Optimizing Peer Learning in Online Groups with Affinities

Mohammadreza Esfandiari, Wei Dong, Sihem Amer-Yahia, Senjuti Basu Roy

Optimizing Peer Learning in Online Groups with Affinities

Mohammadreza Esfandiari [+], Dong Wei[+], Sihem Amer-Yahia[*], Senjuti Basu Roy[+]
{me76,dw277,senjutib}@njit.edu,Sihem.Amer-Yahia@imag.fr
New Jersey Institute of Technology[+],Univ. Grenoble Alpes, CNRS, France[*]
Authors' Copy

ABSTRACT

We investigate online group formation where members seek to increase their *learning potential* via collaboration. We capture two common learning models: LpA where each member learns from all higher skilled ones, and LpD where the least skilled member learns from the most skilled one. We formulate the problem of forming groups with the purpose of optimizing peer learning under different affinity structures: AffD where group affinity is the smallest between all members, and AffC where group affinity is the smallest between a designated member (e.g., the least skilled or the most skilled) and all others. This gives rise to multiple variants of a multi-objective optimization problem. We propose principled modeling of these problems and investigate theoretical and algorithmic challenges. We first present hardness results, and then develop computationally efficient algorithms with constant approximation factors. Our real-data experiments demonstrate with statistical significance that forming groups considering affinity improves learning. Our extensive synthetic experiments demonstrate the qualitative and scalability aspects of our solutions.

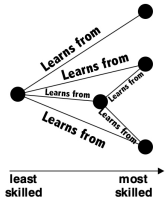
ACM Reference Format:
Mohammadreza Esfandiari [+], Dong Wei[+], Sihem Amer-Yahia[*], Senjuti Basu Roy[+]. 2019. Optimizing Peer Learning in Online Groups with Affinities. In *The 25th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD'19), June 22–24, 2019, Anchorage, AK, USA.* ACM, New York, NY, USA, 11 pages. https://doi.org/10.1145/3292500.3330945

1 INTRODUCTION

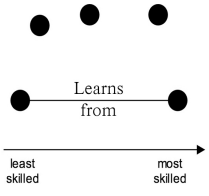
The emergence of platforms that support online networked technologies has changed the way we communicate, collaborate, and learn things together. Existing works have focused on how to identify and rank groups and communities [8], how to efficiently form a set of groups to optimize different group recommendation semantics [7], or form groups for task assignment [4–6, 14, 18]. The effect of online collaboration however goes beyond, as it enables powerful and versatile strategies to improve knowledge of individuals and promote learning. For example, online critiquing communities,[1] social Q&A sites,[2] and crowdsourcing platforms[3] investigate how collaboration can promote knowledge and skill improvement of individuals. *Learning potential*, is a key reason behind effective collaboration. It has been shown that the increase in learning one expects from collaboration yields fruitful coordination and higher quality contributions [2, 3]. For instance, in online fan-fiction communities, informal mentoring improves people's writing skills [11]. In this paper, we propose to explore how affinity between group members improves peer learning and address modeling, theoretical, and algorithmic challenges. To the best of our knowledge, our work is the first to examine algorithmic group formation with affinities for peer learning.

Group formation in online communities has been studied primarily in the context of task assignment [4–6, 14, 18]. The problem is often stated as: given a set of individuals and tasks, form a set of groups for the tasks that optimize some aggregated utility subject to constraints such as group size, maximum workload etc. Utility can be aggregated in different ways: the sum of individual skills, their product, etc [5]. Group formation is combinatorial in nature and proposed algorithms solve the problem under different constraints and utility definitions (e.g., [14]). Unlike these problems, we study how to form groups with the goal of maximizing peer learning under different affinities.

Our first contribution is to present principled models to formalize peer learning and affinity structures. We assume that a peer can only learn from another peer if the skill of the latter is strictly higher than the skill of the former [3]. The learning potential of a peer from a more skilled peer can then naturally be defined as the skill difference between the latter and the former [2, 3]. The learning potential of the latter from the former is null. We use that to formulate two common learning models (see picture below): LpA where each member learns from all higher skilled ones, and LpD where the least skilled member (resp., the most skilled) learns from (resp. teaches to) all others.



LpA: members learn from higher-skilled ones



LpD: the least skilled member learns from the most skilled one

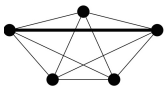
Affinity, on the other hand, depends on the application and can be expressed using common socio-demographic attributes or more generally, using models that capture psychological traits. We study our two learning models in conjunction with two common affinity structures (see picture below): AffD where group affinity is the

[1] https://movielens.org/
[2] http://quora.com/
[3] https://www.figure-eight.com/

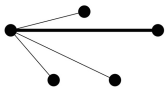
KDD '19, August 4–8, 2019, Anchorage, AK, USA

ACM ISBN 978-1-4503-6201-6/19/08…$15.00
https://doi.org/10.1145/3292500.3330945

smallest between all members, and AffC where group affinity is the smallest between a designated member (e.g., the least skilled or the most skilled) and all others. We investigate these two affinity scenarios through fact-checking and fact-learning applications.



AffD: smallest affinity between all pairs



AffC: smallest affinity between one member and others

Our second contribution is to study the formalized models systematically and present our theoretical findings. In its general form, our problem formulation is a bi-objective optimization, with the goal to build k equi-sized groups over a set of n members that maximize both learning potential and affinity. Interestingly, we prove that no variant of optimizing learning potential alone is hard to solve (LpD and LpA), however, the problems become NP-hard when affinity and group size constraints are considered. Therefore, our solution first finds k groups that yield the highest possible learning potential value and then transforms our two-objective problem into *a constrained optimization* that looks for k groups that optimize affinity, with that learning potential value as a constraint.

Our third contribution is algorithmic. We present a suite of scalable algorithms that form groups to maximize learning potential and optimize affinity within *constant approximation factors*. To attain their approximation guarantees, these algorithms assume that affinity satisfies triangle inequality [14]. Many similarity/distance measures such as Jaccard distance and edit distance are known to satisfy metric properties and these properties are usually assumed to design algorithms with guarantees [14]. Our technical contributions are summarized in Table 2.[4]

This paper is organized as follows. Our model and problem are provided in Section 2. Section 3 discusses various optimization problems: optimizing learning potential alone, optimizing affinity alone, and combining the two. Section 4 contains our algorithms. Real and simulated experiments are reported in Section 5. The related work and conclusion are given in Sections 6 and 7. Supplementary material with all proofs is provided in Section 8.

2 MODELING AND PROBLEM DEFINITION

We present our models following which we define the problems we tackle in this work.

Example 1. *We have a set of fact-checking tasks to be completed by* 12 *individuals with varying skills. We design questions to compute one skill per individual (e.g., on the British royal wedding) and obtain the skill values:* $\{2, 3, 1, 5, 6, 4, 9, 8, 10, 12, 14, 17\}$. *Each pair of individuals has an affinity that reflects how effectively they can collaborate based on their socio-demographics. Therefore, there are* $\binom{12}{2}$ *pairs of affinities forming a complete graph. We show a subset of that graph in Table 1 where each worker is identified by her skill. Our goal is to divide the workers into* 3 *equi-sized groups of* 4 *members each.*

[4]Theoretical proofs are presented in Section 8.

member	2	3	1	5	6	4	9	8	10	12	14	17
2	-	20	-	-	-	-	3	-	-	-	11	17
3	20	-	13	-	-	-	17	-	-	-	11	4
1	-	13	-	-	-	-	10	-	-	-	14	21
…												
17	17	4	21	-	-	-	-	-	-	-	6	-

Table 1: Partial Affinity Table for Example 1

2.1 Modeling

Group. A group is a set of individuals who will complete a task together. The group size is constant throughout task completion.

Skill. Each individual has an approximated skill reflecting an ability to perform a task. We obtain skills from standard tests and questionnaires to assess expertise level. Other approaches such as inferring skills from completed tasks [19], are also possible.

Affinity. Between every pair of individuals working on a task, affinity captures how well they get along. We express affinity as a similarity measure (higher values are better). We assume affinity satisfies triangle inequality [14]. Group affinity is the aggregation of affinities between its members.

Learning Potential. We define the learning potential between two individuals as the difference between their skill values. The learning potential is not a metric since for the person with the higher skill value, we set it to 0 [2, 3]. The learning potential for a group is the sum of learning potentials of its members.

We have a set of n individuals who are working on a collaborative task. Each w_i has a skill value $w_i^s \in R \geq 0$ representing an ability to complete a task. Our goal is to group them into k equi-sized groups such that the aggregated learning potential and affinity of the groups are maximized. Before formalizing the problem, we investigate variants of learning potential and affinity.

2.1.1 Learning Potential Models. Intuitively, the higher the learning potential of a group, the more likely its members will learn from each other. We examine two definitions.

Learning Potential - Diameter. We define LpD as the difference in skills between the most skilled and the least skilled members. This reflects that for a group, we are interested in maximizing the highest learning potential of the least skilled individual in that group.

$$\text{LpD}(g) = \max_{w_i \in g}(w_i^s) - \min_{w_j \in g}(w_j^s) \tag{1}$$

In Example 1, if we create the following 3 groups (we use a member's skill to represent her). $\mathcal{G} = \{g_1 = (2, 3, 1, 5), g_2 = (6, 4, 9, 8), g_3 = (10, 12, 14, 17)\}$ then, $\text{LpD}(g_1) = 5 - 1 = 4$, $\text{LpD}(g_2) = 9 - 4 = 5$, and $\text{LpD}(g_3) = 17 - 10 = 7$. The aggregated learning potential of the grouping $\mathcal{G}$ is 16.

Learning Potential - All. We define LpA as the sum of differences between each member's skill and that of all other members with higher skills.

$$\text{LpA}(g) = \sum_{w_i \in g} \sum_{(w_j \in g, s.t. w_i^s < w_j^s)} (w_j^s - w_i^s) \tag{2}$$

Given the previous grouping, we can compute $\mathcal{G} = \{g_1 = (2, 3, 1, 5), g_2 = (6, 4, 9, 8), g_3 = (10, 12, 14, 17)\}$, $\text{LpA}(g_1) = |1 -$

$2| + |1-3| + |1-5| + |2-3| + |2-5| + |3-5| = 13$, LpA$(g_2) = 17$, and LpA$(g_3) = 23$. The aggregated learning potential of the grouping under LpA is 53.

2.1.2 Affinity Models. It is important to look at the effect of affinity on learning since members with higher affinities are likely to learn better from each other and collaborate more effectively [11, 17, 18]. We examine two affinity variants.

Affinity - Diameter. We can formalize affinity as a complete graph $G = (V, E)$ where V is the set of n individuals and E contains weighted edges that correspond to the affinity between every pair of them. In this case, affinity satisfies triangle inequality. We refer to this case as AffD and define the affinity of a group as the minimum pairwise affinity of all its members as follows:

$$\text{AffD}(g) = \min_{w_i, w_j \in g} aff(w_i, w_j) \tag{3}$$

According to Example 1, if $g = \{2, 12, 14\}$ then
AffD$(g) = \min\{aff(2, 12), aff(2, 14), aff(12, 14)\}$, i.e., 4.

Affinity - Center. Affinity can also be defined based on the relationship between one member and all others. We refer to that as AffC and capture it as a graph $G = (V, E)$ where edges are defined between one designated member w_D and all others.

$$\text{AffC}(g) = \min_{w_D, w_j \in g} aff(w_D, w_j) \tag{4}$$

AffC captures the cases where the designated member is the least skilled or the most skilled. Similarly to AffD, in Example 1, if $g = \{2, 12, 14\}$ and the group center is 14 then
AffC$(g) = \min\{aff(2, 14), aff(12, 14)\}$ which corresponds to $aff(12, 14) = 6$.

For instance, when the task is collaborative fact-checking (e.g., check facts related to the British royal wedding), LpA reflects that each group member will learn from other members with higher skills and AffD captures agreement between the most two disagreeing members in the group. When LpA is combined with AffC, we can capture a task such as text editing where group members collaborate to correct grammar and spelling mistakes in text. In that case, one can intuitively assume that each group member will learn from other members with higher skills and that everyone must have affinity with the highest skilled member. Another example, AffD LpD captures a task where group members are asked to produce facts they believe to be true. In that case, the least skilled member learns from all others and all get along when stating facts.

2.2 Problem Definition

Given a set $W = \{w_1, ..., w_n\}$ of individuals with their corresponding skill values w_i^s, our goal is to form a grouping $\mathcal{G}$ that contains k equi-sized groups $g_1, g_2, ..., g_k$ that maximizes two objective functions, aggregated learning potential and aggregated affinity. More formally:

$$\begin{aligned} \underset{\mathcal{G}}{\text{maximize}} \quad & \sum_{i=1}^{k} LP(g_i), \sum_{i=1}^{k} Aff(g_i) \\ \text{s.t.} \quad & |\mathcal{G}| = k,\ |g_i| = \frac{n}{k} \end{aligned} \tag{5}$$

Problem	Algo.	Approx.	Time
(AffC LpD)	GrAffC-LpD	exact LpD, 3 AffC	$O(klogn + nlogk)$
(AffC LpA)	GrAffC-LpA	exact LpD, 3 AffD	$O(nlogn)$
(AffD LpD)	GrAffD-LpD	exact LpA, 6 AffC	$O(klogn + nlogk)$
(AffD LpA)	GrAffD-LpA	exact LpA, 6 AffD	$O(nlogn)$

Table 2: NP-Hard problems and technical results

where $LP(g_i)$ (resp. $Aff(g_i)$) refers to any of the learning potential (resp. affinity) definitions above.

Since the two objectives are incompatible with one another, our problem qualifies as *multi-objective*. Upon examining the learning potential expressions, we notice that these are polynomial time solvable problems, simply because the primary operation that these problems require is sorting. We present exact algorithms for the two learning potential problems in Section 3.1. The complexity of our problem lies within the affinity structure and the group size constraint. One way to solve our bi-objective optimization problem is therefore to transform it into a single-objective problem with constraints. We can rewrite Equation 5 as follows:

$$\begin{aligned} \underset{\mathcal{G}}{\text{maximize}} \quad & \sum_{i=1}^{k} Aff(g_i) \\ \text{s.t.} \quad & \sum_{i=1}^{k} LP(g_i) \geq OptLP \\ & |\mathcal{G}| = k,\ |g_i| = \frac{n}{k} \end{aligned} \tag{6}$$

where $OptLP$ is the optimal solution for learning potential maximization. Essentially, we are interested in finding a solution for the affinity objective on the Pareto front, that has the highest learning potential. In Section 4, we present approximation algorithms that find a feasible grouping (that maximizes learning potential) and offer provable constant approximation for affinities.

3 OPTIMIZATION

In this section, we first study how to optimize each of our two objectives individually, learning potential and affinity, and in the last subsection we begin studying our bi-objective optimizations by translating them into constrained optimization problems. This exercise has many benefits - (a) it offers a deeper understanding of the individual problems and (b) it provides perspective on how to combine them and design scalable solutions with provable guarantees (refer to Section 4). All proofs are in Section 8.

3.1 Optimizing Learning Potential

Our algorithmic endeavor begins by first describing solutions to group formation that maximize learning potential (LP) alone. Once we obtain the optimal LP value, we use that as a constraint when optimizing affinity (Equation 6 in Section 2.2). Fortunately, both LP problems are computationally tractable, and we present efficient algorithms that form a grouping with exact solutions. While different, our algorithms are designed in the same spirit as those designed to solve the value-based group formation [2] and the p-percentile

partitioning problem [3]. A central idea to those algorithms is to create a grouping based on sorting group members on skill values.

3.1.1 Learning Potential LpD. We want to form k groups that maximize the aggregated learning potential which in LpD is the maximum pairwise skill difference (Equation 4). LpD of a group is always determined by a single pair of its members, the least skilled and the most skilled ones. Therefore, if we have to form a single group, we just need to select the most and least skilled members and make them part of that group. The other members in the group could be any as their participation does not increase or decrease the LpD value. This seemingly simple logic sufficiently extends to forming k groups. To form k groups, we need to find two buckets with a total of $2k$ people, the most skilled bucket containing the k highest skilled workers (the i-worker in that bucket is referred to as $w_i^{s.high}$), and the least skilled bucket containing the k least skilled workers (the i-worker in that bucket is referred to as $w_i^{s.low}$). We can then form k pairs by grouping one member in the least skilled bucket with one in the most skilled bucket and placing them in the same group. The remaining $n - 2k$ workers can be distributed arbitrarily across the k groups, while keeping the group size the same (pseudo-code in Algorithm 1).

Applied to Example 1, this is akin to forming the least skilled bucket with participants of skill values $\{1, 2, 3\}$, the most skilled one with values $\{12, 14, 17\}$, and forming 3 pairs, each one representing a group of size 2, by pairing members across the buckets. We can state the following theorem:

Theorem 1. *Any pairing across the least skilled and most skilled buckets produces the optimal aggregated value for LpD.*

Based on Theorem 1, we can state that multiple groupings maximize the LpD value. This corollary is important, because it provides intuition on the challenges that arise when combining affinity with learning potential.

Corollary 1.1. *There are* $\frac{k! \times (n-2k)!}{(n/k-2)!^k}$ *possible groupings to maximize LpD.*

Lemma 1. *Computing one optimal grouping for LpD takes* $O(n + klogn)$.

3.1.2 Learning Potential LpA. The LpA of a group is the sum of skill differences between every member with every other more skilled member (Equation 3). The LpA of a set of k groups is the sum over the LpA of each group. What becomes intuitively apparent is that if one has to form one group to maximize LpA, one should always group the most skilled member with the remaining less skilled ones. This logic extends to creating k groups by sorting members on skills (in increasing or decreasing order), and creating n/k buckets, each with k members. To form a group of size n/k, we choose a member from each bucket and repeat this process k times.

Using Example 1, this is akin to sorting the skills of the participants and forming a total of 4 buckets:

$$\{1, 2, 3\}, \{4, 5, 6\}, \{8, 9, 10\}, \{12, 14, 17\}$$

We form the first group by arbitrarily selecting one member from each of these 4 buckets, for example, those with skills $\{1, 4, 8, 12\}$.

Algorithm 1 Algorithm to maximize LpD

```
input: set of workers W, k
output: a grouping G, OPTLpD
procedure LpD(W, k)
    OPTLpD ← 0
    create highest and lowest skill buckets with k workers each
    G ← a set of k empty groups
    for i in (1, ..., k) do
        pick w_m ∈ most_skilled and w_l ∈ least_skilled
        g_i ← {w_m, w_l}
        W ← W \ {w_l, w_m}
        OPTLpD ← OPTLpD + (w_m^s − w_l^s)
    end for
    while W is not empty do
        Assign w_i ∈ W in g_i, s.t g_i ≤ n/k.
    end while
end procedure
```

Then we repeat the process twice to get the 2 other groups, e.g., $\{2, 5, 9, 14\}$ and $\{3, 6, 10, 17\}$.

This algorithm turns out to be optimal - moreover, just like for LpD, all possible groupings across n/k buckets are permissible and will produce the same optimal LpA value.

Theorem 2. *Any grouping across the* n/k *buckets produces the optimal aggregated value for LpA.*

The proof is similar to that of Theorem 1. We omit the details for brevity.

Corollary 2.1. *There are* $k!^{n/k}$ *possible groupings for LpA.*

Lemma 2. *Computing one optimal grouping for LpA takes* $O(nlogn)$.

3.2 Optimizing Affinity

Since we express affinity as similarity, optimizing it amounts to minimizing distance. AffC takes the affinity graph over n members and a subset of k members as centers (teachers) as input, and intends to partition the remaining $n - k$ members into k equi-sized groups such that the sum of the radii (maximum distance between the center and a member in each group) is minimized. AffD, on the other hand, only takes the affinity graph over n members and k to partition the members into k equi-size groups, such that, the sum of the diameters (diameter of a group is the maximum pairwise distance in the group) of the grouping is minimized. We first present some theoretical results on the hardness of these two problems.

Even though it intuitively appears that AffC is an easier problem than AffD, both problems are NP-hard. The hardness of AffC is due to the group size constraint.

Theorem 3. *The decision version of the AffC problem in NP-Complete.*

Theorem 4. *The decision version of the AffD problem is NP-Complete.*

3.3 Optimizing Affinity with Learning Potential as a Constraint

Finally, we turn our attention to studying the four constrained optimization problems, with the objective to optimize affinity, while satisfying the learning potential value obtained from the algorithms in Section 3.1. Since affinity is modeled as a distance, our goal is to minimize that distance, considering the underlying affinity structure. Recall that LPD and LPA are polynomial-time problems and that we presented exact solutions for both in the previous section. To ease exposition, we will henceforth call the optimal values obtained for the LP problems as LP-* (it is either LPD or LPA). Our focus now is to study how to optimize AFF-* (AFFC or AFFD), with LP-* as constraints.

THEOREM 5. *The decision versions of AFF-* LP-* problems are NP-Complete.*

Our technical deep dive into these four problems is described in Section 4. We develop greedy algorithms that are extremely lean in computational time with constant approximation factors. Table 2 summarizes the 4 problem variants and our technical results.

4 CONSTRAINED OPTIMIZATION

We now present a suite of algorithms with theoretical guarantees to solve the four different variants of optimizing affinity with learning potential as a constraint. As our problems are NP-hard, we develop approximation algorithms that are scalable and bear theoretical guarantees. Our results are summarized in Table 2 and Section 8 contains the proofs.

Our algorithms are greedy and use the following intuition: LP-* problems are first solved and these solutions produce an intermediate grouping that has the optimal LP values. Our algorithms start from these solutions and greedily choose the rest of the members to output the final grouping that is guaranteed to have optimal LP values and provable constant approximation factors for affinity.

4.1 Algorithm for AFFC LPD

Our discussion of LPD in Section 3.1.1 stated that only $2k$ members (k most skilled and k least skilled) are needed to produce the optimal grouping. Our proposed Algorithm GRAFFC-LPD starts from there (recall Algorithm 1) - that is, it first identifies the $2k$ members which will guarantee the optimal LPD value (thereby satisfying the constraint of the optimization problem). These outputs are referred to as boundary members. That means, 2 members in each group are decided by now and a total of $2k$ members are decided for the grouping. In each group, the highest skilled member is the teacher and acts as the center for that group since AFFC is formalized as the maximum distance between that member and anyone else in the group. The rest of the grouping is performed in a greedy manner. For the remaining $n - 2k$ members, all we have to do is assign them to their respective closest center. Since each group has a size constraint, this greedy assignment may lead to sub-optimality - since for a member w_i, the group with the closest distance between its center and w_i may have reached its size and w_i may need to be assigned to a group such that the distance between w_i and its center c_i is larger (potentially worsening the AFFC value). But as we shall prove later, this greedy assignment cannot be arbitrarily worse, since affinity between members satisfies triangle inequality (pseudo-code in Algorithm 2).

Going back to Example 1, based on GRAFFC-LPD, initially we will have the following partial grouping : $g_1 = \{1, 12\}$, $g_2 = \{2, 17\}$, $g_3 = \{3, 14\}$. After that, Algorithm GRAFFC-LPD greedily adds 2 more members in each group that are not yet part of any group. For example, for g_1, it will add the member who has the highest affinity with 12 and the process will repeat.

Algorithm 2 Algorithm GRAFFC-LPD

input: a set W of n participants, k groups
output: a grouping $\mathcal{G}$
B = Call LPD(W, k)
C = the k highest skilled members in B that are k centers
Assign $w_i \in \{W - B\}$ to the closest center c_j s.t., $|g_j| \leq n/k$

THEOREM 6. *Algorithm GRAFFC-LPD accepts a* 3 *approximation factor to optimize AFFC.*

COROLLARY 6.1. *Running time of GRAFFC-LPD is* $O(klogn+nlogk)$

4.2 Algorithm for AFFC LPA

The idea of this greedy algorithm GRAFFC-LPA is similar to the previous one, that is start with the partial grouping that LPA returns. However, unlike LPD, LPA creates a set of n/k buckets (or partitions) (see Section 3.1.2) that dictate that forming an intra-partition group is forbidden, and any possible inter-partition groups will result in the same optimal LPA value. In fact, as Corollary 2.1 suggests, there are $(k!)^{n/k}$ possible groupings that yield the optimal LPA value. The challenge is to find one grouping that optimizes affinity.

GRAFFC-LPA begins by invoking the LPA procedure to compute n/k buckets that are sorted in increasing order of skills. It selects the teachers as the k members in the last buckets (they are the centers and they have the k highest skills). After that, GRAFFC-LPA operates in a greedy fashion. For the remaining $n - k$ members, it follow a similar approach as Algorithm GRAFFC-LPD. At each iteration, it chooses a member from the bucket and assigns it to the closest center. In Example 1, we create a total of 4 buckets:

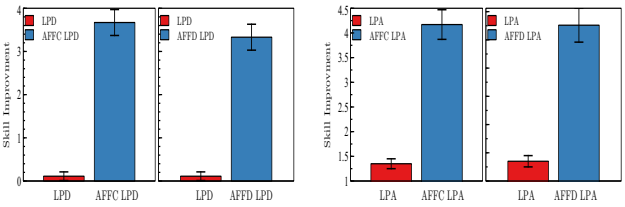
$$\{1, 2, 3\}, \{4, 5, 6\}, \{8, 9, 10\}, \{12, 14, 17\}$$

Next, we assign each high skilled member in the last bucket to a group and consider them as centers. As an example, $g_1 = \{12\}$, $g_2 = \{14\}$, and $g_3 = \{17\}$. Next, for the members of the first bucket, based on their affinities, 1 is assigned to g_1, 2 to g_3, and 3 to g_2. The process continues until all the buckets are empty.

Since each group has a size constraint, this greedy assignment may lead to sub-optimality - as it happened in GRAFFC-LPD. However, this greedy assignment cannot be arbitrarily worse, because affinity between members satisfies triangle inequality.

THEOREM 7. *Algorithm GRAFFC-LPA accepts a* 3 *approximation factor for AFFC.*

COROLLARY 7.1. *Running time of Algorithm GRAFFC-LPA is* $O(nlogn)$.



(a) (b)

Figure 1: Skill improvement with and w/o affinity in LpD (a) and LpA (b)

4.3 Algorithms for AffD Lp-*

There is an interesting relationship between AffC and AffD that merits further delineation. In the AffC problem, we want to minimize the distance from a center to the farthest member in the group (i.e., minimizing the radii). In AffD, we do not have any member as the center, rather we are interested to form groups to minimize the maximum distance (i.e., the diameter). The next theorem states that any solution for the former problem is a solution for the latter that is at most 2 times worse. Based on that, the greedy algorithms in Sections 4.1 and 4.2 could be used to solve AffD,Lp-* problems. We refer to these algorithms as GrAffD-LpD and GrAffD-LpA, respectively for the AffD LpD and AffD LpA problems. GrAffD-LpD is identical to GrAffC-LpD, and GrAffD-LpA is identical to GrAffC-LpA operationally. Their respective running times are the same as their counterparts.

Theorem 8. *Any solution for AffC gives a* 2 *approximate solution for AffD.*

Corollary 8.1. *Algorithms GrAffD-LpD and GrAffD-LpA have a* 6 *approximation factor for AffD.*

Proof. (sketch): Proofs are direct derivatives of Theorem 8. □

5 EXPERIMENTAL EVALUATIONS

Our experimental effort goes in two directions. In Section 5.1, we involve actual human workers and collaborative tasks. In the remaining three subsections, we describe synthetic data experiments.

5.1 Real Data Experiments

These experiments examine *if affinity brings added utility in peer learning*. They are designed for *collaborative fact-checking and fact-learning* and involve Amazon Mechanical Turk (AMT) workers in three stages: 1) pre-task skill assessment, 2) task completion in a group, and 3) post-task completion skill assessment.

Experimental setup and design. We design 4 HITs for the four variants of our problem that consider both AFF and LP, as well as 4 additional HITS that only consider LP (without AFF), a model similar to [2]. We recruit 100 workers from AMT who are redirected to an internal website. Each HIT contains the three aforementioned stages. The fact-checking tasks is about the British royal family. The details of the design and setup are provided in Supplemental Material in Section 8.2.

Evaluation criteria. In order to evaluate the effectiveness of affinity in peer learning, we measure the difference between each member's skills before and after task completion, and refer to that as *skill improvement*. We also measure the average number of comments in each group and the quality of contributions. These two criteria help us interpret worker engagement and skill improvement.

Summary of results. We compare with and without affinity counterparts for each problem variant (e.g., AffC LpD with LpD). Our results confirm that affinity improves learning potential substantially with statistical significance. Figure 1(a) contains the average skill improvement comparison of LpD with and without affinity and shows the important role of affinity. This is consistent with LpA (Figure 1(b)). We also observe that LpA has higher improvements, possibly because facts have many facets that one learns from more skillfull peers. Additionally, Figure 2 presents two sample interactions between two workers during task completion. In the first question, *Worker 2* provides a new piece of information about the Queen, which is set as one of the questions in the post-task skill assessment. This additional information helps *Worker 1* improve her skill during post-task assessment.

The Queen does not need a passport to travel True or False ?	• Worker 1: True. All British Passports are issued in the Name of Her Majesty, The Queen. • Worker 2 : I found an article which agrees with your findings. **Fun fact: she also doesn't need a driver's license or a license plate on her car.**
Members of the royal family have to accept absolutely all gifts.	• Worker 1 : (Mostly false; Large true in practice.) While I couldn't find any law requiring the Royals to accept all gifts. • Worker 2 : I found an article which says they make a list of all gifts they receive throughout the year and release it publicly. In addition, they donate many of their gifts.

Figure 2: A sample worker interaction

Finally, we anecdotally observe that higher learning potential yields higher quality task outcomes. On average, quality (computed as the average number of facts correctly identified by the groups), is higher for groups built with affinity (4.3 facts out of 5 are correct), compared to their counterparts built without affinity (3.9 out of 5).

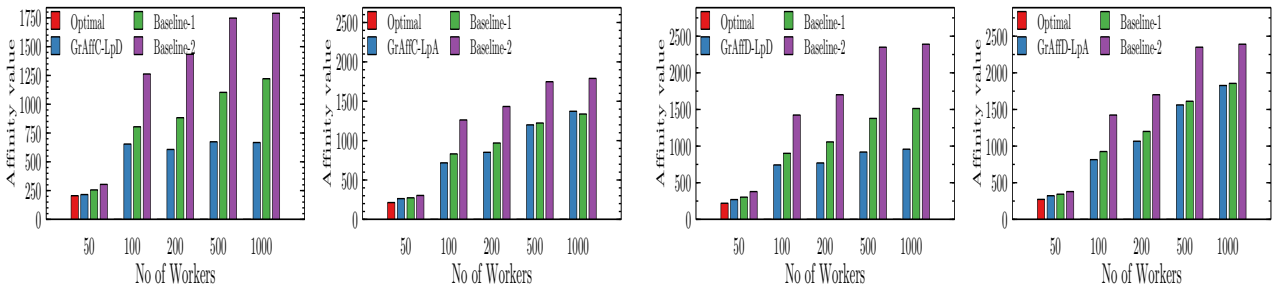
5.2 Synthetic Experiments Setup

These experiments evaluate the qualitative guarantees and the scalability of our algorithms. All algorithms are implemented in Python 3.6 using Intel Core i7 4GHz CPU and 16GB of memory and Windows operating system. All numbers are averages of 10 runs.

5.2.1 Implemented Algorithms. The closest works to ours [2, 3] do not consider affinity and cannot be used for synthetic data experiments. Hence, we implement three additional baselines:

Optimal. An Integer Linear Programming (ILP) algorithm that produces optimal values for Aff-* Lp-*. We use the PuLP library to build the ILP model whose exact ILP formulation is given in Section 8.2.

The rationale behind implementing ILP is to demonstrate that the theoretical approximation factors of our algorithms hold in practice. Since ILP is NP-hard, the algorithm does not terminate beyond $k = 3$ and $n = 50$.



(a) AffC LpD (b) AffC LpA (c) AffD LpD (d) AffD LpA

Figure 3: AFF-* LP-* values varying n for Normal Distribution

Baseline-1 (clustering-based). This baseline is motivated by the popular k-means algorithm. It starts with a random grouping and greedily swaps members across groups as long as that improves affinity, while satisfying group size. Once the grouping converges based on affinity, we check if it satisfies the optimum learning potential value (which could be derived efficiently in polynomial time). If not, we perform another set of swaps to move members across the groups until we find a grouping that reaches the optimal learning value.

Baseline-2. This is a simpler and efficient baseline. It first solves the learning potential problem and finds the seed members in each group that dictate the optimal learning value. The rest of the members are assigned randomly to groups by considering group size.

These solutions are compared with 4 of our algorithms GrAffC-LpD, GrAffC-LpA, GrAffD-LpD, GrAffD-LpA (refer to Section 4, whenever applicable).

5.2.2 Experimental Setup. We simulate a group of workers with two functions that capture the relationship between skill and affinity. Specifically, there are two random number generators, one produces the skill of each member and the other generates pairwise affinities that satisfy triangle inequality.

We consider two skill and affinity distributions: (a) *Normal*, where the mean and standard deviations are set to $\mu = 100$, $\sigma = 20$, respectively; (b) *Zipf*, where the value of the exponent α is set to 1.5.

Parameters: We vary n (the total number of individuals), k (the number of groups), and the skill and affinity distributions.

5.2.3 Summary of Results. 1. Our algorithms exhibit tighter approximation factors than the bounds we proved. Our algorithms also outperform the two baselines.
2. The approximation factors of the algorithms with a Normal skill distribution are better than Zipf.
3. All algorithms are highly scalable considering up to 10^6 members and 160 groups and only take seconds to run.

5.3 Quality Experiments (Synthetic)

We assess quality by measuring the approximation factor and the objective function value. Both of these are described considering affinity, per our problem definition. LP values are always optimal (as the algorithms for LP are exact) and we skip those details for brevity.

Algo.	Parameters	App. Factor
(GrAffC-LpD)	$[n = 15, k = 3]$	1.13(0.12)
	$[n = 50, k = 3]$	1.23(0.02)
(GrAffC-LpA)	$[n = 15, k = 3]$	1.04(0.07)
	$[n = 50, k = 3]$	1.02(0.03)
(GrAffD-LpD)	$[n = 15, k = 3]$	1.21(0.14)
	$[n = 50, k = 3]$	1.31(0.04)
(GrAffD-LpA)	$[n = 15, k = 3]$	1.18(0.11)
	$[n = 50, k = 3]$	1.19(0.12)

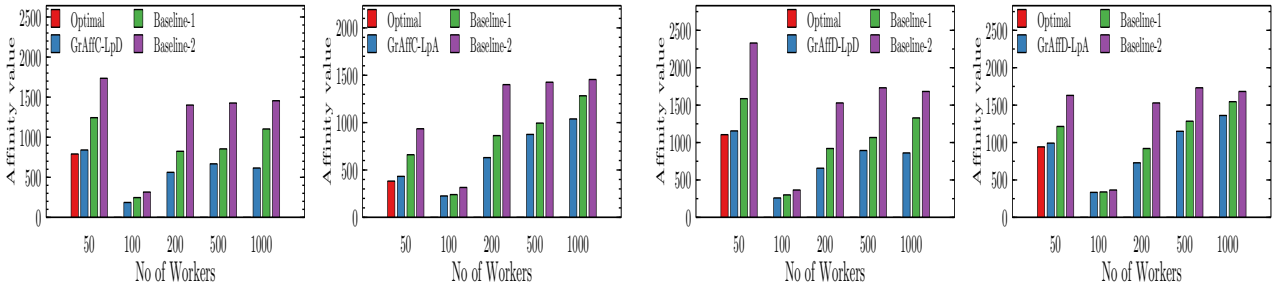
Table 3: Approximation factors

Default parameter setting. Unless otherwise stated, k is set to 25 and n to 1000.

Comparison against ILP: Table 3 presents the approximation factor of the 4 algorithms on a small dataset generated from Normal Distribution. For all the 4 algorithms, the approximation factor in practice is very tight and the deviation is always between 1 and 2.

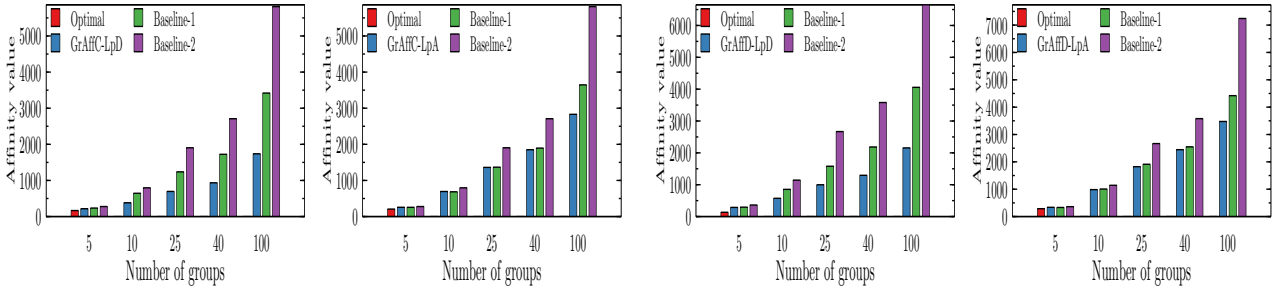
Varying n : Figure 3 reports the results of varying n for Normal distribution. Of course, ILP does not scale beyond $n = 50$, but it is easy to notice that for all the cases we could compare, our greedy solutions attain a very tight approximation factor (close to 1.5). Figure 4 shows the affinity values for Zipf distribution. Similar observations hold. Our proposed algorithms perform significantly better. There are two interesting observations in both Figures 3 and 4. Firstly, the grouping generated by our algorithm attains smaller objective value as the number of workers grow. Secondly, for Normally distributed data, we observe a consistent growth of objective value as the number of workers increases. This is not the case for the Zipfian distributed data. We conjecture that this is caused by the skew in the values generated from the Zipfian distribution. Some values are very large and others are vary small. Another important factor is that the data sampled from a Zipfian distribution consists of mostly duplicate values.

Varying k : Figures 5 and 6 present the results of varying k for Normal and Zipf distributions. The ILP for $k = 5$ is ran on $n = 50$. Our presented algorithms consistently outperform the other baselines. We observe that the change in k affects the objective value significantly more than change in the number of workers (n). We believe this is because larger k signifies more centers to assign workers to. Remember in Algorithm 2, we need to assign



(a) AffC LpD (b) AffC LpA (c) AffD LpD (d) AffD LpA

Figure 4: AFF-* LP-* values varying n for Zipf Distribution



(a) AffC LpD (b) AffC LpA (c) AffD LpD (d) AffD LpA

Figure 5: AFF-* LP-* Values varying k for Normal Distribution

workers to the closest center. This means for larger values of k, we would diverge from the optimal solution easier. In fact, k impacts our algorithm more than n.

5.4 Scalability Experiments (Synthetic)

We measure running times and compare with **Baseline-1**. We exclude ILP since it does not scale, and **Baseline-2** since it produces inferior objective values. Running time is reported in seconds.

Default parameter settings. We found that Normal and Zipf skill distributions have identical running times for each variant of Aff-* Lp-* problems. We also note that, as proved in Section 4.3, the running time of AffD is identical to that of AffC, considering their respective Lp-* counterparts. Therefore, we only present results for Aff-* LpD and Aff-* LpA. We vary n and k with defaults set to $n = 100000$ and $k = 5$.

Results. Figure 7 presents results. Our algorithms are highly scalable and take seconds only. The algorithms run linearly with varying n and k which confirms our theoretical analysis.

6 RELATED WORK

Our work studies computational aspects and relates to team formation and computer-supported learning.

Team formation was first studied to form a single group with one objective and later a 2-approximation algorithm was proposed for bi-criteria team formation in social networks [14]. In [4, 5], Anagnostopoulos et al. propose online algorithms for the balanced social task assignment problem. Capacitated assignment was studied in a follow up work [16]. Generalized density sub-graph algorithms were later proposed [20]. [6, 18] study the problem of forming teams for task assignment considering affinity. In [7], the hardness of forming groups to optimize group satisfaction is studied under different group satisfaction semantics.

Unlike ours, none of these works study the problem of peer learning, hence their proposed modeling and algorithmic solutions do not apply.

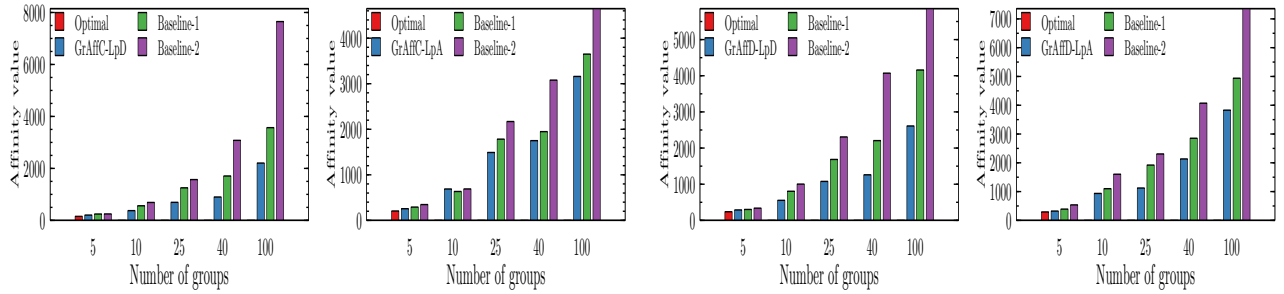
Computer-Supported Collaborative Learning (CSCL). Social science has a long history of studying non-computational aspects of computer-supported collaborative learning [9, 10]. With the development of online educational platforms (such as, Massive Open Online Courses or MOOCs), several parameters were identified for building effective teams: (1) individual and group learning and social goals, (2) interaction processes and feedbacks [21], (3) roles that determine the nature and group idiosyncrasy [10].

To the best of our knowledge, the closest to our work are [1–3], where quantitative models are proposed to promote group-based learning, albeit without affinity.

Our work is grounded in social science and takes a computational approach to the design of scalable solutions with guarantees.

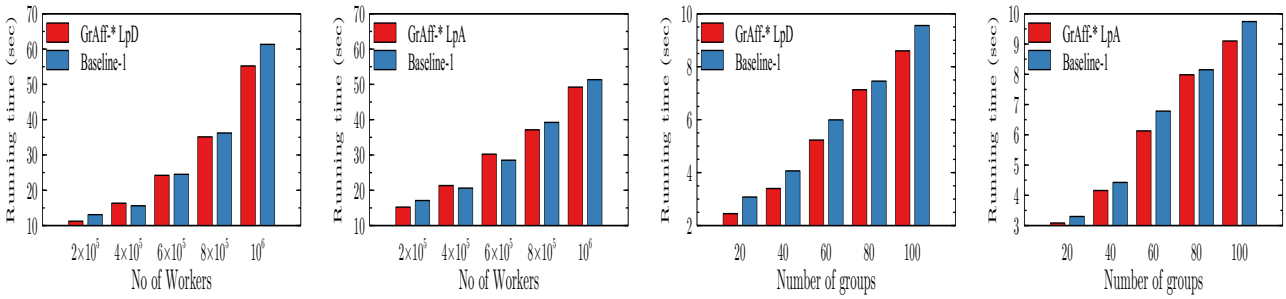
7 CONCLUSION

We examine online group formation where members seek to increase their *learning potential* via task completion with two learning models and affinity structures. We formalize the problem of forming a set of k groups with the purpose of optimizing peer learning under different affinity structures and propose constrained optimization formulations. We show the hardness of our problems and develop



(a) AffC LpD (b) AffC LpA (c) AffD LpD (d) AffD LpA

Figure 6: AFF-* LP-* values varying k for Zipf Distribution



(a) Aff-* LpD varying n (b) Aff-* LpA varying n (c) Aff-* LpD varying k (d) Aff-* LpA varying k

Figure 7: Scalability Results

4 scalable algorithms with constant approximation factors. Our experiments with real workers demonstrate that considering affinity structures drastically improves learning potential, and our synthetic data experiments corroborate the qualitative and scalablity aspects of our algorithms.

ACKNOWLEDGMENTS

The work of Mohammadreza Esfandiari, Dong Wei, and Senjuti Basu Roy are supported by the National Science Foundation Grant No.: 1814595 and Ofice of Naval Research, grant no: N000141812838.

REFERENCES

[1] Rakesh Agrawal, Behzad Golshan, and Evangelos Papalexakis. 2016. Toward data-driven design of educational courses: A feasibility study. *EDM* (2016).
[2] Rakesh Agrawal, Behzad Golshan, and Evimaria Terzi. 2014. Grouping students in educational settings. In *SIGKDD*.
[3] Rakesh Agrawal, Sharad Nandanwar, and Narasimha Murty Musti. 2017. Grouping Students for Maximizing Learning from Peers. In *EDM*.
[4] Aris Anagnostopoulos et al. 2010. Power in unity: forming teams in large-scale community systems. In *CIKM*.
[5] Aris Anagnostopoulos et al. 2012. Online team formation in social networks. In *WWW*.
[6] Senjuti Basu Roy et al. 2015. Task assignment optimization in knowledge-intensive crowdsourcing. *VLDB* (2015).
[7] Senjuti Basu Roy, Laks VS Lakshmanan, and Rui Liu. 2015. From group recommendations to group formation. In *SIGMOD*.
[8] Hongyun Cai et al. 2017. From community detection to community profiling. *VLDB* (2017).
[9] Elizabeth G Cohen. 1994. Restructuring the classroom: Conditions for productive small groups. *Review of educational research* (1994).
[10] Thanasis Daradoumis et al. 2002. Supporting the composition of effective virtual groups for collaborative learning. In *ICCE*. IEEE.
[11] Sarah Evans et al. 2017. More Than Peer Production: Fanfiction Communities as Sites of Distributed Mentoring. In *CSCW*.
[12] Mike Gartrell et al. 2010. Enhancing group recommendation by incorporating social relationship interactions. In *GROUP*.
[13] Michal Kompan and Mária Bieliková. 2014. Social Structure and Personality Enhanced Group Recommendation. In *EMPIRE*.
[14] Theodoros Lappas, Kun Liu, and Evimaria Terzi. 2009. Finding a team of experts in social networks. In *SIGKDD*.
[15] Isabella Lari et al. 2016. Partitioning a graph into connected components with fixed centers and optimizing cost-based objective functions or equipartition criteria. *Networks* (2016).
[16] Anirban Majumder et al. 2012. Capacitated team formation problem on social networks. In *SIGKDD*.
[17] Vicente Rodríguez Montequín et al. 2010. Using Myers-Briggs Type Indicator (MBTI) for Assessment Success of Student Groups in Project based Learning. In *CSEDU*.
[18] Habibur Rahman et al. 2015. Task assignment optimization in collaborative crowdsourcing. In *ICDM*.
[19] Habibur Rahman et al. 2015. Worker Skill Estimation in Team-Based Tasks. *PVLDB* (2015).
[20] Rangapuram et al. 2013. Towards realistic team formation in social networks based on densest subgraphs. In *WWW*.
[21] Ivan Srba and Maria Bielikova. 2015. Dynamic group formation as an approach to collaborative learning support. *TLT* (2015).

8 SUPPLEMENTAL MATERIAL

We present the proofs and important formulation necessary to reproduce the results in the paper in this section. We also provide additional information regarding our experimental evaluation.

8.1 Theorems and Proofs

8.1.1 *Proof of Theorem 1.*

Proof. (sketch) Consider the set of k least skilled members and k most skilled members. It is easy to see that changing the assignment of the least skilled members would not change the overall sum of the skill difference. LpD of the grouping is:

$$OPT\text{LpD} = (w_1^{s.high} - w_1^{s.low}) + (w_2^{s.high} - w_2^{s.low}) + \ldots + (w_k^{s.high} - w_k^{s.low})$$

Indeed, any possible grouping across the buckets over these $2k$ members will not affect the sum, and thus the LpD value. □

8.1.2 *Proof of Corollary 1.1.*

Proof. The members in the highest and the lowest skilled buckets could be paired in $k!$ groupings. The remaining $(n-2k)$ members are to be placed over $(n/k-2)$ positions in each group, and a total over k groups. This gives rise to $\frac{k!\times(n-2k)!}{(n/k-2)!^k}$ groupings. □

8.1.3 *Proof of Theorem 2.*

Proof. The proof is very similar to the proof of LpD. Consider a grouping that we get by running the above algorithm. We sort the members based on their skill values and we create x buckets. The first k workers will go into the first bucket and so on. We create a group by choosing 1 member of each bucket. After k iterations, we obtain our grouping. Without loss of generality, assume that the highest skilled member of group i has the skill value of S_i and the other members have s_i^j where j denotes the bucket that this member has been chosen from. Consider the grouping $G = \{g_1 = (S_1, s_1^1, s_1^2), g_2 = (S_2, s_2^1, s_2^2), \ldots, g_k = (S_k, s_k^1, s_k^2)\}$. We can show that swapping two members from the same bucket will not change the LpA optimal value. Without loss of generality, consider a new grouping G' where the position of s_i^j and $s_{i'}^j$ is swapped. Now consider the LpA value for G and G'. We can show that the difference between these two scores is 0. This holds when we pick the lowest skilled member.

$$LP(\mathcal{G}) = \sum_{i=1}^{k} \sum_{j=1}^{x-1} (S_i - s_i^j) \tag{7}$$

In the Equation 7, the only difference between G and G' is in group i and i'. More accurately, we need to show that $(S_i - s_i^j) + (S_{i'} - s_{i'}^j)$ in the grouping G is equal to $(S_i - s_{i'}^j) + (S_{i'} - s_i^j)$. It's easy to see that these two are identical hence the proof.

□

8.1.4 *Proof of Theorem 3.*

Proof. (sketch) It is easy to see that the problem is in NP. To prove the NP-hardness, the reduction is straightforward (there is a one-to-one correspondence) if we consider the uniform p-centered min-max partition problem as the source problem, which is proved to be NP-hard [15] for general graphs. We omit the rest of the details for brevity. □

8.1.5 *Proof of Theorem 4.*

Proof. For simplicity, we consider a simpler scenario, where affinity (distance) is binary - 0/1.

For this binary scenario, the decision version of the Affinity-All problem is as follows: given a set of n members, is there a grouping of k equal sized groups, such that the sum of diameters of the grouping is k?

It is easy to see that the problem is in NP. To prove NP-hardness, we use the well-known exact cover by 3-Sets (X3C) for reduction. The decision version of X3C is as follows: given a finite set X with $|X| = 3q$ elements and a collection C of 3-element subsets of X, does C contain an exact cover for X, that is, a sub-collection $C' \subseteq C$, where C' contains exactly q subsets, such that every element of X occurs in exactly one member of C'?

Given an instance of X3C, we reduce it to an instance of AffD in the following way: Each element in X is a member. Therefore, the total number of members $n = 3q$. The affinity graph is a weighted complete graph among the n members and it involves adding edge weights between every pair of members. Each subset of 3 elements in C represents 3 members in this graph, and the edge weights between them gets the value 1. This is a polynomial time operation and the number of operations involved in this is the size of C. After that, we need to resolve all the edge weights that are across the subsets. For that, we start considering all the triangles with some unresolved edge weights.

There are three possible scenarios to handle in this process: (1) all 3 edges unresolved, (2) 2 edges unresolved, (3) 1 edge unresolved. For the first case, we can safely add the weight of 0 to each of such edge (this happens when the subsets are fully disjoint). For the second scenario (this happens when 2 nodes in the triangle are part of the same subset but the third node is part of a different subset), one of the unresolved edges gets 1 and the other gets 0. Finally, for the third scenario (this happens when one member in the triangle is part of both subsets), the unresolved edge gets a 0. We note that this step is again fully polynomial and takes at most $O\binom{n}{3}$ time. After completing this step, we will have assigned all the edge weights in the affinity graph. It could be shown that the affinity graph constructed this way satisfies triangle inequality.

After that, we set $k = q$. Now, the reduction is complete. Notice that X3C $\leq_P$ AffD. There exists a solution to the X3C problem if and only if a solution to our instance of AffD problem exists with the total diameter value q (or k). This completes the proof. □

8.1.6 *Proof of Theorem 6.*

Proof. (sketch) Without loss of generality, let us assume a worst case scenario of 3 groups as shown in Figure 8, where members p_1, p_2, p_3 dictate the AffC score of these three groups that are centered around c_2,c_3,c_1, respectively. Because of the greedy assignment, p_1 is assigned to the center c_2, p_2 is assigned to c_3, but at the end because of the size constraints p_3 gets a really bad assignment of c_1. Distance between p_1 and c_2, i.e., $d(p_1, c_2) = \alpha_1$, similarly $d(p_2, c_3) = \alpha_2$, and $d(p_3, c_1) = \alpha_3$. The optimal assignment would have given rise to a different assignment though (as shown

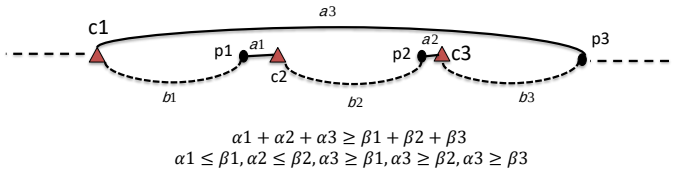


Figure 8: Upper bound of approximation factor

in the dotted line), where $p_1 \in c_1, p_2 \in c_2, p_3 \in c_3$. $d(p_1, c_1) = \beta_1, d(p_2, c_2) = \beta_2$, and $d(p_3, c_3) = \beta_3$. Let OPT denote the optimum AffC value, such that $OPT = \beta_1 + \beta_2 + \beta_3$. Of course, $\alpha_1 + \alpha_2 + \alpha_3 \geq \beta_1 + \beta_2 + \beta_3$. But it is easy to notice that

$$\alpha_3 \leq (\alpha_1 + \beta_1 + \alpha_2 + \beta_2 + \beta_3) \tag{8}$$

$$\alpha_3 \leq (2\beta_1 + 2\beta_2 + \beta_3) \tag{9}$$

Because of the triangle inequality, this is indeed true, $\alpha_1 + \beta_1 \leq 2\beta_1$ (because $\alpha_1 \leq \beta_1$ what the greedy algorithm GrAffC-LpD will ensure. Therefore, we can write that,

$$\alpha_1 + \alpha_2 + \alpha_3 \leq (3\beta_1 + 3\beta_2 + \beta_3) \tag{10}$$

$$\leq 3(\beta_1 + \beta_2 + \beta_3) \tag{11}$$

$$\leq 3 \times OPT \tag{12}$$

It is easy to notice that this argument easily extends to an arbitrary number of groups. Hence the proof. □

8.1.7 Proof of Theorem 8.

Proof. (sketch:) Consider a solution to AffC. Consider that for a group g_i, the distance from the center c_i to the farthest member is α_i. Assume that w_i is the member with this distance equal to α_i. Based on the triangle inequality, we can easily show that in the worst case, there is another member $w_j \in g_i$ where $d(w_i, w_j) < d(w_i, c_i) + d(w_j, c_i) < 2 \times \alpha_i$. Hence, any algorithm that solves AffC also solves AffD with a 2 approximation factor. □

8.2 Additional Experimental Details

8.2.1 Real Data Experiment Design and Setup. In order to replicate our real data experiments in Section 5.1, we present additional information here.

Experimental design. One of our fact-checking tasks is about the British royal family. These experiments are run in three different stages. We first run a pre-task skill test to assess the skills of each worker using 8 true/false questions for which we know the true answer. We set questionnaires for that purpose. Next stage, we set up a collaborative document that contains 5 facts about the royal family and where workers in the same group can collaborate, comment, and edit. Workers are asked to discuss if these facts are true, and provide further evidence that support their answer. Finally, each worker takes a post-task completion skill test that is again 8 true/false questions on the royal family. We also explicitly ask each worker what they have learned by completing that task. We design similar studies for fact-learning, where workers have to actually propose facts with supporting evidence. To keep this experiment tractable, we form groups of size 3 and run 3 different samples of the same experiment. This also allows us to analyze results with statistical significance. We pay each worker $2, if all three stages are completed. Each experiment must run over a window of 24 hours to account for differences in time.

Affinity calculation. For simplicity, we capture affinity as the Euclidean distance between their socio-demographic data (specifically, age, country, education) obtained from AMT. There exists other sophisticated measures such as MBTI tests for project-based learning [17]. We nevertheless note that the simple measures that we have used have been shown to be useful affinity indicators [18].

8.2.2 ILP Formulation of Optimal Algorithm in Section 5.2.1. We formalize the ILP solution using the below formulation. It's easy to implement this formulation in any Linear Programming software. For this experiments we use PuLP, a popular python based ILP package.

$$\underset{\mathcal{G}}{\text{optimize}} \sum_{i=1}^{k} \text{Aff-}^*(g_i)$$

$$\text{s.t.} \sum_{i=1}^{k} LP(g_i) \geq \text{Lp-}^*$$

$$\text{Aff-}^*(g_i) = \sum_{j=1}^{n} \sum_{m=1}^{n} x_{i,j} * x_{i,m} * Aff(w_j, w_m), \forall i = 1 \ldots k$$

$$\sum_{j=1}^{n} x_{i,j} = n/k, \forall i = 1 \ldots k$$

$$x_{i,j} = 0/1 \quad (i = 1 \ldots k \;\&\; j = 1 \ldots n)$$