

Optimizing Peer Teaching to Enhance Team Performance

Zheyuan Shi¹ and Fei Fang²

¹ Swarthmore College, Swarthmore PA 19081, USA

zshi1@swarthmore.edu

² Carnegie Mellon University, Pittsburgh PA 15213, USA

feifang@cmu.edu

Abstract. Collaboration among human agents with different expertise and capabilities is becoming increasingly pervasive and important for developing new products, providing patient-centered health care, propelling scientific advance, and solving social issues. When the roles of the agents in such collaborative teamwork are highly interdependent, the performance of the team will rely not only on each team member's individual capabilities but also on their shared understanding and mutual support. Without any understanding in other team members' area of expertise, the team members may not be able to work together efficiently due to the high cost of communication and the individual decisions made by different team members may even lead to undesirable results for the team. To improve collaboration and the overall performance of the team, the team members can teach each other and learn from each other, and such peer-teaching practice has shown to have great benefit in various domains such as interdisciplinary research collaboration and collaborative health care. However, the amount of time and effort the team members can spend on peer-teaching is often limited. In this paper, we focus on finding the best peer teaching plan to optimize the performance of the team, given the limited teaching and learning capacity. We (i) provide a formal model of the Peer Teaching problem; (ii) present hardness results for the problem in the general setting, and the subclasses of problems with additive utility functions and submodular utility functions; (iii) propose a polynomial time exact algorithm for problems with additive utility function, as well as a polynomial time approximation algorithm for problems with submodular utility functions.

Keywords: Peer Teaching, Teamwork, Optimization

1 Introduction

As we welcome a new age of knowledge segmentation, teamwork is not only escalating in its importance but also shifting in its nature. Its traditional focus on distributing and sharing workload is quickly replaced by the focus on sharing the knowledge and expertise needed for the relevant goal. Such transition is evidenced by, for example, researchers from different domains collaborating

on an interdisciplinary research project, or nurse-physician interprofessional collaboration in health care. However, forming a team with diverse skill sets is far from the end of the story. Richter, Paretti, and McNair [16] show that putting students into interdisciplinary teams and even teaching teamwork skills are not sufficient for effective interdisciplinary collaboration. It is often more desirable for agents to learn some of their teammates' knowledge, than having each agent solely responsible for her own expertise. Bridges et al. [4] find that understanding others' professions in the healthcare team is important in interprofessional collaboration and helps team member better understand her own duties. To have shared knowledge and further enhance the team performance, an efficient and effective way is to have the team members learn from each other. In health and social care, such peer-teaching is viewed as part of the interprofessional education [6, 21], which enables effective collaborative practice [2]. However, the amount of time and effort the team members can spend on peer-teaching is often limited, and it is impossible to ask the team members to gain all of their teammates' knowledge. As such, determining an optimal peer teaching plan is crucial in boosting team performance.

The growing attention to teamwork in the society gives rise to the rapid development of research on team collaboration. Team formation, for instance, addresses the problem of selecting the best team member under limited resources [9, 10, 18]. Various models for team coordination have been proposed, especially when team members can hardly communicate with each other [1, 19]. Works have also been done in the performance measures for teams [13], human-agent teams [22], and communication models [5]. However, few works explicitly consider leveraging the team's diversity and enhancing team performance by having team members teach each other.

Although existing literature on team collaboration does not emphasize peer teaching, the process of, possibly informal, learning from teammates does happen in many teamwork scenarios. Team members often help each other to enhance knowledge in certain topics, to build certain skills, to improve certain ability, or to develop certain capabilities.³ However, there lacks a formal model to study and optimize this process. Our first contribution is the formalization of the peer teaching problem. We characterize a group based on its members and relevant expertise. By quantifying the choices and limits of teaching and learning inside the group, we model the peer teaching problem as a constrained optimization problem.

After formalizing the model, it is natural and important to find the best plan for peer teaching, and we focus on this problem in the rest of the paper. We show that the peer teaching problem in its most general form is hard. However, we analyze two key settings with additive and submodular utility functions and propose two algorithms to find the optimal peer teaching plan. In the first case, we present an exact polynomial time algorithm, and in the second case, we present a polynomial time approximation algorithm.

³ In the rest of the paper, we collectively refer to these abstract concepts as skills which can be taught and learned

2 Related Work

Peer teaching relates to yet differentiates itself from several topics in teamwork which have been studied. Liemhetcharat and Veloso [11] introduce teams with learning agents, where agents have access to external training resources rather than learning from their teammates. Jumadinova, Dasgupta, and Soh [7] treat the peer teaching process as part of a decision problem. However, their work does not explicitly consider the teaching and learning constraints, which are essential to the structure of the peer teaching problem. Compared to the study of cross-training, which refers to agents being trained the expertise of their teammates and is shown to improve the team's performance [14], peer teaching emphasizes the notion of agents autonomously learning from teammates and is thus bounded by various capacity constraints. Several pieces of work on team formation consider the diversity of skills and the synergy among team members [10,12,18]. Peer teaching problem differs from these works in treating a team as given and studies the teaching plan to optimize team performance. Other works focus on the scenario where team members from diverse communities can hardly coordinate prior to collaboration and only loosely coordinate during the collaboration [1,19]. The peer teaching problem applies to this setting and provides the learning and teaching dynamics which the above-mentioned works do not consider.

Much work has been done on multiagent MDP to study the coordination among individual agents on a team [3]. One specific line of research is information sharing, which studies how agents decide when and what observations to share in a partially observable multiagent MDP framework [17, 23]. Peer teaching differs from this area of research in the special nature of knowledge and puts less emphasis on the duration of the process.

We also observe the recent attention on cross-domain collaboration. While this is a place where the process of peer teaching arises frequently, works in this area [20, 24] usually focus on partner recommendation, which is in nature different from our problem.

3 Peer Teaching Problem

In a peer teaching problem, we have a set of agents with the same goal but with different areas of expertise. Before they start working as a team, they can help other team members gain expertise through teaching, and such peer teaching can lead to an improvement in the team performance. However, often there is a limit on how much time and effort an agent can spend on teaching and learning. Therefore, we need to find the best feasible peer teaching plan which can lead to the highest improvement in team performance.

We model the peer teaching problem as a constrained optimization problem defined over a group profile.

Definition 1. *The group profile $\mathcal{G} = (A, S, M, f)$ contains the following*

- $A = \{a_1, \dots, a_n\}$ denotes the set of agents.

- $S = \{s_1, \dots, s_m\}$ denotes the set of areas of expertise, where the area of expertise can be a skill or a type of knowledge.
- $M \subseteq A \times S$ denotes the initial agent-expertise mapping of the group. $(a, s) \in M$ means agent a has expertise in s before any peer teaching takes place. We denote by $M_a(a_i) = \{s_j | (a_i, s_j) \in M\}$ the set of areas of expertise that agent i has and $M_s(s_j) = \{a_i | (a_i, s_j) \in M\}$ the set of agents that has expertise in s_j . We denote by $\bar{M} = A \times S \setminus M$ the complement of M .
- $f : 2^{\bar{M}} \rightarrow \mathbb{R}$ is the utility function. A learning profile $M' \subseteq \bar{M}$ is a set of learning events, and a learning event $(a, s) \in M'$ means agent a gains some expertise in s from some other agent during peer teaching. The utility function indicates how much improvement a learning profile can bring to the team performance.

Next, we introduce several definitions towards the definition of the collection of all feasible learning profiles. $T = \{T_1, \dots, T_n\}$ denotes the teaching capabilities of the agents. $T_i \subseteq M_a(a_i)$ and $s_j \in T_i$ means agent a_i is capable of helping other agents to gain expertise in s_j through teaching. Differentiating T_i from $M_a(a_i)$ provides a way to quantify the level of expertise of each agent, as one might expect that the ability to teach others implies a high level of proficiency. A peer teaching plan is defined as a set of triplets of (teacher, expertise, learner), i.e., $\Theta = \{(a_{i_1}, s_j, a_{i_2}) | (a_{i_1}, s_j) \in T_{i_1}, (a_{i_2}, s_j) \in \bar{M}\}$. A peer teaching plan is feasible if it satisfies teaching and learning capacity constraints defined by c_i^t , c^t , and c_i^l . c_i^t represents the maximum number of expertise agent i can teach, c^t represents the maximum number of agents that any agent can teach simultaneously for one expertise and c_i^l represents the maximum number of expertise one can gain through peer teaching. A learning profile M' is feasible if there exists a feasible peer teaching plan $\Theta_{M'}$ which realizes all learning events in M' . Given a group profile \mathcal{G} and the collection of all feasible learning profiles L , the peer teaching problem is to find a learning profile $M^* \in L$ to optimize the utility function.

Figure 1 provides an example of the peer teaching problem. For illustration purpose, it has three agents and three areas of expertise. The right side of the graph is a bipartite graph which represents each agent's teaching capability. In this example, we assume $T_i = M_a(a_i)$, i.e., an agent is capable of teaching anything that he currently has expertise in. The bipartite graph on the left shows all possible learning events.

4 Optimizing Peer Teaching

The definition of the peer teaching problem leaves much freedom for deciding the dynamics of the peer teaching process. As we show below, without further structures in the problem, the peer teaching problem is hard.

Theorem 1. *The peer teaching problem in its general form is NP-hard.*

Proof. We prove the hardness of the peer teaching problem by reducing from the maximum cut problem. For an arbitrary weighted undirected graph $G =$

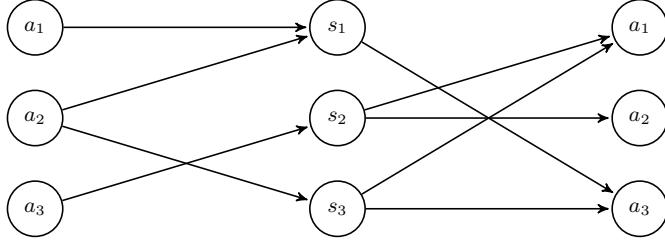


Fig. 1. The knowledge graph and the graph of all possible learning events

(V, E, W) with $|V| = n$ nodes, we consider a corresponding peer teaching problem $\mathcal{G} = (A, S, M, f)$, where $|A| = |S| = n$, $T_i = M_a(a_i) = \{s_j | j \neq i\}$, and $M_s(s_j) = \{a_i | i \neq j\}$. Thus, we have $\bar{M} = \{(a_i, s_i) | i = 1, \dots, n\}$. For any subset $M_I = \{(a_i, s_i) | i \in I\}$ of \bar{M} , we assign the weight of the cut $V_I = \{v_i | i \in I\}$ in graph G to be the utility value $f(M_I)$. The value $f(M_I)$ can be computed in polynomial time. In addition, we ignore teaching and learning capacity constraints by setting $c_i^t, c^t, c_i^l > 1$. Therefore, a subset of nodes V_I in G yields the maximum cut if and only if the corresponding learning profile M_I maximizes the utility function f in the peer teaching problem \mathcal{G} . \square

4.1 Additive Utility Function

The hardness result for the general peer teaching problem calls for more structure in the problem setup. In this subsection, we study a particular type of peer teaching problem characterized by additive utility functions, and present an exact polynomial time algorithm for finding the optimal peer teaching plan.

Consider a group profile $\mathcal{G} = (A, S, M, f)$, where f is such that for all $P, Q \subseteq \bar{M}$, $f(P \cup Q) + f(P \cap Q) = f(P) + f(Q)$. Equivalently, we assign a utility v_{ij} to each learning event $(a_i, s_j) \in \bar{M}$, and define $f(P) = \sum_{(a_i, s_j) \in P} v_{ij}$ for a learning profile $P \subseteq \bar{M}$. This model is natural, for example, when a team is assessed based on the ability of its members individually. As defined in Section 3, each agent a_i has teaching capacity c_i^t and learning capacity c_i^l . For this subsection, we assume all teaching happens in a one-on-one fashion, i.e., $c^t = 1$. We define $l_{ij} = 1$ if agent a_i learns skill s_j , and $t_{ij} = 1$ if agent a_i teaches skill s_j , and zero otherwise. The problem can then be formulated as the following integer linear

program (ILP).

$$\begin{aligned}
\text{minimize} \quad & - \sum_i \sum_j v_{ij} l_{ij} \\
\text{subject to} \quad & \sum_j l_{ij} \leq c_i^l, \quad \forall i = 1, \dots, n \tag{X} \\
& \sum_j t_{ij} \leq c_i^t, \quad \forall i = 1, \dots, n \tag{Y} \\
& \sum_i l_{ij} = \sum_i t_{ij}, \quad \forall j = 1, \dots, m \tag{Z} \\
& l_{ij}, t_{ij} \in \{0, 1\}, \quad \forall i = 1, \dots, n, \forall j = 1, \dots, m
\end{aligned}$$

While solving ILP is hard in general, this problem has the structure of network flow and thus solving the linear program relaxation of the ILP can directly lead to an optimal integer solution [15].

Theorem 2. *The peer teaching problem with additive utility function and $c^t = 1$ can be solved in polynomial time.*

Proof. Recall that a square, integer matrix B is unimodular if $\det(B) = \pm 1$. An integer matrix A is totally unimodular if every square, nonsingular submatrix of A is unimodular. As a sufficient condition, an integer matrix A whose only nonzero entries are ± 1 is totally unimodular if no column of A contains more than two nonzero elements and we may partition the rows of A into I_1 and I_2 such that

- if a column has two entries of the same sign, their rows are in different sets;
- if a column has two entries of different signs, their rows are in the same set.

If A is totally unimodular, and b, u, l are integer vectors, then all the vertices of the polyhedron $P = \{x \mid Ax = b, u \leq x \leq l\}$ are integer points.

Consider the linear program (LP) relaxation of the ILP above. The feasible polyhedron of the relaxed LP can be written as $P = \{x \mid Ax = b, 0 \leq x \leq 1\}$, where we collect all the constraints in X, Y, Z into the equation $Ax = b$ by adding the necessary slack variables. Observe that each column of A has at most two nonzero entries, which are 1 or -1 . Furthermore, we may partition the rows of A into two sets, one containing all constraints in X , the other containing all constraints in Y and Z . Such a partition satisfies the conditions for a totally unimodular matrix as mentioned above. Therefore, it follows that solving the relaxed LP will guarantee us an integer optimal solution. Applying ellipsoid method for the relaxed LP leads to an algorithm that finds the optimal solution in polynomial time. \square

Below we show the running time of the LP compared to two baseline algorithms: the brute force algorithm and the greedy algorithm. The brute force algorithm examines the utility value of all possible learning profiles. The greedy algorithm starts with the learning event with highest utility, and adds the most

beneficial learning event as long as the learning profile remains feasible. The utility values v_{ij} are generated independently from a uniform distribution on integers between 0 and 1000. The teaching and learning capacities c_i^l, c_i^t are generated independently uniformly on integers between 1 and $m = |S|$. We use the *linprog* function in MATLAB R2016a and run on a PC with Intel Core i7-4700MQ processor and 4GB RAM.

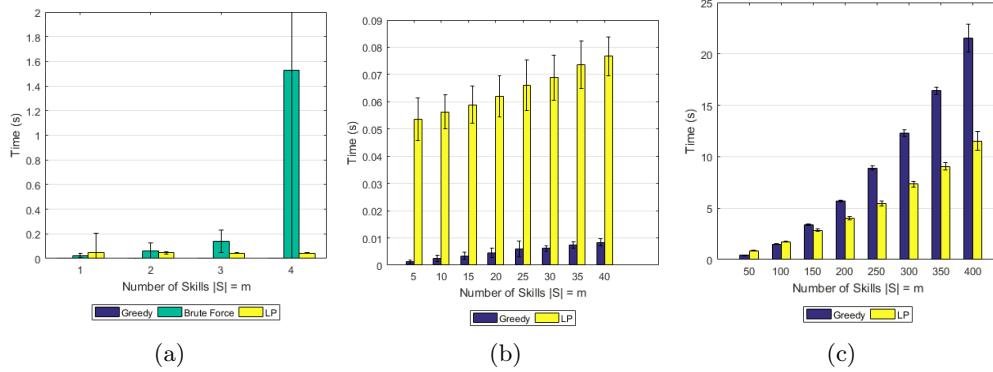


Fig. 2. The running time of the three algorithms. For (a), $|A| = n$ is fixed at 3; for (b), n is fixed at 20; for (c), n is fixed at 200. The standard deviation across five runs is also shown. For (a) and (b), the data are averages over 1000 runs; for (c), the data are averages over 30 runs.

In the experiments we fix the number of agents $|A| = n$ and vary the number of skills $|S| = m$. As shown in Figure 2a, the brute force algorithm quickly blows up, making it infeasible to test its running time beyond $n = 3$, while the running time of the greedy algorithm is negligible compared to others. In Figure 2b, where the problem size is relatively small, the greedy algorithm outperforms the LP in running time. However, as shown in Figure 2c, the LP becomes the better one as the problem grows larger.

We also measure the accuracy of the greedy algorithm by the ratio between its output and the LP optimal utility, as shown in Figure 3. In general, it gives a relatively good approximation, and its accuracy improves as the problem size grows.

4.2 Submodular Utility Function

Under many circumstances, more teaching may not benefit the team as much if the agents are already learning a lot from each other. For instance, given the limited time in a hackathon, students should not bother learning their teammates' programming languages if for each potentially useful language there are already

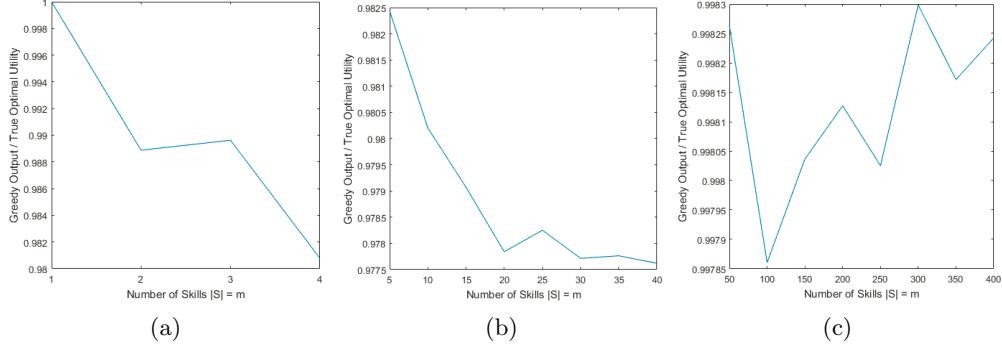


Fig. 3. The accuracy of the greedy algorithm. For (a), $|A| = n$ is fixed at 3; for (b), n is fixed at 20; for (c), n is fixed at 200. For (a) and (b), the data are averages over 1000 runs; for (c), the data are averages over 30 runs.

two or three members who can use it. We may use submodular utility functions to model this diminishing return. To proceed, recall the following definitions.

Definition 2. A function $f : 2^X \rightarrow \mathbb{R}$ is submodular if for any $A, B \subseteq X$, $f(A) + f(B) \geq f(A \cup B) + f(A \cap B)$. X is referred to as the ground set of f .

Definition 3. Let $I \subseteq 2^X$. If a pair $N = (X, I)$ satisfies

- *Downwards closure:* If $P \in I, Q \subseteq P$, then $Q \in I$.
- *Exchange property:* If $P, Q \in I, |P| > |Q|$, then there exists $p \in P \setminus Q$ such that $Q \cup \{p\} \in I$.

then N is a matroid. I is called the collection of independent sets.

In this subsection, we assume that each learning profile $P \subseteq \bar{M}$, feasible or not, is assigned a utility $f(P)$, where f is a submodular function. The solution to the peer teaching problem is a feasible learning profile $P^* \in L$ which maximizes the utility among all feasible learning profiles. To add more structures to the problem, we make the following rules (R1, R2) and assumptions (A1, A2).

- (R1) An agent can teach at most one expertise, but to multiple agents possibly. Equivalently, we set $c_i^t = 1$ for all $a_i \in A$, and set $c^t = n$.
- (R2) An agent can learn at most one expertise. Equivalently, we set $c_i^l = 1$ for all $a_i \in A$.
- (A1) An agent may have multiple expertise but is only able to teach one or two. Equivalently, we assume $|T_i| = 1$ or 2 for all $a_i \in A$.
- (A2) For each expertise, at least two agents can teach it. Equivalently, we assume for all $s_k \in S$, there exist $i \neq j$ such that $s_k \in T_i \cap T_j$.

The two assumptions might not seem very realistic, but we will relax them later. Recall that we refer to $n = |A|$ as the number of agents, and $m = |S|$ as the

number of skills. Consider the knowledge graph in Figure 1. By A1, the number of outgoing edges from agent nodes is less than or equal to $2n$. By A2, the number of incoming edges to skill nodes is greater than or equal to $2m$. Thus, we have $|A| \geq |S|$. These two assumptions allow us to exploit the structure in the feasible learning profiles. More specifically, assuming all the given conditions in this subsection, we have the following theorem.

Theorem 3. *For a given group profile $\mathcal{G} = (A, S, M, f)$, $N = (\bar{M}, L)$ is a matroid.*

Proof. Downwards closure is obvious, we prove the exchange property. Let $P, Q \in L$ be two feasible learning profiles, and $|P| > |Q|$. By R2 and $|P| > |Q|$, there exists an agent a_i who is taught in P but not in Q . Suppose in a peer teaching plan Θ_P corresponding to P , agent a_j teaches a_i the expertise s_k . Let Θ_Q be a peer teaching plan corresponding to Q . We discuss the following possible cases:

Case 1: In Θ_Q , someone is teaching the expertise s_k . According to R1, we can add the learning event (a_i, s_k) to Q , and will maintain feasibility.

Case 2: In Θ_Q , no one is teaching the expertise s_k , but a_j is not teaching anything. We can let a_j teach this expertise s_k to a_i , and add the learning event (a_i, s_k) to Q , and will maintain feasibility.

Case 3: In Θ_Q , no one is teaching the expertise s_k , and a_j is teaching something else. By A2, suppose a_j and a_l know expertise s_k and are both teaching something else, say s_j and s_l . Let $a_{j'}$ be another person who knows s_j , and $a_{l'}$ be another person who knows s_l . Note that $a_{j'}$ and $a_{l'}$ can be the same agent, but they must be different from a_j and a_l by A1. If $a_{j'}$ (or $a_{l'}$) is teaching something else, we repeat the same argument. As this argument propagates, we must be able to find an agent a^* who is not teaching anything, because $n \geq m$ and s_k is not being taught, there must be an agent who is not teaching. Furthermore, by A1, this agent a^* knows some expertise. Then, we can propagate back, and eventually find one of a_j and a_l to teach s_k , without impacting the group's other teaching ability. Once an agent is teaching s_k , we can add the learning event (a_i, s_k) to Q , and will maintain feasibility. \square

With this theorem, the peer teaching problem reduces to maximizing a submodular function subject to a matroid constraint. Unlike minimization, maximizing a submodular function is NP-hard, however. To find the best peer teaching strategy, we may use the algorithm, which we refer to as MAX, proposed by Lee et al. [8]. This is a polynomial time algorithm which achieves a $1/(4 + \epsilon)$ -approximation, assuming a value oracle model, i.e. given a learning profile $P \subseteq \bar{M}$, the algorithm can access the utility value $f(P)$.

The main routine is MAX. LOCAL-SEARCH is a greedy algorithm which improves the current learning profile by adding, deleting, or substituting one learning event at a time. At each step, it checks whether the proposed better learning profile is feasible. Lee et al. [8] do not explicitly provide an algorithm for checking whether a set is independent. However, in our setting the feasibility of a learning profile is not trivial to verify. In FEASIBLE, we consider the bipartite graph representing the current knowledge of the agents (Figure 1). If all learned

Algorithm 1 FEASIBLE

Input: Learning profile $P \subseteq \bar{M}$

if any agent learns more than 1 expertise **then**

Output: false

end if

Get the set of expertise S' that are being learned. Find a maximum matching R on the knowledge graph (Figure 1) restricted to A and S' .

if $|R| = |S'|$ **then**

Output: true

else

Output: false

end if

Algorithm 2 LOCAL-SEARCH

Input: Ground set V , value oracle access to submodular utility f

Set $P = \{e_0\}$, where e_0 is the single learning event with highest utility

while we can do one of the following operations **do**

Delete: If $\exists e \in P$ such that $f(P \setminus \{e\}) \geq (1 + \epsilon/|V|^4)f(P)$, then set $P = P \setminus \{e\}$.

Exchange: If $\exists e \notin P, e' \in P \cup \{\phi\}$ such that $f(P \setminus \{e'\} \cup \{e\}) \geq (1 + \epsilon/|V|^4)f(P)$ and $P \setminus \{e'\} \cup \{e\}$ is feasible, then set $P = P \setminus \{e'\} \cup \{e\}$.

end while

Output: learning profile P

Algorithm 3 MAX

Set $V_1 = \bar{M}$.

Do LOCAL-SEARCH with ground set V_1 , get solution P_1 .

Set $V_2 = \bar{M} \setminus P_1$.

Do LOCAL-SEARCH with ground set V_2 , get solution P_2 .

Output: RETURN the learning profile P_i whose $f(P_i)$ is greater

expertise are being matched in a maximum matching between agents and the expertise being learned in the learning profile, the learning profile is feasible because by R1 each agent can teach at most one expertise. Finally, by doing LOCAL-SEARCH twice, the algorithm MAX achieves the approximation bound of $\frac{1}{4+\epsilon}$.

While MAX is guaranteed to run in polynomial time and achieves a good approximation bound, we wish to relax the Assumptions A1 and A2. First, we consider A2, that for each expertise, there are at least two agents who can teach it.

Theorem 4. *Given A1, we may replace A2 with the assumption that $|A| = n \geq m = |S|$. If we treat n as fixed, we may remove A2, while still having a polynomial time algorithm with the same approximation bound.*

Proof. It is an uninteresting case where no agent knows some skill s_i . Suppose only one agent a_j can teach some expertise s_i , i.e., $s_i \notin T_k$ if $k \neq j$. If a_j can only teach this expertise s_i , then this particular violation of A2 does not fail $N = (\bar{M}, L)$ from being a matroid. Consider the proof of Theorem 3: if a_j is the teaching agent a_j that we picked in P , then we would not even get to Case 3. Otherwise, the pair (a_j, s_i) can be viewed as isolated, and the argument for Case 3 still holds.

If in addition to expertise s_i , agent a_j can also teach expertise s_k , i.e. $T_j = \{s_i, s_k\}$. If s_k also violates A2 such that nobody besides a_j can teach s_k , we may run the algorithm MAX twice and take the better output, where in each run a_j can only teach one of s_i and s_k . If at least one other agent a_l can teach s_k , then we can consider the expertise a_l has, and trim and rearrange their expertise in a way where the only violations of A2 are isolated agent-expertise pairs and the collection of feasible learning profiles L remain unchanged. This is achievable because we instead assume $n \geq m$.

In fact, once we replace A2 with the assumption that $n \geq m$, we may also relax this condition. Suppose we have more expertise than agents, i.e. $m > n$. Let $\{S_i^n\}$ be the collection of all subsets of S where $|S_i^n| = n$. We apply the algorithm MAX to the each of the induced group profiles $\mathcal{G}_i = (A, S_i^n, M_i, f_i)$. Since we assume each agent teaches at most one expertise, this modified algorithm achieves the same approximation bound as if we had only one problem to solve. It is worth noting, however, that naively restricting the problem may violate A1 by having some agents not able to teach any expertise in the subproblem. This can be fixed by assigning all agents who cannot teach any expertise in the subproblem to an imaginary expertise. Meanwhile, we add an imaginary agent who can also teach this imaginary expertise to maintain $m = n$. Then, we extend f_i by assigning the same utility to learning profiles which contain events involving imaginary knowledge or agent as without those events. This preserves the submodularity, and we can continue with the above-proposed procedure. \square

We may also consider relaxing assumption A1. Assuming A2 holds, if agent a_p can teach p expertise where $p \geq 3$, we can initially split the problem into p subproblems, and in each subproblem a_p can only teach one expertise. We may trim and rearrange the knowledge graph so that in each subproblem the matroid

is maintained, and collectively all feasible learning profiles can be reached. However, we may have a combinatorial number of subproblems because to preserve the matroid other agents might require us to divide their outgoing edges in the knowledge graph as well.

5 Conclusion and Future Work

Team collaboration is gaining more attention in the society, and the research community as the segmentation of knowledge continues to grow. It is likely that a team’s performance might improve if members are learning from each other and hence have a better sense of the work. In this paper, we focused on the problem of how teammates should teach and learn from their peers with limited resources to boost group performance. We formalized this process as the peer teaching problem. This problem in its most general case is hard, yet we provided good algorithmic solutions for some two specific setups of the problem, which are still general enough to model many real-world scenarios. We showed that with additive utility functions, we could solve the peer teaching problem with a linear program. In the case of submodular utility, a polynomial time approximation algorithm for maximizing submodular functions can be leveraged to find the optimal peer teaching plan.

There are many future directions to consider. One possible extension of the current peer teaching model is to explicitly quantify to what extent an agent has learned a skill instead of only considering whether or not she has learned the skill. One piece of knowledge often builds on another. Thus, studying planning with precedence graph could better characterize the peer teaching dynamics. Another direction is to consider multi-round collaboration, where at each round agents have different teaching and learning capacity and utility function. Some learning profiles might not be optimal considering the extended duration of collaboration, even if it achieves the best utility at a certain round. The learning aspect of the peer-teaching problem also needs further investigation. In this paper, we assumed agents could access the utility value through a value oracle. It is interesting to study the problem where such access comes with noise, and agents can learn the utility function across time or through available data. Such scenarios appear when, for example, researchers collaborate on interdisciplinary projects. Furthermore, in this paper we only model the member-skill relationship, while one may also consider the familiarity between team members, for example, if the members (or some of the members) have previously worked together on similar projects. In addition, knowing the optimal peer teaching strategy could offer insights into the team formation problem. When selecting group members, candidates’ current expertise matter as well as the potential learning outcome they as a group could achieve. Last but not least, in real world problems, it is also useful to study how peer teaching interacts with learning from other resources.

Acknowledgments. The research presented in this paper was done when the first author was a visiting student at Massachusetts Institute of Technology, and

the second author was a postdoctoral fellow at Harvard University. The research is supported by Harvard Center for Research on Computation and Society fellowship. We thank Barbara Grosz for her comments and helpful discussions.

References

1. Amir, O., Grosz, B.J., Gajos, K.Z.: Mutual influence potential networks: Enabling information sharing in loosely-coupled extended-duration teamwork. In: Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence, IJCAI 2016, New York, NY, USA, 9-15 July 2016. pp. 796–803 (2016)
2. Baker, P.G., et al.: Framework for action on interprofessional education and collaborative practice (2010)
3. Boutilier, C.: Planning, learning and coordination in multiagent decision processes. In: Proceedings of the 6th conference on Theoretical aspects of rationality and knowledge. pp. 195–210. Morgan Kaufmann Publishers Inc. (1996)
4. Bridges, D.R., Davidson, R.A., Odegard, P.S., Maki, I.V., Tomkowiak, J.: Interprofessional collaboration: three best practice models of interprofessional education. *Medical education online* 16 (2011)
5. Fussell, S.R., Kraut, R.E., Lerch, F.J., Scherlis, W.L., McNally, M.M., Cadiz, J.J.: Coordination, overload and team performance: effects of team communication strategies. In: Proceedings of the 1998 ACM conference on Computer supported cooperative work. pp. 275–284. ACM (1998)
6. Gilbert, J.H., Yan, J., Hoffman, S.J.: A who report: framework for action on interprofessional education and collaborative practice. *Journal of Allied Health* 39(3), 196–197 (2010)
7. Jumadinova, J., Dasgupta, P., Soh, L.K.: Strategic capability-learning for improved multiagent collaboration in ad hoc environments. *IEEE Transactions on Systems, Man, and Cybernetics: Systems* 44(8), 1003–1014 (2014)
8. Lee, J., Mirrokni, V.S., Nagarajan, V., Sviridenko, M.: Non-monotone submodular maximization under matroid and knapsack constraints. In: Proceedings of the forty-first annual ACM symposium on Theory of computing. pp. 323–332. ACM (2009)
9. Liemhetcharat, S., Veloso, M.: Modeling mutual capabilities in heterogeneous teams for role assignment. In: Intelligent Robots and Systems (IROS), 2011 IEEE/RSJ International Conference on. pp. 3638–3644. IEEE (2011)
10. Liemhetcharat, S., Veloso, M.: Modeling and learning synergy for team formation with heterogeneous agents. In: Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems-Volume 1. pp. 365–374. International Foundation for Autonomous Agents and Multiagent Systems (2012)
11. Liemhetcharat, S., Veloso, M.: Team formation with learning agents that improve coordination. In: Proceedings of the 2014 international conference on Autonomous agents and multi-agent systems. pp. 1531–1532. International Foundation for Autonomous Agents and Multiagent Systems (2014)
12. Liemhetcharat, S., Veloso, M.: Weighted synergy graphs for effective team formation with heterogeneous ad hoc agents. *Artificial Intelligence* 208, 41–65 (2014)
13. MacBryde, J., Mendibil, K.: Designing performance measurement systems for teams: theory and practice. *Management Decision* 41(8), 722–733 (2003)
14. Marks, M.A., Sabella, M.J., Burke, C.S., Zaccaro, S.J.: The impact of cross-training on team effectiveness. *Journal of Applied Psychology* 87(1), 3 (2002)

15. Papadimitriou, C.H., Steiglitz, K.: Combinatorial optimization: algorithms and complexity. Courier Corporation (1982)
16. Richter, D.M., Paretti, M.C., McNair, L.D.: Teaching interdisciplinary collaboration: learning barriers and classroom strategies. In: proceedings of 2009 ASEE Southeast Section Conference (2009)
17. Roth, M., Simmons, R., Veloso, M.: What to communicate? execution-time decision in multi-agent pomdps. In: Distributed Autonomous Robotic Systems 7, pp. 177–186. Springer (2006)
18. Soriano Marcolino, L., Xin Jiang, A., Tambe, M.: Multi-agent team formation: diversity beats strength? IJCAI (2013)
19. Stone, P., Kaminka, G.A., Kraus, S., Rosenschein, J.S., Agmon, N.: Teaching and leading an ad hoc teammate: Collaboration without pre-coordination. Artificial Intelligence 203, 35–65 (2013)
20. Tang, J., Wu, S., Sun, J., Su, H.: Cross-domain collaboration recommendation. In: Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining. pp. 1285–1293. ACM (2012)
21. Toner, J.A.: Effective interprofessional education–argument, assumption & evidence (2006)
22. Traum, D., Rickel, J., Gratch, J., Marsella, S.: Negotiation over tasks in hybrid human-agent teams for simulation-based training. In: Proceedings of the second international joint conference on Autonomous agents and multiagent systems. pp. 441–448. ACM (2003)
23. Wu, F., Zilberstein, S., Chen, X.: Online planning for multi-agent systems with bounded communication. Artificial Intelligence 175(2), 487–511 (2011)
24. Wu, S., Sun, J., Tang, J.: Patent partner recommendation in enterprise social networks. In: Proceedings of the sixth ACM international conference on Web search and data mining. pp. 43–52. ACM (2013)