$.

1: $\mathcal{C}^{DR} = \emptyset$
2: **while** $\mathcal{C}^I \neq \emptyset$ **do**
3: Select the team, say C_{ti}, that has the highest profit from $\mathcal{C}^I$.
4: $\mathcal{C}^{DR} = \mathcal{C}^{DR} \cup \{C_{ti}\}$.
5: $\mathcal{C}^I = \mathcal{C}^I \setminus \{C_{ti} \cup \mathcal{N}(C_{ti})\}$.
6: Return $\mathcal{C}^{DR}$.

Let $\rho(\mathcal{C}^I) = \max_{C_{ti} \in \mathcal{C}^I} |C_{ti}|$ denote the size of the largest team in $\mathcal{C}^I$. We next show that the profit of $\mathcal{C}^{DR}$ is at least $1/\rho(\mathcal{C}^I)$ faction of the one obtained from the fractional solution x^*.

LEMMA 1. *Given a feasible solution x^* of **Primal LP of P.1**, a set of input teams $\mathcal{C}^I \subseteq \mathcal{C}(x^*)$, $\sum_{C_{ti} \in \mathcal{C}^{DR}} \lambda_t \geq \sum_{C_{ti} \in \mathcal{C}^I} (x^*_{ti} \cdot \lambda_t)/\rho(\mathcal{C}^I)$, where $\rho(\mathcal{C}^I) = \max_{C_{ti} \in \mathcal{C}^I} |C_{ti}|$.*

Proof: Consider any team $C_{ti} \in \mathcal{C}^{DR}$. We have

$$x^*_{ti} \cdot \lambda_t + \sum_{C_{lj} \in \mathcal{N}(C_{ti})} (x^*_{lj} \cdot \lambda_l) \leq x^*_{ti} \cdot \lambda_t + \sum_{C_{lj} \in \mathcal{N}(C_{ti})} (x^*_{lj} \cdot \lambda_t) \tag{1}$$

$$= \lambda_t \times (x^*_{ti} + \sum_{C_{lj} \in \mathcal{N}(C_{ti})} x^*_{lj}) \tag{2}$$

$$\leq \lambda_t \times \sum_{u \in C_{ti}} \sum_{C_{lj} \in C_{ti} \cup \mathcal{N}(C_{ti}): C_{lj} \ni u} x^*_{lj} \tag{3}$$

$$\leq \lambda_t \times \sum_{u \in C_{ti}} \sum_{C_{lj} \in \mathcal{C}(x^*): C_{lj} \ni u} x^*_{lj} \tag{4}$$

$$\leq \lambda_t \times \sum_{u \in C_{ti}} 1 \tag{5}$$

$$\leq \rho(\mathcal{C}^I) \cdot \lambda_t. \tag{6}$$

The first inequality is due to C_{ti} having the highest profit among all its adjacent teams; the second inequality is due to the definition of $\mathcal{N}(C_{ti})$; the fourth inequality is due to x^* being a feasible solution of **Primal LP of P.1**, indicating that $\sum_{C_{lj}\in\mathcal{C}(x^*):C_{lj}\ni u} x^*_{lj} \leq 1, \forall u \in C_{ti}$; the last inequality is due to $\rho(\mathcal{C}^I) = \max_{C_{ti}\in\mathcal{C}^I} |C_{ti}|$.

Therefore, for any $C_{ti} \in \mathcal{C}^{DR}$, $\lambda_t \geq (x^*_{ti}\cdot\lambda_t + \sum_{C_{lj}\in\mathcal{N}(C_{ti})}(x^*_{lj}\cdot\lambda_l))/\rho(\mathcal{C}^I)$. Summation of this inequality over all teams from $\mathcal{C}^{DR}$ gives $\sum_{C_{ti}\in\mathcal{C}^{DR}}\lambda_t \geq \sum_{C_{ti}\in\mathcal{C}^{DR}}(x^*_{ti}\cdot\lambda_t + \sum_{C_{lj}\in\mathcal{N}(C_{ti})}(x^*_{lj}\cdot\lambda_l))/\rho(\mathcal{C}^I) = \sum_{C_{ti}\in\mathcal{C}^I}(x^*_{ti}\cdot\lambda_t)/\rho(\mathcal{C}^I)$. That is, the profit of $\mathcal{C}^{DR}$ is at least $1/\rho(\mathcal{C}^I)$ of the one obtained from the fractional solution x^*. □

4.3. Algorithm Design and Performance Analysis

We present our final algorithm, Approx-TG. In the rest of this paper, OPT denotes the profit gained from the optimal grouping. Approx-TG selects the better solution from two candidates as the final output. We present these two candidate solutions in detail.

Candidate Solution I: In Algorithm 2, we directly apply the deterministic rounding (Algorithm 1) to $\mathcal{C}(x^*)$, that is, we feed $\mathcal{C}^I = \mathcal{C}(x^*)$ as input teams to Algorithm 1. We prove that if x^* is a μ-approximate solution of **Primal LP of P.1** that is found by the ellipsoid method, then Algorithm 2 achieves an approximation ratio of μ/Δ, where Δ denotes the size of the largest minimal team.

Algorithm 2 Candidate Grouping - I

Input: x^*.

1: Apply deterministic rounding (Algorithm 1) to x^* and output a group of teams.

Lemma 2. *Assume that x^* is a μ-approximate solution of **Primal LP of P.1** that is found by the ellipsoid method. Then Algorithm 2 achieves an approximation ratio of μ/Δ for **P.1**.*

Proof: By Lemma 1, Algorithm 1 takes $\mathcal{C}(x^*)$ as input and returns a grouping that achieves a profit of at least $\frac{1}{\Delta}\sum_{C_{ti}\in\mathcal{C}(x^*)}(x^*_{ti}\cdot\lambda_t)$, where Δ is the size of the largest possible team in $\mathcal{C}(x^*)$. By the assumption that x^* is a μ-approximate solution of **Primal LP of P.1**, we have $\sum_{C_{ti}\in\mathcal{C}(x^*)}(x^*_{ti}\cdot\lambda_t) \geq \mu\cdot OPT$. Thus, Algorithm 2 achieves a profit of at least

$$\frac{1}{\Delta}\sum_{C_{ti}\in\mathcal{C}(x^*)}(x^*_{ti}\cdot\lambda_t) \geq \frac{\mu}{\Delta}\cdot OPT.$$

□

Candidate Solution II: Let $\mathcal{C}(x^*)_t = \mathcal{C}(x^*) \cap \mathcal{C}_t$ denote the subset of $\mathcal{C}(x^*)$ that is assigned to task $t \in \mathcal{T}$. Hence, $\mathcal{C}(x^*) = \cup_{t\in\mathcal{T}}\mathcal{C}(x^*)_t$. The framework of the second candidate solution (Algorithm 3) is summarized as follows:

Step 1: For every task $t \in \mathcal{T}$, we first partition $\mathcal{C}(x^*)_t$ into the disjoint subsets $\mathcal{C}(x^*)^1_t$ and $\mathcal{C}(x^*)^2_t$ such that $\forall C \in \mathcal{C}(x^*)^1_t : |C| \le \sqrt{m}$ and $\forall C \in \mathcal{C}(x^*)^2_t : |C| > \sqrt{m}$. That is, $\mathcal{C}(x^*)^1_t$ (resp. $\mathcal{C}(x^*)^2_t$) contains all teams with no more (resp. less) than $\sqrt{m}$ individuals. Let $\mathcal{C}(x^*)^1 = \cup_{t\in\mathcal{T}}\mathcal{C}(x^*)^1_t$ and $\mathcal{C}(x^*)^2 = \cup_{t\in\mathcal{T}}\mathcal{C}(x^*)^2_t$.

Step 2: Apply deterministic rounding (Algorithm 1) to $\mathcal{C}(x^*)^1$ to obtain a group of teams $\widetilde{\mathcal{C}}$.

Step 3: Select a team from $\mathcal{C}(x^*)^2$ whose task $t_{\max}$ has the highest profit $\lambda_{t_{\max}}$ (e.g., $C_{t_{\max}}$).

Step 4: Output the better solution between $\widetilde{\mathcal{C}}$ and $\{C_{t_{\max}}\}$ as the final output; that is, the profit of the returned solution is $\max\{\sum_{C_{ti}\in\widetilde{\mathcal{C}}}\lambda_t, \lambda_{t_{\max}}\}$.

Algorithm 3 Candidate Grouping - II

Input: x^*.

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

We next prove that if x^* is a μ-approximate solution of **Primal LP of P.1** that is found by the ellipsoid method, then the approximation ratio of Algorithm 3 can be bounded by $\mu/(2\sqrt{m})$.

LEMMA 3. *Assume that x^* is a μ-approximate solution of **Primal LP of P.1** that is found by the ellipsoid method. Algorithm 3 achieves an approximation ratio of $\mu/(2\sqrt{m})$ for **P.1**.*

Proof: To prove this lemma, we show that $\max\{\sum_{C_{ti}\in\widetilde{\mathcal{C}}}\lambda_t, \lambda_{t_{\max}}\} \ge \frac{1}{\sqrt{m}} \cdot \frac{\mu}{2} \cdot OPT$.

We first bound the gap between the profit gained from $\widetilde{\mathcal{C}}$ and $\sum_{C_{ti}\in\mathcal{C}(x^*)^1}(x^*_{ti} \cdot \lambda_t)$. By Lemma 1, we have

$$\sum_{C_{ti}\in\widetilde{\mathcal{C}}} \lambda_t \ge \frac{1}{\rho} \cdot \sum_{C_{ti}\in\mathcal{C}(x^*)^1} (x^*_{ti} \cdot \lambda_t) \ge \frac{1}{\sqrt{m}} \cdot \sum_{C_{ti}\in\mathcal{C}(x^*)^1} (x^*_{ti} \cdot \lambda_t), \tag{7}$$

where the second inequality is due to the assumption that $\rho \leq \sqrt{m}$ holds for all teams from $\mathcal{C}(x^*)^1$.

We next bound the gap between the profit gained from $C_{t_{\max}}$ and $\sum_{C_{ti} \in \mathcal{C}(x^*)^2}(x^*_{ti} \cdot \lambda_t)$. In particular, we show that

$$\lambda_{t_{\max}} \geq \frac{1}{\sqrt{m}} \cdot \sum_{C_{ti} \in \mathcal{C}(x^*)^2} (x^*_{ti} \cdot \lambda_t). \tag{8}$$

The following chain proves this inequality:

$$\sum_{C_{ti} \in \mathcal{C}(x^*)^2} (x^*_{ti} \cdot \lambda_t) \leq \sum_{C_{ti} \in \mathcal{C}(x^*)^2} (x^*_{ti} \cdot \lambda_{t_{\max}}) = \lambda_{t_{\max}} \cdot \sum_{C_{ti} \in \mathcal{C}(x^*)^2} x^*_{ti} \leq \lambda_{t_{\max}} \cdot \frac{m}{\sqrt{m}}. \tag{9}$$

The first inequality is due to the assumption that $C_{t_{\max}}$ delivers the highest profit among $\mathcal{C}(x^*)^2$. We then prove the second inequality. Because x^* is a feasible solution of **Primal LP of P.1** and $\mathcal{C}(x^*)^2 \subseteq \mathcal{C}$, we have $\sum_{C_{ti} \in \mathcal{C}(x^*)^2 : C_{ti} \ni u} x^*_{ti} \leq 1, \forall u \in \mathcal{V}$. Therefore,

$$\sum_{C_{ti} \in \mathcal{C}(x^*)^2} (x^*_{ti} \cdot |C_{ti}|) \leq m. \tag{10}$$

All teams in $\mathcal{C}(x^*)^2$ contain at least $\sqrt{m}$ individuals, so $\sum_{C_{ti} \in \mathcal{C}(x^*)^2}(x^*_{ti} \cdot |C_{ti}|) \geq \sum_{C_{ti} \in \mathcal{C}(x^*)^2}(x^*_{ti} \cdot \sqrt{m})$. This, together with (10), implies that $\sum_{C_{ti} \in \mathcal{C}(x^*)^2}(x^*_{ti} \cdot \sqrt{m}) \leq m$; thus, $\sum_{C_{ti} \in \mathcal{C}(x^*)^2}(x^*_{ti}) \leq \sqrt{m}$. This finishes the proof of the second inequality.

By the assumption that x^* is a μ-approximate solution of **Primal LP of P.1**, we have

$$\sum_{C_{ti} \in \mathcal{C}(x^*)} (x^*_{ti} \cdot \lambda_t) \geq \mu \cdot OPT \Rightarrow \sum_{C_{ti} \in \mathcal{C}(x^*)^1} (x^*_{ti} \cdot \lambda_t) + \sum_{C_{ti} \in \mathcal{C}(x^*)^2} (x^*_{ti} \cdot \lambda_t) \geq \mu \cdot OPT. \tag{11}$$

We now prove this theorem.

$$\max\{\sum_{C_{ti} \in \widetilde{\mathcal{C}}} \lambda_t, \lambda_{t_{\max}}\} \geq \frac{\sum_{C_{ti} \in \widetilde{\mathcal{C}}} \lambda_t + \lambda_{t_{\max}}}{2} \tag{12}$$

$$\geq \frac{1}{\sqrt{m}} \cdot \frac{\sum_{C_{ti} \in \mathcal{C}(x^*)^1}(x^*_{ti} \cdot \lambda_t) + \sum_{C_{ti} \in \mathcal{C}(x^*)^2}(x^*_{ti} \cdot \lambda_t)}{2} \tag{13}$$

$$\geq \frac{1}{\sqrt{m}} \cdot \frac{\mu}{2} \cdot OPT \tag{14}$$

The second inequality is due to (7) and (8), and the third inequality is due to (11). □

Putting It All Together. Given solutions returned from Algorithms 2 and 3, Approx-TG returns the one with the higher profit as the final output. Lemmas 2 and 3 jointly imply our main theorem.

THEOREM 3. *Approx-TG achieves an approximation ratio of* $\max\{\mu/\Delta, \mu/2\sqrt{m}\}$ *for* ***P.1****.*

Consider a special case of TEAMGROUPING where there is no requirement of social compatibility. In this case, the MINCOSTTEAMSELECTION problem reduces to the classical *weighted set cover problem*, which admits an $\ln n$ approximation. In addition, $\Delta \leq n$, because the number of possible skills is at most n, and if there is no constraint on social compatibility, then any minimal team contains at most n individuals. Corollary 1 holds by replacing μ with $\ln n$, and Δ with n in Theorem 3.

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

In practice, $n \ll m$; that is, the number of skills is much smaller than the number of individuals, so the above approximation ratio can be further rewritten as $\ln n/n$.

Consider a special case that uses connectivity as an indicator of social compatibility. As discussed in Section 4.1, in this setting, the MINCOSTTEAMSELECTION problem reduces to a *node weight group Steiner tree* problem (Khandekar et al. 2012), which admits a performance ratio of $O(|\mathcal{E}|^{1/2} \ln |\mathcal{E}|)$. Therefore, we have Corollary 2.

COROLLARY 2. *If all teams are required to be connected, Approx-TG achieves an approximation ratio of* $\max\{O(|\mathcal{E}|^{1/2} \ln |\mathcal{E}|)/\Delta), O(|\mathcal{E}|^{1/2} \ln |\mathcal{E}|)/2\sqrt{m})\}$ *for* ***P.1****.*

5. Performance Evaluation

In this section, we conduct simulations to evaluate the performance of our algorithm. All experiments were run 10 times on a desktop with Intel(R) Xeon(R) Gold 5218R CPU @ 2.1GHz and 94GB memory, running 64-bit Linux server. We show that our algorithm outperforms three benchmarks, and we also validate its robustness under various settings.

5.1. Setting

The input of our basic setting is composed of a set of 10 skills, a set of 20 tasks and a set of 100 individuals. We set the profit of each task to

$$\#\text{ skills required by a task } \times r,$$

where r is a random number chosen from $\{1, 2, 3\}$. We consider two scenarios as follows:

Scenario 1: In the first scenario, we assume there is no constraint on social compatibility. Hence, MincostTeamSelection reduces to the weighted set cover problem. We use greedy algorithm Slavık (1997) to solve this problem to obtain a $\ln n$-approximation solution, where n is the number of elements to be covered.

Scenario 2: In the second scenario, we incorporate the constraint of social compatibility. In particular, we use *connectivity* (Lappas et al. 2009) as an indicator of social compatibility; therefore, each team must form a connected graph. We generate a random network consisting of 1000 edges in the basic setting such that the connecting density of this network is 0.202. We add more edges to the network in the robustness section. In this scenario, our MincostTeamSelection problem reduces to the group steiner tree problem, and we use the ImprovAPP algorithm from (Sun et al. 2021) to solve this problem. It has been shown that this algorithm achieves a $(|\Gamma| - 1)$-approximation ratio, where $|\Gamma|$ denotes the number of vertex groups.

5.2. Benchmark

We compare our algorithm with three heuristics.

- Random: In each iteration, Random selects a random task, and then builds up a team randomly to cover this task. We remove all selected individuals from consideration in the subsequent iterations. This process continues until the individual pool is exhausted or the remaining individuals can not cover any of the tasks.
- Greedy: Greedy first sorts all tasks in non-increasing order of their profits. Then starting with the first task (e.g., t), Greedy selects a group of teams that covers t sequentially, where each team is selected by solving a weighted set cover problem (resp. the group steiner tree problem) using the greedy algorithm (resp. the ImprovAPP algorithm) in the basic setting (resp. the general setting). If we can not find more teams to cover t, then we move to the next task in the list. This process iterates until the individual pool is exhausted or we can not find more teams to cover the last task in the list.
- Greedy+: Unlike Greedy which ranks tasks according to their profits, Greedy+ ranks tasks according to $\frac{\lambda_t}{|t|}$, the ratio of profit and the number of skills required by a task t. The rest of the procedure is identical to Greedy.

5.3. Results

In this section, we report the performance of four algorithms under the basic setting. Figure 2 shows the statistics of 10-times-running without considering social compatibility, Figure 3 shows the statistics of 10-times-running subject to the constraint of social compatibility.

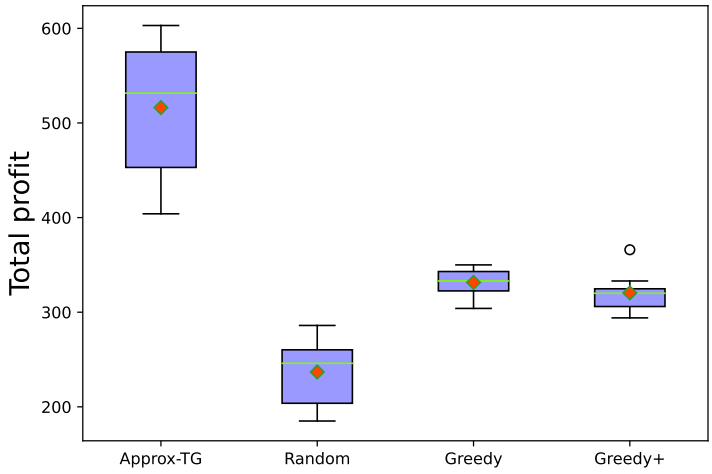

Figure 2 **without social compatibility**

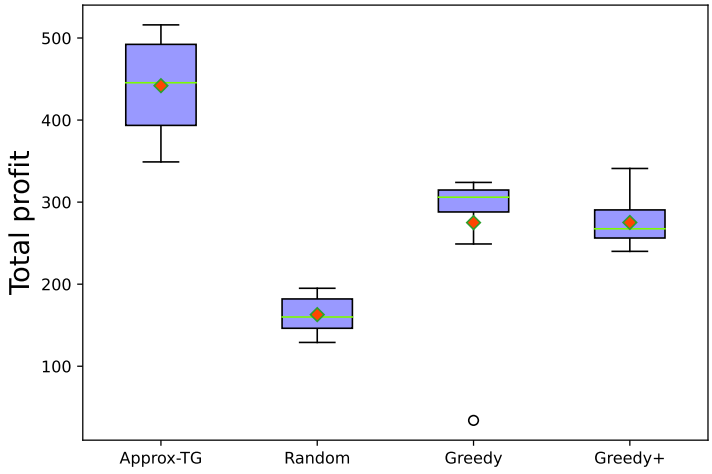

Figure 3 **with social compatibility**

Table 2 lists the mean total profit of 10 times and the improvement of other three algorithms over random policy.

Table 2 Profit comparison of four algorithms: mean(improvement against random policy)

Setting	Random	Greedy	Greedy+	Approx-TG
without social compatibility	236.9	331.5(40%)	320.4(35%)	516.3(118%)
with social compatibility	162.9	275.0(69%)	275.2(69%)	441.9(171%)

Our algorithm outperforms three benchmark solutions under all settings. By treating Random as a baseline, Greedy, Greedy+ and our algorithm increase the total profit by 40%, 35% and 118%, respectively, without considering social compatibility. If we consider the constraint of social compatibility, Greedy, Greedy+ and our algorithm increase the total profit by 69%, 69% and 171%, respectively. The absolute profits achieved by all algorithms decrease as we consider the constraint of social compatibility. This is because considering social compatibility adds additional constraints for finding a feasible solution.

Table 3 Profit comparison of four algorithms: mean(improvement against random policy) under different scenarios

index	scenario	Random	Greedy	Greedy+	Approx-TG
①	(-,10,20,100)	236.9	331.5(40%)	320.4(35%)	516.3(118%)
②	(-,10,20,50)	96.0	165.5 (72%)	150.6(57%)	230.3 (140%)
③	(-,10,20,200)	464.5	668.5 (43%)	614.8(32%)	973.1(109%)
④	(-,10,10,100)	215.3	308.0(43%)	286.8(33%)	362.3(68%)
⑤	(-,10,50,100)	236.7	348.5(47%)	335.9(42%)	615.4(160%)
⑥	(-,20,20,100)	204.1	271.3(32%)	252.5(23%)	337.3(65%)

Table 4 Profit comparison of four algorithms: mean(improvement against random policy) under different network scenarios

index	scenario	Random	Greedy	Greedy+	Approx-TG
⑦	(1000,10,20,100)	162.9	275.0(69%)	275.2(69%)	441.9(171%)
⑧	(2000,10,20,100)	186.8	353.0(89%)	313.0(68%)	474.5(154%)
⑨	(3000,10,20,100)	210.2	385.0(83%)	337.6(61%)	479.8(128%)

5.4. More Results on Robustness Check

We conduct additional experiments to validate the robustness of our algorithm. We denote the scenario that involves c_1 edges, c_2 skills, c_3 tasks and c_4 individuals as (c_1, c_2, c_3, c_4), and denote with $(-, c_2, c_3, c_4)$ the same scenario without considering the social compatibility. We vary the number of edges, the number of skills, the number of tasks and the number of individuals and report the mean total profit of 10-times-running in Table 3 and 4.

Random performs the worst in all settings, and our algorithm achieves the largest improvement over Random. Table 3 shows the results without social compatibility. We report the results under the baseline setting $(-, 10, 20, 100)$ in ①. We vary the number of individuals from 50 to 200 and report the results in ② and ③ respectively. As the same task can be performed by an arbitrary number of teams, a larger pool of individuals leads to a higher profit. Our algorithm achieves a profit more than twice that of Random, regardless of the number of individuals. We vary the number of tasks in ④ and ⑤. We find that the number of tasks has little impact on the performance of three benchmarks. Both Greedy and Greedy+ tend to select tasks with higher profits. Approx-TG achieves an improvement of 160% when the number of tasks is large. We increase the number of skills to 20 in ⑥. While the performance of all algorithms decline, our algorithm still performs the best.

Considering the constraint of social compatibility, we conduct experiments under different number of edges and report our results in Table 4. We set the number of edges to 2000 and 3000 in ⑧ and ⑨, respectively. As we add more edges to the network, it is easier to form a feasible team for a given task, this improves the profit of all algorithms. Our algorithm still performs the best, i.e., it achieves a profit more than twice that achieved by Random.

6. Extensions

6.1. Incorporation of the Capacity Constraint of Each Task

So far, we have assumed that each task can be performed an unlimited number of times. However, this may not always hold in practice. For example, puzzle assembly 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 $t \in \mathcal{T}$; that is, each task $t \in \mathcal{T}$ can be performed up to g_t times. We formally define this extension in **P.2**.

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

subject to:

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

Similar to the LP-Based algorithm developed in Section 4, we propose a LP-Based algorithm for **P.2**.

LP Relaxation The primal LP of **P.2** can be formulated as follows.

Primal LP of P.2: *Maximize* $\sum_{C_{ti}\in\mathcal{C}}(x_{ti}\cdot\lambda_t)$

subject to:

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

The dual to the above primal LP assigns a price $y(u)$ to each node $u\in V$ and a price $p(t)$ to each task $t\in\mathcal{T}$.

Dual LP of P.2: *Minimize* $\sum_{u\in\mathcal{V}} y(u)+\sum_{t\in\mathcal{T}}(p(t)\cdot g_t)$

subject to:

$$\begin{cases}\sum_{u\in C_{ti}} y(u)+p(t)\ge\lambda_t,\forall C_{ti}\in\mathcal{C}\\ y(u)\ge 0,\forall u\in\mathcal{V};p(t)\ge 0,\forall t\in\mathcal{T}\end{cases}$$

Similar to the solution for **P.1**, we run the ellipsoid algorithm on the dual LP using algorithm $\mathcal{A}$, an approximation algorithm for MincostTeamSelection, as the approximate separation oracle. More precisely, let $S(L)$ denote the set of $y\in\mathbb{R}_+^{\mathcal{V}}$ satisfying that

$$\sum_{u\in\mathcal{V}} y(u)+\sum_{t\in\mathcal{T}}(p(t)\cdot g_t)\le L,$$

$$\sum_{u\in C_{ti}} y(u)+p(t)\ge\lambda_t,\forall C_{ti}\in\mathcal{C}.$$

We adopt binary search to find the smallest value of L for which $S(L)$ is nonempty. The separation oracle works as follows: First, it checks the inequality $\sum_{u\in\mathcal{V}} y(u)+\sum_{t\in\mathcal{T}}(p(t)\cdot g_t)\le L$. Next, it runs the algorithm $\mathcal{A}$ on each task $t\in\mathcal{T}$ and selects a group $C_{ti}\in\mathcal{C}_t,\forall t\in\mathcal{T}$, using $y(u)$ as the price function. If for all C_{ti}, the cost of C_{ti} is larger than $\lambda_t-p(t)$, then $y\in S(L)$. If there exists some C_{ti} whose cost is less than $\lambda_t-p(t)$, then $y\notin S(L)$ and C_{ti} gives us a separating hyperplane. Based on similar analysis in Section 4, we have the following theorem.

THEOREM 4. *If there is a polynomial μ-approximation algorithm for* MINCOSTTEAMSELECTION*, then there exists a polynomial μ-approximation algorithm for* ***Primal LP of P.2****.*

LP Rounding We present a deterministic rounding algorithm (Algorithm 4). Given a feasible solution x^* of **Primal LP of P.2**, Algorithm 4 takes $\mathcal{C}^I$, a subset of $\mathcal{C}(x^*)$, as input, and outputs a group of teams from $\mathcal{C}^I$ such that (1) each individual participates in at most one team and (2) the same task t is performed by at most g_t teams for each task $t \in \mathcal{T}$. We next provide a summary of Algorithm 4:

Initially, let $\mathcal{C}^{DR} = \emptyset, z = x^*$.

Step 1: Select the team with the highest profit from $\mathcal{C}^I$ (e.g., C_{ti}).

Step 2: Let $\mathcal{C}_t^I = \mathcal{C}_t \cap \mathcal{C}^I$ denote the set of all teams in $\mathcal{C}^I$ that is assigned to task t. Reduce the value of z_{tj} for some $C_{tj} \in \mathcal{C}_t^I \setminus \{C_{ti}\}$ to some non-negative value such that

$$\sum_{C_{tj} \in \mathcal{C}_t^I \setminus \{C_{ti}\}} z_{tj} \text{ is reduced by } \min\{\sum_{C_{tj} \in \mathcal{C}_t^I \setminus \{C_{ti}\}} z_{tj}, 1 - z_{ti}\} \tag{15}$$

This can be done in an arbitrary way. For example, one can select an arbitrary team, say C_{tq}, from $\mathcal{C}_t^I \setminus \{C_{ti}\}$, reduce z_{tq} to its smallest non-negative value (zero, if necessary) such that the cumulative amount of reduction does not exceed $\min\{\sum_{C_{tj} \in \mathcal{C}_t^I \setminus \{C_{ti}\}} z_{tj}, 1 - z_{ti}\}$. Then we select another team from $\mathcal{C}_t^I \setminus \{C_{ti}\}$ and reduce its fractional value in the same manner. This process iterates until condition (15) is satisfied; that is, we terminate this process once the cumulative amount of reduction reaches $\min\{\sum_{C_{tj} \in \mathcal{C}_t^I \setminus \{C_{ti}\}} z_{tj}, 1 - z_{ti}\}$.

Step 3: Recall that $\mathcal{N}(C_{ti})$ denotes the set of all adjacent teams of C_{ti} from $\mathcal{C}^I \subseteq \mathcal{C}(x^*)$. Remove $\mathcal{N}(C_{ti}) \cup \{C_{ti}\}$ and $\{C_{tj} \in \mathcal{C}_t^I \mid z_{tj} = 0\}$ from $\mathcal{C}^I$. It will become clear later that this step ensures that no individual participates in multiple tasks and meanwhile, each task is assigned to at most g_t teams.

Step 4: Go to Step 1 unless $\mathcal{C}^I$ becomes empty. Output $\mathcal{C}^{DR}$.

Let $\mathcal{C}^{DR}$ denote the output of Algorithm 4, we first show that $\mathcal{C}^{DR}$ is a feasible solution to **P.2**.

LEMMA 4. *Let $\mathcal{C}^{DR}$ denote the set of groups returned from the deterministic rounding (Algorithm 4). $\mathcal{C}^{DR}$ is a feasible solution to* ***P.2****.*

Proof: First, by the design of Algorithm 4, once a team is selected, we remove all its adjacent teams from consideration. Hence, in the final solution $\mathcal{C}^{DR}$, each individual participates in

Algorithm 4 Deterministic Rounding

Input: $x^*, \mathcal{C}^I \subseteq \mathcal{C}(x^*)$.

1: $\mathcal{C}^{DR} = \emptyset, z = x^*$.
2: **while** $\mathcal{C}^I \neq \emptyset$ **do**
3: Select the team that has the highest profit from $\mathcal{C}^I$ (e.g., C_{ti}).
4: $\mathcal{C}^{DR} = \mathcal{C}^{DR} \cup \{C_{ti}\}$.
5: Reduce z_{tj} for some $C_{tj} \in \mathcal{C}_t^I \setminus \{C_{ti}\}$ to satisfy condition (15).
6: Remove $\mathcal{N}(C_{ti}) \cup \{C_{ti}\}$ and $\{C_{tj} \in \mathcal{C}_t^I \mid z_{tj} = 0\}$ from $\mathcal{C}^I$.
7: Output $\mathcal{C}^{DR}$.

at most one team. We next show that $\mathcal{C}^{DR}$ satisfies the capacity constraint of each task. To prove this, we show that for each task $t \in \mathcal{T}$, $\mathcal{C}_t^I$ becomes empty after adding at most g_t number of teams from $\mathcal{C}_t$ to $\mathcal{C}^{DR}$, indicating that no more teams from $\mathcal{C}_t$ will be added to the solution. Assume by contradiction that after selecting a group $\mathcal{C}'$ of g_t teams from $\mathcal{C}_t^I$,

$$\sum_{C_{ti} \in \mathcal{C}_t^I \setminus \mathcal{C}'} z_{ti} > 0 \tag{16}$$

Recall that after selecting a team C_{ti}, we reduce the value of $\sum_{C_{tj} \in \mathcal{C}_t^I \setminus \{C_{ti}\}} z_{tj}$ by an amount of $\min\{\sum_{C_{tj} \in \mathcal{C}_t^I \setminus \{C_{ti}\}} z_{tj}, 1 - z_{ti}\}$. Because of (16), we conclude that the cumulative amount of reduction in z due to the selection of $\mathcal{C}'$ is exactly $\sum_{C_{ti} \in \mathcal{C}'} (1 - x_{ti}^*)$. It follows that

$$\sum_{C_{ti} \in \mathcal{C}_t} x_{ti}^* \geq \sum_{C_{ti} \in \mathcal{C}'} 1 + \sum_{C_{ti} \in \mathcal{C}_t^I \setminus \mathcal{C}'} z_{ti} \tag{17}$$

$$\geq g_t + \sum_{C_{ti} \in \mathcal{C}_t^I \setminus \mathcal{C}'} z_{ti} \tag{18}$$

$$> g_t, \tag{19}$$

where the second inequality is due to the assumption that $|\mathcal{C}'| = g_t$ and the third inequality is due to (16). This contradicts to the assumption that x^* is a feasible solution to **Primal LP of P.2**; that is, x^* violates the second set of constraints listed in **Primal LP of P.2**. □

We next show that the profit of $\mathcal{C}^{DR}$ is at least $1/(\rho + 1)$ of the one obtained from the fractional solution x^*.

LEMMA 5. *Given a feasible solution x^* of* ***Primal LP of P.2****, a set of input teams $\mathcal{C}^I \subseteq \mathcal{C}(x^*)$, let $\mathcal{C}^{DR}$ denote the set of groups returned from the deterministic rounding (Algorithm 4), $\sum_{C_{ti}\in\mathcal{C}^{DR}} \lambda_t \geq \sum_{C_{ti}\in\mathcal{C}(x^*)}(x^*_{ti}\cdot\lambda_t)/(\rho+1)$, where $\rho = \max_{C_{ti}\in\mathcal{C}^I}|C_{ti}|$.*

Proof: Consider an arbitrary team from $\mathcal{C}^{DR}$ (e.g., C_{ti}). According to the design of Algorithm 4, after selecting C_{ti}, we perform the following two operations that may cause profit loss: (a) remove $\mathcal{N}(C_{ti})\cup\{C_{ti}\}$, and (b) reduce the value of $\sum_{C_{tj}\in\mathcal{C}^I_t\setminus\{C_{ti}\}} z_{tj}$ by an amount of $\min\{\sum_{C_{tj}\in\mathcal{C}^I_t\setminus\{C_{ti}\}} z_{tj}, 1-z_{ti}\}$. We next bound the profit loss due to these two operations separately.

First, by the design of Algorithm 4, C_{ti} has the highest profit among $\mathcal{N}(C_{ti})\cup\{C_{ti}\}$. Following the same proof of (6), we can bound the amount of profit loss $z_{ti}\cdot\lambda_t + \sum_{C_{lj}\in\mathcal{N}(C_{ti})}(z_{li}\cdot\lambda_l)$ due to the removal of $\mathcal{N}(C_{ti})\cup\{C_{ti}\}$ as follows:

$$z_{ti}\cdot\lambda_t + \sum_{C_{lj}\in\mathcal{N}(C_{ti})}(z_{li}\cdot\lambda_l) \leq \rho\cdot\lambda_t. \tag{20}$$

On the other hand, because all teams in $\mathcal{C}^I_t$ have equal profit λ_t, the amount of the reduced profit due to operation (b) is $\lambda_t \times \min\{\sum_{C_{tj}\in\mathcal{C}^I_t\setminus\{C_{ti}\}} z_{tj}, 1-z_{ti}\}$. We next show that this value is at most λ_t.

$$\lambda_t \times \min\{\sum_{C_{tj}\in\mathcal{C}^I_t\setminus\{C_{ti}\}} z_{tj}, 1-z_{ti}\} \leq \lambda_t\times(1-z_{ti}) \leq \lambda_t, \tag{21}$$

where the third inequality is due to $1-z_{ti}\leq 1$.

Eqs. (20) and (21) together imply that

$$(1+\rho)\lambda_t \geq (z_{ti}\cdot\lambda_t + \sum_{C_{lj}\in\mathcal{N}(C_{ti})}(z_{lj}\cdot\lambda_l)) + \lambda_t\cdot\min\{\sum_{C_{tj}\in\mathcal{C}^I_t\setminus\{C_{ti}\}} z_{tj}, 1-z_{ti}\}.$$

It follows that

$$\lambda_t \geq \frac{1}{1+\rho}\left((z_{ti}\cdot\lambda_t + \sum_{C_{lj}\in\mathcal{N}(C_{ti})}(z_{lj}\cdot\lambda_l)) + \lambda_t\cdot\min\{\sum_{C_{tj}\in\mathcal{C}^I_t\setminus\{C_{ti}\}} z_{tj}, 1-z_{ti}\}\right). \tag{22}$$

Note that λ_t represents the profit of C_{ti} and $(z_{ti}\cdot\lambda_t + \sum_{C_{lj}\in\mathcal{N}(C_{ti})}(z_{lj}\cdot\lambda_l)) + \lambda_t\cdot\min\{\sum_{C_{tj}\in\mathcal{C}^I_t\setminus\{C_{ti}\}} z_{tj}, 1-z_{ti}\}$ represents the amount of reduced profit due to the selection of C_{ti}. Hence, (22) indicates that we retain at least $1/(1+\rho)$ fraction of the original profit after selecting C_{ti}. Summing up (22) over all teams from $\mathcal{C}^{DR}$ gives $\sum_{C_{ti}\in\mathcal{C}^{DR}}\lambda_t \geq \sum_{C_{ti}\in\mathcal{C}^H}(x^*_{ti}\cdot\lambda_t)/(\rho+1)$. □

Algorithm Design and Performance Analysis Approx-TG can be naturally adapted to handle this generalization by replacing its LP rounding method with Algorithm 4. In analogy to Lemma 2 and Lemma 3, we present two lemmas to prove the performance bounds of the first and the second candidate solutions, respectively.

LEMMA 6. *Assume x^* is a μ-approximate solution of* ***Primal LP of P.2*** *that is found by the ellipsoid method, Algorithm 2 (whose LP rounding method is replaced with Algorithm 4) achieves an approximation ratio of $\mu/(\Delta+1)$ for* ***P.2****.*

Proof: By Lemma 5, our deterministic rounding technique (Algorithm 4), taking $\mathcal{C}(x^*)$ as input, returns a grouping that achieves a profit of at least $\frac{1}{\Delta+1}\sum_{C_{ti}\in\mathcal{C}(x^*)}(x^*_{ti}\cdot\lambda_t)$, where Δ is the size of the largest possible team in $\mathcal{C}(x^*)$. By the assumption that x^* is a μ-approximate solution of **Primal LP of P.2**, we have $\sum_{C_{ti}\in\mathcal{C}(x^*)}(x^*_{ti}\cdot\lambda_t)\geq\mu\cdot OPT$. It follows that Algorithm 2 achieves a profit of at least

$$\frac{1}{\Delta+1}\sum_{C_{ti}\in\mathcal{C}(x^*)}(x^*_{ti}\cdot\lambda_t)\geq\frac{\mu}{\Delta+1}\cdot OPT$$

□

LEMMA 7. *Assume x^* is a μ-approximate solution of* ***Primal LP of P.2*** *that is found by the ellipsoid method, Algorithm 3 (whose LP rounding method is replaced with Algorithm 4) achieves an approximation ratio of $\mu/2(\sqrt{m}+1)$ for* ***P.2****.*

Proof: To prove this lemma, it suffices to show that $\max\{\sum_{C_{ti}\in\widetilde{\mathcal{C}}}\lambda_t,\lambda_{t_{\max}}\}\geq\frac{1}{\sqrt{m}+1}\cdot\frac{\mu}{2}\cdot OPT$.

We first bound the gap between the profit gained from $\widetilde{\mathcal{C}}$ and $\sum_{C_{ti}\in\mathcal{C}(x^*)^1}(x^*_{ti}\cdot\lambda_t)$. By Lemma 5, we have

$$\sum_{C_{ti}\in\widetilde{\mathcal{C}}}\lambda_t\geq\frac{1}{\rho(\widetilde{\mathcal{C}})+1}\cdot\sum_{C_{ti}\in\mathcal{C}(x^*)^1}(x^*_{ti}\cdot\lambda_t)\geq\frac{1}{\sqrt{m}+1}\cdot\sum_{C_{ti}\in\mathcal{C}(x^*)^1}(x^*_{ti}\cdot\lambda_t)\tag{23}$$

where the second inequality is due to the assumption that $\rho(\widetilde{\mathcal{C}})\leq\sqrt{m}$ holds for all teams from $\mathcal{C}(x^*)^1$.

Adopting the same argument used in the proof of Lemma 3, we can bound the gap between the profit gained from $C_{t_{\max}}$ and $\sum_{C_{ti}\in\mathcal{C}(x^*)^2}(x^*_{ti}\cdot\lambda_t)$ as follows:

$$\lambda_{t_{\max}}\geq\frac{1}{\sqrt{m}}\cdot\sum_{C_{ti}\in\mathcal{C}(x^*)^2}(x^*_{ti}\cdot\lambda_t)\tag{24}$$

By the assumption that x^* is a μ-approximate solution of **Primal LP of P.2**, we have

$$\sum_{C_{ti}\in\mathcal{C}(x^*)}(x^*_{ti}\cdot\lambda_t)\geq\mu\cdot OPT\Rightarrow\sum_{C_{ti}\in\mathcal{C}(x^*)^1}(x^*_{ti}\cdot\lambda_t)+\sum_{C_{ti}\in\mathcal{C}(x^*)^2}(x^*_{ti}\cdot\lambda_t)\geq\mu\cdot OPT \tag{25}$$

Now we are ready to prove this theorem.

$$\max\{\sum_{C_{ti}\in\widetilde{\mathcal{C}}}\lambda_t,\lambda_{t_{\max}}\}\geq\frac{\sum_{C_{ti}\in\widetilde{\mathcal{C}}}\lambda_t+\lambda_{t_{\max}}}{2} \tag{26}$$

$$\geq\frac{\frac{1}{\sqrt{m}+1}\cdot\sum_{C_{ti}\in\mathcal{C}(x^*)^1}(x^*_{ti}\cdot\lambda_t)+\frac{1}{\sqrt{m}}\cdot\sum_{C_{ti}\in\mathcal{C}(x^*)^2}(x^*_{ti}\cdot\lambda_t)}{2} \tag{27}$$

$$\geq\frac{1}{\sqrt{m}+1}\cdot\frac{\sum_{C_{ti}\in\mathcal{C}(x^*)^1}(x^*_{ti}\cdot\lambda_t)+\sum_{C_{ti}\in\mathcal{C}(x^*)^2}(x^*_{ti}\cdot\lambda_t)}{2} \tag{28}$$

$$\geq\frac{1}{\sqrt{m}+1}\cdot\frac{\mu}{2}\cdot OPT \tag{29}$$

where the second inequality is due to (23) and (24), and the last inequality is due to (42). □

Lemma 6 and Lemma 12 together imply the following theorem.

THEOREM 5. *Approx-TG achieves an approximation ratio of* $\max\{\mu/(\Delta+1),\mu/2(\sqrt{m}+1)\}$ *for* ***P.2****.*

6.2. Incorporation of Heterogenous Load Limits

Our basic model assumes that each individual can only participate in *one* task. For the general case when each individual u can participate in up to f_u number of tasks, one naive approach is to simply create f_u copies of u with identical skill set for each u. It turns out we can still apply Approx-TG to this expanded set to achieve an approximation ratio of $\max\{\mu/\Delta,\mu/2\sqrt{m}\}$. However, this is not a polynomial time algorithm if f_u is exponential in the size of input. We next present a polynomial time approximation algorithm based on LP relaxation.

P.3: *Maximize* $\sum_{C_{ti}\in\mathcal{C}}(x_{ti}\cdot\lambda_t)$

subject to:

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

LP Relaxation The primal LP of **P.3** can be formulated as follows.

Primal LP of P.3: *Maximize* $\sum_{C_{ti}\in\mathcal{C}}(x_{ti}\cdot\lambda_t)$

subject to:

$$\begin{cases}\sum_{C_{ti}\in\mathcal{C}:C_{ti}\ni u}x_{ti}\leq f_u,\forall u\in\mathcal{V}\\ 0\leq x_{ti},\forall C_{ti}\in\mathcal{C}\end{cases}$$

In the dual problem, we assign a price $y(u)$ to each node $u \in \mathcal{V}$:

Dual LP of P.3: *Minimize* $\sum_{u\in\mathcal{V}} f_u \cdot y(u)$

subject to:

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

We can still adopt the ellipsoid method for exponential-sized LP with an (approximate) separation oracle to solve **Dual LP of P.3** to obtain a fractional solution x^*.

THEOREM 6. *If there is a polynomial μ-approximation algorithm for* MINCOSTTEAMSELECTION, *then there exists a polynomial μ-approximation algorithm for* ***Primal LP of P.3****.*

LP Rounding Our randomized rounding (Algorithm 5) consists of two stages: a *initial rounding stage* and a *conflict resolution stage*. In the initial rounding stage, we covert x^* to a group of teams that might not be feasible; then in the second stage, we remove some teams to obtain a feasible solution. We next explain each stage in detail. Given a feasible solution x^* of **Primal LP of P.3**, Algorithm 5 takes $\mathcal{C}^I$, a subset of $\mathcal{C}(x^*)$, as input.

1. For each team $C_{ti} \in \mathcal{C}^I$, add C_{ti} to $\mathcal{C}^{RR}$ with probability $\frac{x^*_{ti}}{2\rho}$, where ρ is the size of the largest team in $\mathcal{C}^I$. We say C_{ti} survives in the first stage if C_{ti} has been added to $\mathcal{C}^{RR}$.

2. Note that $\mathcal{C}^{RR}$ might violate the constraint of load limits. This can be resolved as follows: For each team $C_{ti} \in \mathcal{C}^{RR}$, keep C_{ti} in $\mathcal{C}^{RR}$ if and only if for all $u \in C_{ti}$, $\sum_{C_{lj}\in\mathcal{C}^{RR}:C_{lj}\ni u} 1 \leq f_u$. We say C_{ti} survives in the second stage if C_{ti} has been kept in $\mathcal{C}^{RR}$. Return $\mathcal{C}^{RR}$ as the output.

Algorithm 5 Randomized Rounding

Input: $x^*, \mathcal{C}^I \subseteq \mathcal{C}(x^*)$.

1: $\mathcal{C}^{RR} = \emptyset$.
2: **for** each team C_{ti} in $\mathcal{C}^I$ **do**
3: Add C_{ti} to $\mathcal{C}^{RR}$ with probability $\frac{x^*_{ti}}{2\rho}$.
4: **for** each $C_{ti} \in \mathcal{C}^{RR}$ **do**
5: Remove C_{ti} from $\mathcal{C}^{RR}$ if for some $u \in C_{ti}$, $\sum_{C_{lj}\in\mathcal{C}^{RR}:C_{lj}\ni u} 1 > f_u$.
6: Return $\mathcal{C}^{RR}$.

LEMMA 8. *Given any feasible solution x^* of **Primal LP of P.3** and input teams $\mathcal{C}^I \subseteq \mathcal{C}(x^*)$, for each team $C_{ti} \in \mathcal{C}^I$, C_{ti} survives in the first stage with probability $\frac{x^*_{ti}}{2\rho}$.*

The above lemma can be directly derived from our algorithm description. Next we use 0/1 random variable X_{ti} to indicate whether C_{ti} has survived in the first stage, we can immediately have $\mathbf{E}[X_{ti}] = \frac{x^*_{ti}}{2\rho}$.

LEMMA 9. *For any team $C_{ti} \in \mathcal{C}^I$ that is having survived in the first stage, the probability that C_{ti} still survives in the second stage is at least $\frac{1}{2}$.*

Proof: For each $C_{ti} \in \mathcal{C}^I$, let Y_{ti} be a 0/1 random variable representing whether C_{ti} has survived in the second stage. The event that C_{ti} survives in the first phase but removed in the second stage can be represented as: $Y_{ti} = 0$, under the condition that $X_{ti} = 1$. And the probability of this event is $\Pr[Y_{ti} = 0 | X_{ti} = 1]$. We note that this event can only happen if for some $u \in C_{ti}$,

$$\sum_{C_{lj} \in \mathcal{C}^I \setminus C_{ti} : C_{lj} \ni u} X_{lj} \geq f_u$$

By Markov's inequality, the probability of this event can be bounded by

$$\Pr[Y_{ti} = 0 | X_{ti} = 1] \leq \sum_{u \in C_{ti}} \Pr[\sum_{C_{lj} \in \mathcal{C}^I \setminus C_{ti} : C_{lj} \ni u} X_{lj} \geq f_u] \tag{30}$$

$$\leq \sum_{u \in C_{ti}} \frac{\mathbf{E}[\sum_{C_{lj} \in \mathcal{C}^I \setminus C_{ti} : C_{lj} \ni u} X_{lj}]}{f_u} \tag{31}$$

Based on linearity of expectation and $\mathbf{E}[X_{lj}] = \frac{x^*_{lj}}{2\rho}$, for each $u \in C_{ti}$, we have

$$\mathbf{E}[\sum_{C_{lj} \in \mathcal{C}^I \setminus C_{ti} : C_{lj} \ni u} X_{lj}] = \sum_{C_{lj} \in \mathcal{C}^I \setminus C_{ti} : C_{lj} \ni u} \mathbf{E}[X_{lj}] = \sum_{C_{lj} \in \mathcal{C}^I \setminus C_{ti} : C_{lj} \ni u} \frac{x^*_{lj}}{2\rho} \tag{32}$$

By the first constraint of **Primal LP of P.3**, we further have

$$\sum_{C_{lj} \in \mathcal{C}^I \setminus C_{ti} : C_{lj} \ni u} \frac{x^*_{lj}}{2\rho} = \frac{\sum_{C_{lj} \in \mathcal{C}^I \setminus C_{ti} : C_{lj} \ni u} x^*_{lj}}{2\rho} \leq \frac{\sum_{C_{lj} \in \mathcal{C} : C_{lj} \ni u} x^*_{lj}}{2\rho} \leq \frac{f_u}{2\rho} \tag{33}$$

Hence,

$$\Pr[Y_{ti} = 0 | X_{ti} = 1] \leq \sum_{u \in C_{ti}} \frac{\mathbf{E}[\sum_{C_{lj} \in \mathcal{C}^I \setminus C_{ti} : C_{lj} \ni u} X_{lj}]}{f_u} \tag{34}$$

$$= \sum_{u \in C_{ti}} (\sum_{C_{lj} \in \mathcal{C}^I \setminus C_{ti} : C_{lj} \ni u} \frac{x^*_{lj}}{2\rho}) \cdot \frac{1}{f_u} \tag{35}$$

$$\leq \sum_{u \in C_{ti}} \frac{f_u}{2\rho} \cdot \frac{1}{f_u} \tag{36}$$

$$\leq 1/2 \tag{37}$$

where the first inequality is due to (31), the equality is due to (32), the second inequality is due to (33), and the last inequality is due to $|C_{ti}| \leq \rho$.

Therefore, the probability that each team that survives in the first stage still survives in the second stage is at least $1 - \frac{1}{2} = \frac{1}{2}$. □

Lemma 8 and Lemma 9 together imply that for each $C_{ti} \in \mathcal{C}^I$, it survives in both stages with probability $x^*_{ti}/(4\rho)$, hence, the following theorem follows.

LEMMA 10. *Given a feasible solution x^* of **Primal LP of P.3**, a set of input teams $\mathcal{C}^I \subseteq \mathcal{C}(x^*)$, let $\mathcal{C}^{RR}$ denote the group of teams returned from the randomized rounding (Algorithm 5), $\sum_{C_{ti} \in \mathcal{C}^{RR}} \lambda_t \geq \frac{1}{4\rho} \cdot \sum_{C_{ti} \in \mathcal{C}^I} (x^*_{ti} \cdot \lambda_t)$, where $\rho = \max_{C_{ti} \in \mathcal{C}^I} |C_{ti}|$.*

Algorithm Design and Performance Analysis We still use Approx-TG to handle this extension. However, we make two crucial modifications to its original version developed in Section 4.3 as follows. First, we replace its LP rounding method with Algorithm 5. Second, we modify the second candidate solution (Algorithm 3) such that we adopt a different criterion to partition each $\mathcal{C}(x^*)_t$. In particular, for every task $t \in \mathcal{T}$, we partition $\mathcal{C}(x^*)_t$ into two disjoint subsets $\mathcal{C}(x^*)^1_t$ and $\mathcal{C}(x^*)^2_t$ such that: $\forall C \in \mathcal{C}(x^*)^1_t : |C| \leq \sqrt{f_{\max} m}$ and $\forall C \in \mathcal{C}(x^*)^2_t : |C| > \sqrt{f_{\max} m}$, where $f_{\max} = \max_{u \in \mathcal{V}} f_u$ represents the largest number of tasks an individual can participate in. The rest of the algorithm is identical to its original version.

In analogy to Lemma 2 and Lemma 3, we present two lemmas to prove the performance bounds of the first and the second candidate solutions, respectively.

LEMMA 11. *Assume x^* is a μ-approximate solution of **Primal LP of P.3** that is found by the ellipsoid method, Algorithm 2 (whose LP rounding method is replaced with Algorithm 5) achieves an approximation ratio of $\mu/4\Delta$ for **P.3**.*

Proof: By Lemma 5, our randomized rounding technique (Algorithm 5), taking $\mathcal{C}(x^*)$ as input, returns a grouping that achieves a profit of at least $\frac{1}{4\Delta} \sum_{C_{ti} \in \mathcal{C}(x^*)} (x^*_{ti} \cdot \lambda_t)$, where Δ is the size of the largest possible team in $\mathcal{C}(x^*)$. By the assumption that x^* is a μ-approximate solution of **Primal LP of P.3**, we have $\sum_{C_{ti} \in \mathcal{C}(x^*)} (x^*_{ti} \cdot \lambda_t) \geq \mu \cdot OPT$. It follows that Algorithm 2 achieves a profit of at least

$$\frac{1}{4\Delta} \sum_{C_{ti} \in \mathcal{C}(x^*)} (x^*_{ti} \cdot \lambda_t) \geq \frac{\mu}{4\Delta} \cdot OPT$$

□

LEMMA 12. *Assume x^* is a μ-approximate solution of **Primal LP of P.3** that is found by the ellipsoid method, Algorithm 3 (whose LP rounding method is replaced with Algorithm 5), using a modified partitioning criterion, achieves an approximation ratio of $\frac{\mu}{8\sqrt{f_{\max}m}}$ for **P.3**, where $f_{\max} = \max_{u\in\mathcal{V}} f_u$.*

Proof: To prove this lemma, it suffices to show that $\max\{\sum_{C_{ti}\in\widetilde{\mathcal{C}}}\lambda_t, \lambda_{t_{\max}}\} \geq \frac{\mu}{8\sqrt{f_{\max}m}}\cdot OPT$.

We first bound the gap between the profit gained from $\widetilde{\mathcal{C}}$ and $\sum_{C_{ti}\in\mathcal{C}(x^*)^1}(x^*_{ti}\cdot\lambda_t)$. By Lemma 10, we have

$$\sum_{C_{ti}\in\widetilde{\mathcal{C}}}\lambda_t \geq \frac{1}{4\rho(\widetilde{\mathcal{C}})}\cdot\sum_{C_{ti}\in\mathcal{C}(x^*)^1}(x^*_{ti}\cdot\lambda_t) \geq \frac{1}{4\sqrt{f_{\max}m}}\cdot\sum_{C_{ti}\in\mathcal{C}(x^*)^1}(x^*_{ti}\cdot\lambda_t) \tag{38}$$

where the second inequality is due to the assumption that $\rho(\widetilde{\mathcal{C}}) \leq \sqrt{f_{\max}m}$ holds for all teams from $\mathcal{C}(x^*)^1$.

We next bound the gap between the profit gained from $C_{t_{\max}}$ and $\sum_{C_{ti}\in\mathcal{C}(x^*)^2}(x^*_{ti}\cdot\lambda_t)$. In particular, we show that

$$\lambda_{t_{\max}} \geq \frac{1}{\sqrt{m}}\cdot\sum_{C_{ti}\in\mathcal{C}(x^*)^2}(x^*_{ti}\cdot\lambda_t) \tag{39}$$

The following chain proves this inequality:

$$\sum_{C_{ti}\in\mathcal{C}(x^*)^2}(x^*_{ti}\cdot\lambda_t) \leq \sum_{C_{ti}\in\mathcal{C}(x^*)^2}(x^*_{ti}\cdot\lambda_{t_{\max}}) = \lambda_{t_{\max}}\cdot\sum_{C_{ti}\in\mathcal{C}(x^*)^2}x^*_{ti} \leq \lambda_{t_{\max}}\cdot\sqrt{f_{\max}m} \tag{40}$$

The first inequality is due to the assumption that $C_{t_{\max}}$ delivers the highest profit among $\mathcal{C}(x^*)^2$. We next focus on proving the second inequality. Because x^* is a feasible solution of **Primal LP of P.3** and $\mathcal{C}(x^*)^2 \subseteq \mathcal{C}$, we have $\sum_{C_{ti}\in\mathcal{C}(x^*)^2:C_{ti}\ni u}x^*_{ti} \leq f_u, \forall u\in\mathcal{V}$. It follows that

$$\sum_{C_{ti}\in\mathcal{C}(x^*)^2}(x^*_{ti}\cdot|C_{ti}|) \leq \sum_{u\in\mathcal{V}}f_u \leq f_{\max}m \tag{41}$$

Meanwhile, recall that all teams in $\mathcal{C}(x^*)^2$ contain at least $\sqrt{f_{\max}m}$ individuals, we have $\sum_{C_{ti}\in\mathcal{C}(x^*)^2}(x^*_{ti}\cdot|C_{ti}|) \geq \sum_{C_{ti}\in\mathcal{C}(x^*)^2}(x^*_{ti}\cdot\sqrt{f_{\max}m})$. This, together with (41), implies that $\sum_{C_{ti}\in\mathcal{C}(x^*)^2}(x^*_{ti}\cdot\sqrt{f_{\max}m}) \leq f_{\max}m$, thus $\sum_{C_{ti}\in\mathcal{C}(x^*)^2}(x^*_{ti}) \leq \sqrt{f_{\max}m}$. This finishes the proof of the second inequality.

By the assumption that x^* is a μ-approximate solution of **Primal LP of P.3**, we have

$$\sum_{C_{ti}\in\mathcal{C}(x^*)}(x^*_{ti}\cdot\lambda_t)\geq\mu\cdot OPT\Rightarrow\sum_{C_{ti}\in\mathcal{C}(x^*)^1}(x^*_{ti}\cdot\lambda_t)+\sum_{C_{ti}\in\mathcal{C}(x^*)^2}(x^*_{ti}\cdot\lambda_t)\geq\mu\cdot OPT \tag{42}$$

Now we are ready to prove this theorem.

$$\max\{\sum_{C_{ti}\in\widetilde{\mathcal{C}}}\lambda_t,\lambda_{t_{\max}}\}\geq\frac{\sum_{C_{ti}\in\widetilde{\mathcal{C}}}\lambda_t+\lambda_{t_{\max}}}{2} \tag{43}$$

$$\geq\frac{\frac{1}{4\sqrt{f_{\max}m}}\cdot\sum_{C_{ti}\in\mathcal{C}(x^*)^1}(x^*_{ti}\cdot\lambda_t)+\frac{1}{\sqrt{f_{\max}m}}\cdot\sum_{C_{ti}\in\mathcal{C}(x^*)^2}(x^*_{ti}\cdot\lambda_t)}{2} \tag{44}$$

$$\geq\frac{1}{4\sqrt{f_{\max}m}}\cdot\frac{\sum_{C_{ti}\in\mathcal{C}(x^*)^1}(x^*_{ti}\cdot\lambda_t)+\sum_{C_{ti}\in\mathcal{C}(x^*)^2}(x^*_{ti}\cdot\lambda_t)}{2} \tag{45}$$

$$\geq\frac{\mu}{8\sqrt{f_{\max}m}}\cdot OPT \tag{46}$$

where the second inequality is due to (38) and (39), and the last inequality is due to (42). □

Lemma 6 and Lemma 12 together imply the following theorem.

THEOREM 7. *The modified* Approx-TG *achieves an approximation ratio of* $\max\{\mu/(4\Delta),\mu/(8\sqrt{f_{\max}m})\}$ *for* ***P.3****, where* $f_{\max}=\max_{u\in\mathcal{V}}f_u$.

7. Conclusion

In this paper, we study the profit-driven team grouping problem. We assume a collection of tasks $\mathcal{T}$, where each task requires a specific set of skills, and yields a different profit upon completion. Individuals may collaborate with each other in the form of *teams* to accomplish a set of tasks. 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 generalized cover decomposition problems.

References

Anagnostopoulos, Aris, Luca Becchetti, Carlos Castillo, Aristides Gionis, Stefano Leonardi. 2012. Online team formation in social networks. *Proceedings of the 21st international conference on World Wide Web*. ACM, 839–848.

Bagaria, Vivek Kumar, Ashwin Pananjady, Rahul Vaze. 2013. Optimally approximating the coverage lifetime of wireless sensor networks. *arXiv preprint arXiv:1307.5230* .

Chvatal, Vasek. 1979. A greedy heuristic for the set-covering problem. *Mathematics of operations research* **4** 233–235.

Coase, Ronald H. 1937. The nature of the firm. *economica* **4** 386–405.

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

Gajewar, Amita, Atish Das Sarma. 2012. Multi-skill collaborative teams based on densest subgraphs. *SDM*. SIAM, 165–176.

Golshan, Behzad, Theodoros Lappas, Evimaria Terzi. 2014. Profit-maximizing cluster hires. *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 1196–1205.

Grötschel, Martin, László Lovász, Alexander Schrijver. 1981. The ellipsoid method and its consequences in combinatorial optimization. *Combinatorica* **1** 169–197.

Kargar, Mehdi, Aijun An. 2011. Discovering top-k teams of experts with/without a leader in social networks. *Proceedings of the 20th ACM international conference on Information and knowledge management*. ACM, 985–994.

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

Khandekar, Rohit, Guy Kortsarz, Zeev Nutov. 2012. Approximating fault-tolerant group-steiner problems. *Theoretical Computer Science* **416** 55–64.

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

Li, Cheng-Te, Man-Kwan Shan. 2010. Team formation for generalized tasks in expertise social networks. *Social Computing (SocialCom), 2010 IEEE Second International Conference on*. IEEE, 9–16.

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

Pananjady, Ashwin, Vivek Kumar Bagaria, Rahul Vaze. 2014. Maximizing utility among selfish users in social groups. *Communications (NCC), 2014 Twentieth National Conference on*. IEEE, 1–6.

Slavık, Petr. 1997. A tight analysis of the greedy algorithm for set cover. *Journal of Algorithms* **25** 237–254.

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

Sun, Yahui, Xiaokui Xiao, Bin Cui, Saman Halgamuge, Theodoros Lappas, Jun Luo. 2021. Finding group steiner trees in graphs with both vertex and edge weights. *Proceedings of the VLDB Endowment* **14** 1137–1149.