Building Collaborative Teams: A Local Navigation-based Approach

Adish Singla[*]
ETH Zurich
adish.singla@inf.ethz.ch

Eric Horvitz[†]
Microsoft Research
horvitz@microsoft.com

Pushmeet Kohli[†]
Microsoft Research
pkohli@microsoft.com

Andreas Krause[†]
ETH Zurich
krausea@ethz.ch

Ryen White[†]
Microsoft Research
ryen.white@microsoft.com

Abstract

Seeking expertise from others and building teams or coalitions is an important part of collaborative problem solving. We discuss the challenges and opportunities that arise in the context of a human computation system in which individual users possess heterogeneous skills and are embedded in an underlying social network that serves as the backbone of information exchange among them. How do structures of acquaintance networks and the limitation of user's knowledge to local connections affects her ability to form collaborations? What is the role played by a user based on her skills and network positioning in terms of her ability to contribute to the whole system?

In this paper, we seek an answer to these questions, starting with a simple generative model to capture the social structures among heterogeneous population seen in reality. Then, we generalize the challenge of team formation and expertise seeking as that of maximizing a submodular function in a decentralized setting where we only have access to local network knowledge. We discuss how the degree of locality of the network knowledge, complexity of utility functions, as well as the properties of the underlying graph affects the hardness of the problem and the ability to get an approximate solution. Our methodology and findings sheds light on how collaborations form among sets of individuals, which we believe is an under-explored problem in human computation and crowdsourcing systems.

1 Introduction

There is much potential in designing human computation (aka. crowdsourcing) systems that can exploit the social ties and ability of people to collaborate. In real-world, such collaborations among people emerge naturally for solving complex tasks, for instance, the recent DARPA Red Balloon challenge [1], the co-authorship networks in academic community, or a user seeking advice from friends in online social network. The traditional online crowdsourcing markets have often focused on micro-tasking through general crowds, dealing with simple tasks such as image annotation, rating of web pages, *etc.* Recent research has explored the collaborative aspects of human computation, though it has been limited to that of designing prototypes or building systems [1][2]. Our work is motivated by such applications, and we study the underlying algorithmic questions that arise in modeling and exploiting such social and collaborative aspects in human computation.

[*] Adish Singla performed this research during an internship at Microsoft Research.

[†] Authors are ordered alphabetically.

[1] `http://archive.darpa.mil/networkchallenge/rules.html`

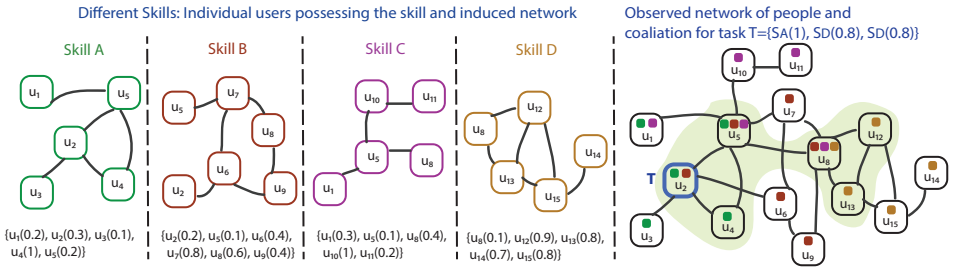

Figure 1: Illustration of the approach on a toy example. The task T originates at user u_2 and requires one user possessing skill S_A with minimum of 1 proficiency and two users of skill S_D with minimum of 0.8 proficiency. User u_5 and u_8 have limited task-solving skills, though act as routing the tasks effectively. u_8 acts as bridge to the group of users with skill S_D, a closed-knit group of users. u_4 and u_{10} are experts for skills S_A and S_C respectively, though have no routing capability.

Network of people and local information. In this work, we are specifically focusing on systems where an individual is seeking expertise from others or need to form a team or coalition of experts, though has only local visibility of the network around her. This local visibility and limited network knowledge leads to challenging algorithmic questions of how should individuals decide whom to query or add to team. While such problems have been studied from point-of-view of central agent with full visibility of the network, the scale of the networks in real-world applications can only offer limited visibility. For instance, such challenges also emerge for information gathering in peer-to-peer networks, or seeking expertise from friends in online social networks. Another key aspect that arise from this decentralized team formation with local visibility is the distinctive notion of *task-routing* and *task-solving* skills. While an individual may not possess specific skills required for the team to be successful, she may still play an important role in team formation by "bridging the gap" and connecting to other group of experts.

1.1 Overview of our approach

We present a toy example in Figure 1 to demonstrate some of the key points of our approach. We consider a population of users, each possessing certain features, that could be useful skills for a task or attributes such as demographics. In this example, there are total of 4 distinctive skills, and 14 total users. For instance, skill A is possessed by users $\{u_1, u_2, u_3, u_4, u_5\}$. Furthermore, instead of binary skills, the skills could be represented by the level of expertise that a user has for a skill. For skill A, the u_4 has expertise level of 1, whereas user u_3 has only an expertise level of 0.1 for this skill. Each of these features could induce its own network, capturing the fact that social ties in real world emerge from number of different aspects (*e.g.*, co-author, or friends, or colleagues sharing same office, *etc.*). In Figure 1, user u_2 has a task T to solve. Task T requires one user possessing skill A with expertise level of at least 1, and two users possessing skill D with expertise level of at least 0.8. The example illustrates an interesting aspect of *task-routing* skills, as possessed by user u_8 for this particular task T when originating at u_2. While user u_8 doesn't provide the required expertise for the task, it still plays an important role in formation of successful team by connecting the users of skill D needed for task (*i.e.*, u_{12} and u_{13}).

We formalize the problem as that of function maximization in a decentralized manner with local knowledge of the network. We would like to point out that, while the team formation is decentralized as per limited network visibility, however, in our model, the team members can all coordinate centrally with the task originator in deciding which user to add next to the team. The key challenges and new algorithmic questions that arise here are because of the limited and local visibility of the network. We discuss the algorithmic aspects of this problem, inherent hardness and then design a general-purpose procedure ϵ-LOKALGREEDY. We analyze and provide theoretical bounds on the performance of our proposed procedure by varying the degree of locality of the network knowledge, complexity of utility functions, as well as the properties of the underlying graph.

2 Problem Statement

We now formally introduce the model.

Users and features. We consider a population of $|U|$ users or people denoted by set $\mathcal{U} = \{u_1, u_2, \ldots, u_{|U|}\}$. Users are associated with features denoting skills (expertise required for the

tasks) and attributes (such as geo-location). Let $|S|$ be the total number of unique skills denoted by set $\mathcal{S}$, $|A|$ be the number of attributes denoted by set $\mathcal{A}$, and $|X|(= |S| + |A|)$ be the total features given by set $\mathcal{X} = \mathcal{S} \cup \mathcal{A}$.

We consider the skills being associated with bounded level of expertise with support $(0, 1]$, here 1 denoting the full expertise of that skill. Note that, having an expertise level 0 simply means that a user doesn't posses a skill. This expertise level can alternatively be interpreted as success probability of being able to "deliver" the skill. The range of attributes vary, and could be categorical (*e.g.*, geo location), or a number (*e.g.*, age). For a user $u \in \mathcal{U}$, we denote the value of her feature $x \in \mathcal{X}$ as x_u (and corresponding values for the skills or attributes as s_u and a_u, respectively).

The features $\mathcal{X}$ associated with the population are drawn from a joint distribution over the space of $\mathcal{X}$, denoted by $\mathcal{D}_{\mathcal{X}}$. We shall not make any assumptions on the distribution of these features among population, and will discuss this further during the performance analysis of our procedures. Specifically, the performance analysis and the bounds of our procedure would depend on the specific distributions we consider.

Network graph. There is an underlying network over these users $\mathcal{U}$, denoted by $G(\mathcal{U}, \mathcal{E})$ and we consider a simple generative model for this network. We consider each feature $x \in \mathcal{X}$ as a "social dimension" that can induce a social graph among users given by G_x. For instance, x associated with geo-location or job position in an organizational hierarchy attributes could denote social ties arising from proximity. Similarly, feature x associated with skill (lets say "machine learning") could lead to social ties arising from collaborations and co-authorship. The final graph that we observe is obtained by an overlay of these networks, given by $G = \cup_{x \in \mathcal{X}} G_x$. This essentially captures the fact that two users u_i and u_j will have an edge in G, if they have tie arising from at least one of the feature. This simple model, allows us to capture other factors as trust, and instead of union, one could think of modeling an intersection of the networks with a trust graph. We let each such graph generated by an independent process, and could simply be a random graph, or through preferential attachment process with scale free properties. We will analyze the performance of the algorithms based on the specific assumptions we make.

Tasks and Teams. A task that is posted by some user during the execution of the system is denoted by T. Let us denote the user from where the task originates as u^o_T (or simply u^o when task T is clear from context). Let us begin by considering a simple scenario. Let the skills $s \in \mathcal{S}$ be binary-valued and and tasks are simply represented by a set of required skills $S_T \subseteq \mathcal{S}$. The goal in this case is to form a team of set of users $\mathcal{C}_T$ (or simply $\mathcal{C}$) of size $|C|$ to cover all the required skills, i.e., $\cup_{u \in \mathcal{C}} S_u \supseteq S_T$.

However, this simple representation has limits in terms of the applications. It is often desirable to consider probabilistic notion of the user skills, for instance, in information gathering. Even more complex applications such as viral marketing may also require a more generic framework. Keeping this in account, we model the tasks as generic utility function, defined next.

Modeling complex tasks through utility functions. We model each task T through a set function over the users (or nodes of the graph), given as $f_T : 2^{\mathcal{U}} \to \mathbb{R}$. Given a team of users $\mathcal{C}$ created for task T, the utility achieved from this team is then given by $f_T(\mathcal{C})$.

We require the set function f to be *non-negative*, *monotone* (*i.e.*, whenever $V \subseteq V' \subseteq \mathcal{U}$, it holds that $f(V) \leq f(V')$) and *submodular*. Submodularity is an intuitive notion of diminishing returns, stating that, for any sets $V \subseteq V' \subseteq \mathcal{U}$, and any given user $v \notin V'$, it holds that $f(V \cup \{v\}) - f(V) \geq f(V' \cup \{v\}) - f(V')$. These conditions are general, and are satisfied by many realistic, as well as complex utility functions [3]. In fact, the above simple example of binary skills and task being represented as a set of required skills, is equivalent to set-cover utility which is submodular.

Local visibility. We are interested in designing procedures where the graph is revealed only incrementally as the team is built. Let us denote $l_{\geq 1}$ hop neighborhood of a user u as $\mathcal{N}(u, l) \subseteq \mathcal{U}$, to be the set of all nodes that are connected to user u either directly or with at most $l - 1$ intermediate nodes. For simplicity, we shall assume that u is also included in $\mathcal{N}(u, l)$. For a set of nodes V, we define its l hop neighborhood as $\mathcal{N}(V) = \cup_{v \in V} \mathcal{N}(v)$. For $l = 1$, this neighborhood would correspond to the directly connected users. We define the local visibility in terms of two parameters l_{deg}, and l_{val}. Consider any given user u:

- l_{deg}: Consider any $v \in \mathcal{N}(u, l_{deg})$. Then, u has visibility of immediate connections of the user v. Essentially, this means that user u has sufficient information to be able to find how much additional "visibility" of the network v can provide when added to the current team.
- l_{val}: Consider any $v \in \mathcal{N}(u, l_{val})$. Then, u has visibility of the features of v given by $x_v \; \forall \, x \in \mathcal{X}$. Essentially, this means that user u has sufficient information to be able to compute the expected marginal gain of utility of adding v to the current team.

Note that, while user u has visibility of these values, we consider setting that team still need to grow in a connected manner, *i.e.*, user u cannot directly add v to team if they are not directly connected. This makes the analysis of the procedures easier, in terms of some of the key properties of the underlying graph. Also, we shall specifically focus on the realistic setting of $l_{deg} = 1$ and $l_{val} = 1$, where this technicality doesn't matter.

Optimization Problem. Consider a task T originating at a any user u_T^o. Let f_T encapsulate the utility function associated with this task. The goal is to find a connected team $\mathcal{C}_T$, starting from user u_T^o, that achieves a quota value Q of the utility function f_T within a tolerance level of β. Our goal is to design a procedure M that has only l_{deg} and l_{val} visibility, and solves the above problem with minimal cost (or size) of the team, under the additional constraints that $\mathcal{C}_T^*$ is connected and contains u_o^T. Formally, we can state the minimization problem as follows:

$$\mathcal{C}_T^* = \arg\min_{C_T \subseteq \mathcal{U}} |C_T| \tag{1}$$

$$\text{subject to } f_T(C_T) \geq (1-\beta) \cdot Q \tag{2}$$

$$C_T \texttt{ is connected } \text{ and } u_T^o \in C_T \tag{3}$$

$$\texttt{Procedure M} \text{ has } l_{deg}, l_{val} \text{ visibility} \tag{4}$$

We shall compare the performance of this procedure against an optimal solution OPT, without computational constraints, that achieves value of Q and doesn't have constraints given by Steps 3,4, *i.e.*, it has global visibility (both l_{deg} and l_{val} are more than the diameter of the graph G, and $\mathcal{C}_T^*$ doesn't need to be connected or pass through u_o^T). We also consider an alternate optimal solution $\overline{\text{OPT}}$, without computational constraints, though obeying all the constraints in Steps 3,4.

3 Our Procedures

3.1 Methodolody

We now give a high level overview of the methodology upon which our main procedure is built.

Exploration of the world. One of the natural choice of adding new user to the team is to add one who can provide maximal visibility of the unseen network. For a given team C, adding a new user v would provide an additional visibility given by $\big(\widetilde{\mathcal{N}}(C \cup \{v\}, 1) - \widetilde{\mathcal{N}}(C)\big)$. This user v may not provide immediate value or required expertise for the task, though the additional visibility would be useful as it gives us more knowledge of the network, potentially helping to discover and connect with the required experts. We call this choice as "exploration".

Exploitation of value. The other natural choice of adding new user to the team is to add the one who provides immediate value to the team in terms of required expertise. For a given team C, adding a new user v would provide a marginal gain in value as $\big(f_T(C \cup \{v\}) - f_T(C)\big)$. This can be thought as "exploitation" step, as it is a greedy choice, given the current knowledge of the network.

We now describe our main procedure ϵ-LOKALGREEDY.

3.2 Explore-Exploit based ϵ-LOKALGREEDY

Our main procedure ϵ-LOKALGREEDY is based on simple idea of interleaving these two choices and is illustrated in Procedure 1. It turns out, that this simple approach allows us to derive tight theoretical bounds on our procedure, which also runs quite efficient for various problem instances. We now specify the main steps based on which the procedure is built.

Procedure 1: Procedure ϵ-LOKALGREEDY

1 **Input**:
- □ Social graph: $G(\mathcal{U}, \mathcal{E})$;
- □ Task: T; user posting the task: u^o; Utility function: f;
- □ Exploration parameter: ϵ;
- □ Local visibility: $\{l_{deg}, l_{val}\}$;
- □ Solution parameters: (β, Q);

2 **Output**: team C;

3 **Initialize**:
- □ $C = \{u^o\}$; $i = 1$; $\epsilon^i = \epsilon$;

4 **while** $f(C) < (1 - \beta) \cdot Q$ **do**

5 **if** $\mathcal{N}(C, 1) = \mathcal{U}$ **then**

6 $\epsilon^i = 0$; ▷ Update ϵ^i

end

7 With prob. ϵ^i, $a^i \leftarrow$ EXPLORE; else, $a^i \leftarrow$ EXPLOIT;

8 **if** $a^i =$ EXPLORE **then**

9 $\Pi^* = \arg\max_{\Pi_l^C : l \in [1 \ldots l_{deg}]} \frac{\widetilde{\mathcal{N}}(C \cup \Pi_l^C, 1) - \widetilde{\mathcal{N}}(C, 1)}{l}$;

10 $\mathcal{C}_T = \mathcal{C}_T \cup \Pi^*$;

11 **if** $l_{deg} = 1$ **then**

12 Randomly pick $v^* \in \widetilde{\mathcal{N}}(C \cup \Pi^*, 1) - \widetilde{\mathcal{N}}(C, 1)$;

13 $\mathcal{C}_T = \mathcal{C}_T \cup \{v^*\}$;

end

else

14 $\Pi^* = \arg\max_{\Pi_l^C : l \in [1 \ldots l_{val}]} \frac{f(C \cup \Pi_l^C) - f(C)}{l}$;

15 $\mathcal{C}_T = \mathcal{C}_T \cup \Pi^*$;

end

16 $i = i + 1; \epsilon^i = \epsilon^{i-1}$;

end

17 **Output**: $\mathcal{C}_T$

Interleaving network exploration with expertise exploitation. The procedure interleaves exploration and exploitation with ϵ probability, as illustrated in Step 7 of Procedure 1. The ϵ is constant for the procedure and is provided as input. If further prior information about the network properties or the optimal solution are available, this parameter can be tuned (for instance, whether to do more exploration or more exploitation based on such properties). Furthermore, this can be adjusted or learnt during execution of the procedure (for instance, intuitively, the extent of exploration can be reduced over execution of the procedure) . In Procedure 1, we just use a simple constant ϵ, and when the dominating set for the network is already built, we set $\epsilon = 0$ (see Step 6).

Adding multiple users. As the procedure has visibility of l_{deg} and l_{val}, multiple users can be added in one round. Our idea is based on the intuition used in [4] on how to effectively add upto two ($l_{deg} = 2$) users in a way to be able to built efficient connected dominating sets. Negative results from [4] show that this "look ahead" is indeed necessarily to do, as $l_{deg} = 1$ greedy approach may need up to $|U|$ users to build a connected dominating set. We generalize this idea of one-step look ahead (for $l_{deg} = 2$) to that of following a gradient up to l-hops if average gain in that direction is high.

To formalize this, we introduce the concept of chain of length l. Consider a set of users V who are currently in team. The "exposed" neighborhood of V is the 1-hop neighborhood given by $\widetilde{\mathcal{N}}(V, 1) = \mathcal{N}(V, 1) \setminus V$. We define a l-chain from V, denoted by Π_l^V as an ordered set denoted defined recursively as follows:

$$\Pi_1^V \in \widetilde{\mathcal{N}}(V, 1);$$
$$\Pi_2^V \in \widetilde{\mathcal{N}}(V \cup \Pi_1^V, 1) \setminus \widetilde{\mathcal{N}}(V, 1);$$
$$\Pi_{i \geq 3}^V \in \widetilde{\mathcal{N}}(V \cup \Pi_{i-1}^V, 1) \setminus \widetilde{\mathcal{N}}(V \cup \Pi_{i-2}^V, 1);$$

Step 9,14 illustrates of how all the possible chains of length 1 to l are enumerated to find the set of users to add at each round.

The case of $l_{deg} = 1$. As shown in the seminal work of [4], a simple deterministic procedure for $l_{deg} = 1$ that add users greedily based on maximizing the "exposure" may have worst case cost of $|U|$. This worst case is resovled for $l_{deg} = 2$ by using the idea of look ahead, that we have generalized above by the notion of l-hop chains above. Recently, this barrier of $l_{deg} = 1$ has been resolved by [5] using a simple randomization technique. Specifically, after a user v is added, another random user from her neighborhood $\mathcal{N}(v, 1)$ that is newly exposed is added as well. This is illustrated in the Step 12 of Procedure 1, which is execution when $l_{deg} = 1$.

4 Performance Analysis

We now analyze the performance of the procedure ϵ-LOKALGREEDY.

4.1 General Analysis

We first analyze the results for general settings, where we do not make any assumptions on the graph or skill distribution.

Characteristic properties of the graphs. Let us first introduce some of the characteristic properties of the graph G that we shall use to express the performance.

- **Maximum vertex degree Δ_G.** This denotes the degree of graph G, defined as the maximum degree of any vertex in the graph given by $\max_{u \in \mathcal{U}} \Delta_G(u)$, where $\Delta_G(u)$ is the degree of vertex u in the graph.
- **Smallest size of *connected* dominating set γ_c.** This denotes the size of smallest dominating set of the graph, that is also connected. The dominating set of a graph is denoted by set of nodes $D \subseteq \mathcal{U}$, such that if we consider any vertex $v \in \mathcal{U}$, either $v \in D$, or v is adjacent to some vertex v' where $v' \in D$. Note that finding smallest connected dominating set D is NP-Hard [4].

Theorem 1. *For $l_{deg} = 2$, $l_{val} = 1$, the procedure ϵ-LOKALGREEDY terminates with team C that satisfies the constraints of Equation 1 with following upper bound on the size of C in expectation (over the coin flips):*

$$\mathbb{E}[|C|] \leq \Big(\frac{1}{\epsilon} \cdot \big(2 + 2\ln(\Delta_G)\big) \cdot \gamma_G^c\Big) + \Big(\frac{1}{1-\epsilon} \cdot |\textsc{Opt}| \cdot \ln(\frac{1}{\beta})\Big)$$

Theorem 2. *For $l_{deg} = 1$, $l_{val} = 1$, the algorithm ϵ-LOKALGREEDY terminates with team C that satisfies the Equation 1 with following upper bound on the size of C in expectation (over the coin flips) that holds with probability at least $1 - e^{-\gamma_G^c}$:*

$$\mathbb{E}[|C|] \leq \Big(\frac{1}{\epsilon} \cdot \big(4 + 2 \cdot \ln(\Delta_G)\big) \cdot \gamma_G^c\Big) + \Big(\frac{1}{1-\epsilon} \cdot |\textsc{Opt}| \cdot \ln(\frac{1}{\beta})\Big)$$

For the general settings, we can state the following lower bound, showing that the dependency on γ_G^c is indeed unavoidable.

Theorem 3. *For any bounded l_{deg} and l_{val}, there exists an underlying distrbution of skills and graph structure for which any procedure will have expected size of team at size least:*

$$\mathbb{E}[|C|] \geq \max\Big(\gamma_G^c,\ |\textsc{Opt}|\Big)$$

4.2 Preferential Attachment with independently distribtued skills

We now analyze the results when the graphs are scale-free, along with specific properties over the skill distribution and utility function associated with the task.

Skill distribution. The skills and features are independently distributed. For each feature $x \in \mathcal{X}$, let the population possessing that feature is given by size $|U_x|$ and associated probability $\theta_x = \frac{|U_x|}{|U|}$. Under independence assumption, we can state that if we take any user u and consider its M outgoing links, then the probability that at least one of them poses feature x is at least $(1 - (1 - \theta_x)^M)$.

Characteristic properties of the graphs. Each network G_x is formed by preferential attachment process [6], where nodes $u \in \mathcal{U}_x$ arrive over time, and each node on arrival forms m_x links to existing nodes with probability proportional to the degree. This network leads to power law distribution, often seen in the collaborative networks such as co-authorship graphs, online social network or the Web graph. We assume that the skill level is proportional to the outgoing degree, scaled to have bounded support of $(0, 1]$.

Characteristics properties of function. We now characterize the utility function properties. We consider a separable function given by:

$$f_T(C) = \sum_{s \in S} w_T^s \cdot f_T^s(C)$$

where w_T^s denotes the weight of function f_T^s. Here, the function f_T^s depends only the skill s of the users. There are set of "irrelevant" skills for which task T carries no utility, *i.e* $w_T^s = 0$. For other "useful" skills for which $w_T^s > 0$, function f_T^s carries positive, non-zero utility for every user possessing that skill.

Also, the marginal utilities of individual users are monotonic w.r.t the their skill level. For instance, consider adding a user u_1 or u_2 to team C, where $s_{u_1} \leq s_{u_2}$. Then, $\big(f_T^s(C \cup \{u_1\}) - f_T^s(C)\big) \leq \big(f_T^s(C \cup \{u_2\}) - f_T^s(C)\big)$.

Characteristics properties of OPT. For a given task, consider the function associated with any specific useful skill s for which $w_T^s > 0$. Then, maximum value of function $f_T^s(C) = 1$ will be achieved after including some constant number κ of highly skilled users for skill s. Highly skilled users are the ones for which $s_u \geq O\big(\frac{1}{\ln(|U_s|)}\big)$. This is equivalent to $\Delta_{G_s}(u) \geq O\big(\frac{1}{\ln(|U_s|)}\big)\Delta_{G_s}$. This is motivated by real-world settings where the goal is essentially to find few experts for all the required skills. As per the power law distribution from the preferential attachment process, the number of such highly skilled users are relatively few.

Theorem 4. *Consider $l_{deg} = 1$, $l_{val} = 1$ and a task T originating from user u_T^o possessing skill s^o. Let the goal is to find a small number of κ highly skilled value users of certain skills, where the highly skilled users are the ones for which $s_u \geq O\big(\frac{1}{\ln(|U_s|)}\big)$ for any skill s. With probability of at least $1 - o(1)$, size of C is bounded as:*

$$\mathbb{E}[|C|] \leq \Big(\frac{1}{\epsilon} \cdot O\big(\ln^4(|U_{s^0}|)\big)\Big) + \Big(\frac{1}{1-\epsilon} \cdot O\big(\sum_{s \in S} ln^4(|U_s|) + \kappa\big)\Big) + |OPT|$$

The main key ideas behind the above theorem are as follows.

Reaching high degree nodes of skill s^0. The first term in the summation $\Big(\frac{1}{\epsilon} \cdot O\big(\ln^4(|U_{s^0}|)\big)\Big)$ is attributed from the process of reaching high degree users among the users of skill s^o, let us denote one such high degree user as r_{s^0}. The exploration step of the procedure will add users to the team towards reaching r_{s^0}.

Based on the results of [5], consider starting from any node and navigating graph G_x, if we follow the maximum degree node in the $l_{deg} = 1$ local visibility, we reach a node r_x within $O\big(\ln^4(|U_x|)\big)$ steps such that $\Delta_{G_x}(r_x) \geq \frac{1}{\ln^2(|U_x|)} \cdot \Delta_{G_x}$, this holds with probability at least $1 - o(1)$ (where the o notation is w.r.t to size of the graph). Furthermore, with probability at least $1 - o(1)$, the maximum degree of of any node in G_x, denoted Δ_{G_x} is bounded by $\Delta_{G_x} \leq m_x \cdot \sqrt{|U_x|} \cdot \ln(|U_x|)$, based on results of [7]. Hence, the degree $\Delta_G(r_s^0) \geq \frac{m_s^0 \cdot \sqrt{|U_s^0|}}{\ln(|U_s^0|}$.

Jumping to users with useful skills. Then, the exploitation step would ensure that we can make a high probabilistic jump from r_s^0 to the users with useful skills. Given the independence assumption of the skills distribution, the procedure can jump to users with useful skills, with probability at least: $1 - \sum_{s \in S}(1 - \theta_s)^{\Delta_G(r_s^0)}$

Reaching high degree nodes of useful skills. Both the exploration and exploitation step would help with this process. However, the exploration step may still continue adding higher degree nodes in the graph G_{s^0} or other skills that do not provide any utility, hence the second term in bound of Theorem 4 has the factor of $\frac{1}{1-\epsilon}$.

Based on the results of [5], consider starting from any node and navigating graph G_x, if we follow the maximum degree node in the $l_{deg} = 1$ local visibility, we reach a small constant number of high degree κ nodes, denoted by R_x^κ, within $O\big(\ln^4(|U_x|)+\kappa\big)$ steps such that $\Delta_{G_x}(r_x) \geq \frac{1}{\ln^3(|U_x|)} \cdot \Delta_{G_x}$, this holds with probability at least $1 - o(1)$. By definition, finding such κ highly-skilled users are enough for any given useful skill.

Next, we state the lower bound for any procedure under this model.

Theorem 5. *For $l_{deg} = 1$, $l_{val} = 1$, under the specific model considered in Theorem 4, there exists a problem instance for which any procedure will have expected size of team at least:*

$$\mathbb{E}[|C|] \geq O\Big(\frac{\ln(|U|)}{\ln\ln(|U|)}\Big)$$

This lower bound follows from the expected diameter for the graph obtained from the preferential attachment process [8].

References

[1] Haoqi Zhang, Edith Law, Rob Miller, Krzysztof Gajos, David Parkes, and Eric Horvitz. Human computation tasks with global constraints. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 217–226. ACM, 2012.

[2] Matthew Richardson and Ryen W. White. Supporting synchronous social q&a throughout the question lifecycle. In *Proceedings of the 20th International Conference on World Wide Web, WWW*, pages 755–764, 2013.

[3] Andreas Krause and Daniel Golovin. Submodular function maximization. *Tractability: Practical Approaches to Hard Problems*, 3:19, 2012.

[4] Sudipto Guha and Samir Khuller. Approximation algorithms for connected dominating sets. *Algorithmica*, 20(4):374–387, 1998.

[5] Christian Borgs, Michael Brautbar, Jennifer Chayes, Sanjeev Khanna, and Brendan Lucier. The power of local information in social networks. In *Internet and Network Economics*, pages 406–419. Springer, 2012.

[6] Albert-László Barabási and Réka Albert. Emergence of scaling in random networks. *science*, 286(5439):509–512, 1999.

[7] B. Bollobs. Mathematical results on scale-free random graphs. In *Handbook of Graphs and Networks*, pages 1–37. Wiley, 2003.

[8] Béla Bollobás and Oliver Riordan. The diameter of a scale-free random graph. *Combinatorica*, 24(1):5–34, 2004.