Aggregating Inconsistent Information: Ranking and Clustering

Nir Ailon[*] Moses Charikar[†] Alantha Newman[‡]

May 2008

Abstract

We address optimization problems in which we are given contradictory pieces of input information and the goal is to find a globally consistent solution that minimizes the extent of disagreement with the respective inputs. Specifically, the problems we address are rank aggregation, the feedback arc set problem on tournaments, and correlation and consensus clustering. We show that for all these problems (and various weighted versions of them), we can obtain improved approximation factors using essentially the same remarkably simple algorithm. Additionally, we almost settle a long-standing conjecture of Bang-Jensen and Thomassen and show that unless NP⊆BPP, there is no polynomial time algorithm for the problem of minimum feedback arc set in tournaments.

1 Introduction

The problem of aggregating inconsistent information from many different sources arises in numerous contexts and disciplines. For example, the problem of ranking a set of contestants or a set of alternatives based on possibly conflicting preferences is a central problem in the areas of voting and social choice theory. Combining k different complete ranked lists on the same set of n elements into a single ranking, which best describes the preferences expressed in the given k lists, is known as the problem of *rank aggregation*. This problem dates back to as early as the late 18th century when Condorcet and Borda each proposed voting systems for elections with more than two candidates [Con85, Bor81]. There are numerous applications in sports, databases, and statistics [DKNS01a, FKS03] in which it is necessary to effectively combine rankings from different sources. Another example of aggregating information is the problem of integrating possibly contradictory clusterings from existing data sets into a single representative clustering. This problem is known as *consensus clustering* or *ensemble clustering* and can be applied to remove noise and incongruencies from data sets [FS03] or combine information from multiple classifiers [Str02].

In the last half century, rank aggregation has been studied and defined from a mathematical perspective. In particular, Kemeny proposed a precise criterion for determining the "best" aggregate ranking[1] [Kem59, KS62]. Given n candidates and k permutations of the candidates, $\{\pi_1, \pi_2, \ldots, \pi_k\}$,

[*] Partly supported by a Princeton University Honorific Fellowship. Email: nailon@cs.princeton.edu

[†] Supported by NSF ITR grant CCR-0205594, DOE Early Career Principal Investigator award DE-FG02-02ER25540, NSF CAREER award CCR-0237113, an Alfred P. Sloan Fellowship and a Howard B. Wentz Jr. Junior Faculty Award. Email: moses@cs.princeton.edu

[‡] Work done while visiting Princeton University, supported by Moses Charikar's Alfred P. Sloan fellowship. Email: alantha@dimacs.rutgers.edu

[1] Historically known as *Kemeny aggregation*.

a *Kemeny optimal* ranking of the candidates is the ranking π that minimizes a "sum of distances", $\sum_i^k d(\pi, \pi_i)$, where $d(\pi_j, \pi_k)$ denotes the number of pairs of candidates that are ranked in different orders by π_j and π_k.[2] For example, if $\pi_j = (1, 2, 3, 4)$ and $\pi_k = (2, 3, 1, 4)$, then $d(\pi_j, \pi_k) = 2$ since elements 1 and 2 appear in different orders in the two rankings as do elements 1 and 3. In other words, a Kemeny optimal ranking minimizes the number of pairwise *disagreements* with the given k rankings. Throughout this paper we will refer to the problem of finding a Kemeny optimal ranking as Rank-Aggregation.

More recently, Rank-Aggregation has been studied from a computational perspective. Finding a Kemeny optimal ranking is NP-hard [BTT89] and remains NP-hard even when there are only four input lists to aggregate [DKNS01a]. This motivates the problem of finding a ranking that *approximately* minimizes the number of disagreements with the given input rankings. Several 2-approximation algorithms are known [DG77, DKNS01a]. In fact, if we take the best of the input rankings, then the number of disagreements between this ranking and the k input rankings is no more than twice optimal.

The *feedback arc set problem* on *tournaments* is closely related to the Rank-Aggregation problem. A tournament is a directed graph $G = (V, A)$ such that for each pair of vertices $i, j \in V$, either $(i, j) \in A$ of $(j, i) \in A$. The minimum feedback arc set is the smallest set $A' \subseteq A$ such that $(V, A - A')$ is acyclic. The size of this set is exactly the minimum number of backward edges induced by a linear ordering of V. Throughout the paper, we refer to this problem as Fas-Tournament. This problem turns out to be useful in studying Rank-Aggregation, but is also interesting in its own right. For example, imagine a sports tournament where each player plays against every other player once: How should we rank the players based on these possibly non-transitive (inconsistent) outcomes? The complementary problem to finding a minimum feedback arc set is the *maximum acyclic subgraph* problem, also known as the *linear ordering* problem. Rank-Aggregation can be cast as a special case of *weighted* Fas-Tournament, where the objective is to minimize the total weight of backward edges in a linear order of the vertices. When the weight of edge (i, j) is the fraction of input rankings that order i before j, solving Rank-Aggregation is equivalent to solving this weighted Fas-Tournament instance.

The last problem we consider is that of clustering objects based on complete but possibly conflicting pairwise information. An instance of this problem can be represented by a graph with a vertex for each object and an edge labeled "+" or "$-$" for each pair of vertices, indicating that two elements should be in the same or different clusters, respectively. The goal is to cluster the elements so as to minimize the number of "$-$" edges within clusters and "+" edges crossing clusters. This problem is known as Correlation-Clustering (on complete graphs) [BBC04]. A useful application of Correlation-Clustering is optimally combining the output of different machine learning classifiers [Str02, BBC04]. Bansal, Blum and Chawla provide in-depth descriptions of other applications of Correlation-Clustering [BBC04]. An analog to Rank-Aggregation is known as Consensus-Clustering. In this problem, we are given k clusterings of the same set of n elements. The goal is to find a clustering that minimizes the number of pairwise disagreements with the given k clusterings. This problem can also be used to optimally combine datasets. For example, Consensus-Clustering has been applied to the problem of integrating data resulting from experiments that measure gene expression [FS03].

1.1 Previous Work

The minimum feedback arc set problem can be approximated to within $O(\log n \log \log n)$ in general graphs [ENSS98, Sey95] and has (at least) the same hardness of approximation as the vertex cover

[2] The distance function $d(\cdot, \cdot)$ is in fact a distance function and is known as the *Kendall tau* distance.

problem [Kar72], which is 1.36 [DS02]. More than a decade ago, Bang-Jensen and Thomassen conjectured that Fas-Tournament is NP-hard [BJT92]. However, for the past decade, no progress has been made on settling this conjecture. In contrast, the minimum feedback *vertex* set problem on tournaments is NP-hard [Spe89] and is approximable to within 5/2 [CDZ00].

We are not aware of any approximation for Fas-Tournament that improves on the bound for the feedback arc set problem in general graphs. The complementary maximization problem on tournaments has been studied; Arora, Frieze and Kaplan [AFK96] and Frieze and Kannan [FK99] gave PTASs for the maximum acyclic subgraph problem in dense graphs, which implies a PTAS for the problem on tournaments. Interestingly, since the appearance of the conference version of this work [ACN05], Kenyon-Mathieu and Schudy [KMS07] used the maximization PTAS as a main component in a minimization PTAS. This significantly improves on the result in this work for the ranking problems (in particular for Rank-Aggregation), since here we guarantee only constant approximation factors. Neverthelss, our algorithms are very simple and practical and more suitable for applications. Refer to Section 10 for a complete survey and comparison with followup work.

There are two well-known factor 2-approximation algorithms for Rank-Aggregation. Since both Rank-Aggregation and Consensus-Clustering are equivalent to finding the median of a set of points with a metric distance function, it easy to see that choosing one of the given lists or given clusters at random, yields a 2-approximation algorithm. We refer to these algorithms as Pick-A-Perm and Pick-A-Cluster, respectively. The *Spearman's footrule* distance between two permutations π_i and π_j on n elements is defined to be: $F(\pi_i, \pi_j) = \sum_{k=1}^{n} |\pi_i(k) - \pi_j(k)|$. The footrule distance is no more than twice the Kemeny distance [DG77] and can be computed in polynomial time via a minimum cost matching [DKNS01a, DKNS01b]. These observations yield another 2-approximation.

Correlation-Clustering has been studied both on general and complete graphs. Both the minimization and maximization versions have been investigated. Bansal, Blum and Chawla gave the first constant-factor approximation for the problem of minimizing disagreements on the complete graph [BBC04]. This factor was improved to 4 by rounding a linear program [CGW03]. The weighted version of Correlation-Clustering, in which edges have fractional ± assignments has also been studied. Each edge is assigned fractional values w_{ij}^{+} and w_{ij}^{-} rather than a discrete "+" or "−" label. When the edge weights satisfy the *probability constraints* (i.e. $w_{ij}^{+} + w_{ij}^{-} = 1$ for all edges), the best previous approximation factor was 7 [GMT05, BBC04]. When the edge weights satisfy the probability and the triangle inequality constraints (see Section 1.2), the best previous approximation factor was 3 [GMT05]. Correlation-Clustering on complete graphs is MAX-SNP-hard [CGW03] and Consensus-Clustering is NP-hard [Wak98]. However, Consensus-Clustering is not known to be NP-hard if the number of input clusters is constant [FS03].

1.2 Our Results

We give improved approximation algorithms for the following optimization problems:

- Fas-Tournament,
- Rank-Aggregation,
- Correlation-Clustering and
- Consensus-Clustering.

We show that they can all be approximated using essentially the same remarkably simple algorithm. For example, the algorithm for Fas-Tournament, called KwikSort, is as follows: First, we pick a random vertex i to be the "pivot" vertex. Second, we place all vertices connected to i with an in-edge on the left side of i and all vertices connected to i with an out-edge on the right side of i. We then recurse on the two tournaments induced by the vertices on each side.

The analysis of KwikSort yields a 3-approximation algorithm for Fas-Tournament, improving on the best-known previous factor of $O(\log n \log \log n)$. Our analysis relies on a new technique for arguing a lower bound for Fas-Tournament by demonstrating a fractional packing of edge disjoint directed triangles. The KwikSort algorithm is presented in Section 3, in which we introduce the basic ideas we use throughout the paper. In Section 4, we extend these ideas to approximate weighted Fas-Tournament.

We further extend our techniques to Rank-Aggregation in Section 5. We convert the Rank-Aggregation instance into a weighted Fas-Tournament instance, which we convert to an unweighted Fas-Tournament instance using the *majority tournament* (see Definition 1), and we then run KwikSort on this majority tournament. Although this algorithm by itself is yet another 2-approximation, the following is an 11/7-approximation: run both KwikSort and Pick-A-Perm and output the best solution. This improved approximation ratio is due to the fact that each algorithm does well on instances in which the other algorithm does poorly.

A simple lower bound on the value of an optimal solution for the weighted Fas-Tournament is to take the sum over all vertices $i < j$ of $\min\{w_{ij}, w_{ji}\}$. In contrast, our analysis uses a stronger lower bound based on the weight of directed triangles ("*bad triangles*") in the majority tournament. Interestingly, the analysis of our simple combinatorial algorithm bounds the integrality gap of a natural LP relaxation for Fas-Tournament. In fact, it demonstrates an LP dual solution based on probabilities of random events occurring during the execution.

For Correlation-Clustering and Consensus-Clustering we present similar combinatorial algorithms and analyses, with a different notion of "*bad triplets*". Interestingly, this gives results that are analogous to the results for Fas-Tournament and Rank-Aggregation and improve upon previously known approximation factors. We discuss Correlation-Clustering and Consensus-Clustering in Section 6.

Our analysis is applied to various cases of weighted Fas-Tournament (resp. weighted Correlation-Clustering). More precisely, we analyze the following cases:

(i) **Probability Constraints:** $w_{ij} + w_{ji} = 1$ (resp. $w^+_{ij} + w^-_{ij} = 1$) for all $i, j \in V$.

(ii) **Triangle Inequality:** $w_{ij} \leq w_{ik} + w_{kj}$ (resp. $w^-_{ij} + w^-_{jk} \leq w^-_{jk}$) for all $i, j, k \in V$.

(iii) **Aggregation:** Edge weights are a convex combination of actual permutations (resp. clusters). Constraints (i) and (ii) are implied in this case.

As indicated, in instances of weighted Fas-Tournament that correspond to Rank-Aggregation, the edge weights obey both the probability constraints and triangle inequality, although these instances corresponding to Rank-Aggregation are even more restricted.

Table 1 summarizes the approximation factors we achieve for the different scenarios with the combinatorial algorithms. Additionally, we consider LP relaxations for Fas-Tournament and Correlation-Clustering. After choosing a pivot vertex, instead of deterministically placing vertices on the right or left side (in KwikSort), or in a cluster (in KwikCluster), we decide randomly based on LP values. This results in vastly improved approximation factors.

Finally, we show that Fas-Tournament has no polynomial time algorithm assuming NP$\not\subseteq$BPP. The question of NP-hardness of Fas-Tournament has been a long-standing conjecture of Bang-Jensen and Thomassen [BJT92]. We show a randomized reduction from the problem of finding a

	Ordering	Clustering	Ordering-LP	Clustering-LP
Unweighted Instances	3 (*)	3 (4)[CGW03]	5/2	5/2
Probability Constraints (i)	5 (*)	5 (9) [CGW03, BBC04]	5/2	5/2
Triangle Inequality (ii)	2 (*)	N/A (**)		
Probability Constraints + Triangle Inequality (i,ii)	2 (*)	2 (3) [GMT05]	2	2
Aggregation (iii)	11/7 (2)	11/7 (2)	4/3	4/3

Table 1: The previous best-known factors are shown in parentheses. (*) The best-known factor was the $O(\log n \log\log n)$ algorithm [ENSS98, Sey95] for digraphs. (**) Our techniques cannot directly be applied to weighted Correlation-Clustering with triangle inequality but no probability constraints.

minimum feedback arc set in general digraphs (which is known to be NP-hard) to the special case of tournaments. This construction has been recently derandomized by Noga Alon [Alo06], and the conjecture is therefore proven completely. We present the weaker randomized version here.

In Section 7, we extend our ideas to round LP's for Fas-Tournament and Correlation-Clustering. In Section 8, we prove certain polynomial inequalities that are stated in several lemmas in the preceding sections. In Section 9, we prove hardness results for Fas-Tournament. In Section 10 we discuss work that has appeared since the publication of the conference version of this work [ACN05], and finally in Section 11 we discuss open problems.

2 Preliminaries and Definitions

We study the following problems in this paper. In what follows, we fix a ground set $V = \{1, \ldots, n\}$.

- Fas-Tournament: (Minimum Feedback Arc Set in Tournaments) We are given a tournament $G = (V, A)$ (a digraph with either $(i, j) \in A$ or $(j, i) \in A$ for all distinct $i, j \in V$). We want to find a permutation π on V minimizing the number of pairs ordered pairs (i, j) such that $i <_\pi j$ and $(j, i) \in A$ (backward edges with respect to π).[3] In a *weighted* Fas-Tournament instance, we are given weights $w_{ij} \geq 0$ for all ordered $i, j \in V$. We want to find a permutation π on V minimizing $\sum_{i,j:i<_\pi j} w_{ji}$. Clearly, the unweighted case can be encoded as a 0/1 weighted case.

- Rank-Aggregation: We are given a list of permutations (rankings) corresponding to k voters $\pi_1, .., \pi_k$ on V. We want to find a permutation π minimizing the sum of distances $\sum_{i=1}^{k} d(\pi, \pi_i)$, where $d(\pi, \rho)$ is the number of ordered pairs (i, j) such that $i <_\pi j$ but $j <_\rho i$ (the Kemeny distance).

- Correlation-Clustering: Between any two unordered $i, j \in V$ we either have a "+" or a "−" relation, indicating that i and j are similar or different, respectively. We let E^+ (resp. E^-)

[3] By $i <_\pi j$ we mean that π ranks i before j.

denote the set of pairs $i \neq j$ which are "+"-related (resp. "−"-related). We want to find disjoint clusters $C_1, \ldots, C_m$ covering V and minimizing the number of disagreement pairs ("+" pairs in different clusters or "−" pairs in the same cluster). In a *weighted* CORRELATION-CLUSTERING instance, we assign for each pair i, j two weights $w_{ij}^+ \geq 0$ and $w_{ij}^- \geq 0$. The cost of a clustering will now be the sum of w_{ij}^+ over all i, j in different clusters, plus the sum of w_{ij}^- over all i, j in the same cluster. Clearly, the unweighted case can be encoded as a 0/1 weighted case.

- CONSENSUS-CLUSTERING: We are given a list of clusterings corresponding to k voters $\mathcal{C}_1, \ldots, \mathcal{C}_k$ of V, and we wish to find one clustering $\mathcal{C}$ that minimizes $\sum_{i=1}^{k} d(\mathcal{C}, \mathcal{C}_i)$, where the distance $d(\mathcal{C}, \mathcal{D})$ between two clusterings is the number of unordered pairs $i, j \in V$ that are clustered together by one and separated by the other.

For a weighted FAS-TOURNAMENT instance, we will apply our algorithm for FAS-TOURNAMENT on an unweighted graph to a $m$ajority tournament, which is an unweighted tournament that corresponds to the input weighted tournament. Similarly, a weighted CORRELATION-CLUSTERING instance has a corresponding unweighted $m$ajority instance.

Definition 1 *Given an instance* (V, w) *of weighted* FAS-TOURNAMENT, *we define the unweighted* majority tournament $G_w = (V, A_w)$ *as follows:* $(i, j) \in A_w$ *if* $w_{ij} > w_{ji}$. *If* $w_{ij} = w_{ji}$, *then we decide* $(i, j) \in A_w$ *or* $(j, i) \in A_w$ *arbitrarily.*

Given an instance (V, w^+, w^-) *of weighted* CORRELATION-CLUSTERING, *we define the unweighted* majority instance (V, E_w^+, E_w^-) *as follows:* $(i, j) \in E_w^+$ *if* $w_{ij}^+ > w_{ij}^-$, *and* $(i, j) \in E_w^-$ *if* $w_{ij}^- > w_{ij}^+$. *If* $w_{ij}^+ = w_{ij}^-$, *then we decide arbitrarily.*

Note that although the majority instances depend on the weights of the weighted instances, they are *unweighted* instances.

We will use (i, j, k) to denote the *directed triangle* $(i \to j, j \to k, k \to i)$. It will be clear from the context whether a triangle is the set of its vertices or its edges.

3 Minimum Feedback Arc Set in Tournaments

Let $G = (V, A)$ be a FAS-TOURNAMENT instance. We present the following algorithm KWIKSORT for approximating it.

```
KwikSort($G = (V, A)$)

  If $V = \emptyset$ then return empty-list
  Set $V_L \to \emptyset, V_R \to \emptyset$.
  Pick random pivot $i \in V$.

  For all vertices $j \in V \setminus \{i\}$:
      If $(j, i) \in A$ then
          Add $j$ to $V_L$ (place $j$ on left side).
      Else (If $(i, j) \in A$)
          Add $j$ to $V_R$ (place $j$ on right side).

  Let $G_L = (V_L, A_L)$ be tournament induced by $V_L$.
  Let $G_R = (V_R, A_R)$ be tournament induced by $V_R$.

  Return order KwikSort($G_L$),$i$,KwikSort($G_R$).
    (Concatenation of left recursion, i, and right recursion.)
```

In our analysis, we will use the following notation. Let C^{OPT} denote the cost of an optimal solution. Let C^{KS} denote the cost of KwikSort on $G = (V, A)$.

Theorem 2 KwikSort *is a randomized algorithm for* Fas-Tournament *with expected cost at most three times the optimal cost.*

Proof: We want to show that $E\left[C^{KS}\right] \leq 3C^{OPT}$. An edge $(i, j) \in A$ becomes a backward edge if and only if there exists a third vertex k such that (i, j, k) form a directed triangle in G and k was chosen as a pivot when all three were input to the same recursive call. Pivoting on k would then place i to its right and j to its left, rendering edge (i, j) backward. In this case, we will charge a unit cost of the backward edge (i, j) to the directed triangle (i, j, k). Let T denote the set of directed triangles. For a directed triangle $t \in T$, denote by A_t the event that one of its vertices is chosen as pivot when all three are part of the same recursive call. Let p_t denote the probability of event A_t. Now we observe, that a triangle t is charged a unit cost exactly when A_t occurs, and it can be charged at most once. Therefore, the expected cost of KwikSort is exactly $E\left[C^{KS}\right] = \sum_{t \in T} p_t$.

Clearly, if we had a set of *edge disjoint* triangles, then its cardinality would be a lower bound for C^{OPT}. This is also true *fractionally*: If $\{\beta_t\}_{t \in T}$ is a system of nonnegative weights on triangles in T such that for all $e \in A$, $\sum_{t: e \in t} \beta_t \leq 1$, then $C^{OPT} \geq \sum_{t \in T} \beta_t$. Indeed, consider the following LP relaxation for the problem: minimize $\sum_{e \in A} x_e$, subject to $x_{e_1} + x_{e_2} + x_{e_3} \geq 1$ for edge sets $\{e_1, e_2, e_3\} \in T$, and $x_e \geq 0$ for all $e \in A$. The solution to this LP clearly lower bounds C^{OPT}. It is easy to show that a packing $\{\beta_t\}$ is a feasible solution to the dual LP, hence a lower bound on the optimal. Specifically, let $\mathcal{C}$ represent the set of directed cycles in G, and let y_c correspond to cycle $c \in \mathcal{C}$. Then the dual LP is:

$$\begin{aligned} & \max \sum_{c \in \mathcal{C}} y_c \\ \forall e \in E, & \sum_{c \in \mathcal{C}: e \in c} y_c \leq 1 \\ & 0 \leq y_c \leq 1. \end{aligned}$$

We will demonstrate such a *packing* using the probabilities p_t. Let $t = (i, j, k)$ be some triangle. Conditioned on the event A_t, each one of the 3 vertices of t was the pivot vertex with probability 1/3, because all vertices input to a recursive call are chosen as pivot with equal probability. Therefore, any edge $e = (i, j)$ of t becomes a backward edge with probability 1/3 (still, conditioned on A_t). More formally, if we let B_e denote the event that e becomes a backward edge, then

$$\Pr\left[B_e \wedge A_t\right] = \Pr\left[B_e | A_t\right] \Pr\left[A_t\right] = \frac{1}{3} p_t .$$

The event $B_e \wedge A_t$ means that the backwardness of edge e was charged to triangle t to which it is incident. The main observation of this proof is as follows: for two different triangles $t, t' \in T$ sharing an edge e, the events $B_e \wedge A_t$ and $B_e \wedge A_{t'}$ are disjoint. Indeed, an edge e can be charged to only one triangle t incident to e. Therefore, for all $e \in E$,

$$\sum_{t : e \in t} \frac{1}{3} p_t \leq 1 \ . \tag{1}$$

So $\{p_t/3\}_{t \in T}$ is a fractional packing of T. Thus, $C^{OPT} \geq \sum_{t \in T} p_t/3 = E\left[C^{KS}\right]/3$, as required. □

4 Minimum Feedback Arc Set in Weighted Tournaments

Let (V, w) be a weighted FAS-TOURNAMENT instance, where $w \in (\mathbb{R}^+)^{n(n-1)}$. We suggest the following approximation algorithm: construct the unweighted majority tournament $G_w = (V, A_w)$ and return the ordering generated by KWIKSORT(G_w). We analyze this algorithm.

For an edge $e = (i, j) \in A_w$, we let $w(e) = w_{ij}$, and $\overline{w}(e) = w_{ji} = 1 - w(e) \leq w(e)$. Fix an optimal solution π^*, and let $c^*(e)$ denote the cost incurred to it by $e = (i, j) \in A_w$, that is, $c^*(e) = w(e)$ if $j <_{\pi^*} i$, else $c^*(e) = \overline{w}(e)$. So $C^{OPT} = \sum_{e \in A_w} c^*(e)$. Let T denote the set of directed triangles in G_w. For any $t = (e_1, e_2, e_3) \in T$, we define $c^*(t) = c^*(e_1) + c^*(e_2) + c^*(e_3)$ and $w(t) = w(e_1) + w(e_2) + w(e_3)$. Note that $c^*(t)$ is always less than $w(t)$. Finally, let C^{KS} denote the cost the solution returned by KWIKSORT(V, G_w).

Lemma 3 *For an instance (V, w) of weighted* FAS-TOURNAMENT*, if there exists a constant $\alpha > 0$ such that $w(t) \leq \alpha c^*(t)$ for all $t \in T$, then $E\left[C^{KS}\right] \leq \alpha C^{OPT}$, that is,* KWIKSORT$(G_w)$ *is an expected α-approximate solution.*

Proof: Note that for any triangle t, any ordering will incur cost at most $w(t)$ on the edges of this triangle, whereas the optimal cost is $c^*(t)$. The assumption that $w(t) < \alpha c^*(t)$ means that we do not incur much more cost than the optimal solution. In order to extend this to the whole graph, we generalize the triangle packing idea presented in Section 3.

When KWIKSORT is run on G_w, an edge $e \in A_w$ is *heavily charged* if it becomes a backward edge, and thus incurs the heavy cost $w(e)$. It is *lightly* charged if it incurs the light cost $\overline{w}(e)$. Clearly, $e = (i, j) \in A_w$ is *heavily charged* if and only if a third vertex k is chosen as pivot when all three i, j, k are in the same recursive call, and (i, j, k) form a directed triangle in G_w. We charge this cost to triangle $t = (i, j, k)$. Again we consider the set T of directed triangles in G_w, and their corresponding events A_t with probability p_t (see Section 3). Fix a triangle $t \in T$ with edges e_1, e_2, e_3. Conditioned on A_t, each of e_1, e_2 and e_3 are equally likely to be heavily charged, so the expected charge of t is $\frac{1}{3} p_t w(t)$. The probability that an edge $e \in A_w$ does not incur a heavy cost (not charged to a triangle

$t \in T$) is exactly $1 - \sum_{t:e\in t} \frac{1}{3}p_t$. Therefore, $E\left[C^{KS}\right] = B^{KS} + F^{KS}$, where

$$B^{KS} = \sum_{t\in T} \frac{1}{3} p_t w(t)$$
$$F^{KS} = \sum_{e\in A_w} \left(1 - \sum_{t:e\in t} \frac{1}{3} p_t\right) \overline{w}(e).$$

We rearrange the sum $C^{OPT} = \sum_{e\in T} c^*(e)$ as $C^{OPT} = B^{OPT} + F^{OPT}$, where

$$B^{OPT} = \sum_{t\in T} \frac{1}{3} p_t c^*(t)$$
$$F^{OPT} = \sum_{e\in A_w} \left(1 - \sum_{t:e\in t} \frac{1}{3} p_t\right) c^*(e) \ .$$

Notice that for all $e \in A_w$, the term $(1 - \sum_{t:e\in t} \frac{1}{3}p_t)$ is nonnegative (see Section 3). Obviously, $F^{KS} \leq F^{OPT}$, because $\overline{w}(e) \leq c^*(e)$ for any $e \in A_w$. Therefore, if for some $\alpha > 0$, $w(t) \leq \alpha c^*(t)$ for all t, then $E\left[C^{KS}\right] \leq \alpha C^{OPT}$ as required. □

Lemma 4 *If the weights satisfy the probability constraints ($w_{ij} + w_{ji} = 1$), then $w(t) \leq 5c^*(t)$ for all $t \in T$. If the weights satisfy the triangle inequality constraints ($w_{ij} \leq w_{ik} + w_{kj}$), then $w(t) \leq 2c^*(t)$.*

Proof: First assume probability constraints on the weights. In this case, we claim that $w(t) \leq 5c^*(t)$. Indeed, in this case $w(e) \geq 1/2$ for all $e \in A_w$, and $\overline{w}(e) = 1 - w(e)$. Fix a triangle t containing edges e_1, e_2, e_3, and assume

$$1/2 \leq w(e_1) \leq w(e_2) \leq w(e_3) \leq 1 \ . \tag{2}$$

Clearly, $w(t) = w(e_1) + w(e_2) + w(e_3) \leq 2 + w(e_1)$. Any solution has to direct at least one of the edges in t backwards, therefore $c^*(t) \geq w(e_1)$. Since $w(e_1) \in [1/2, 1]$, we therefore have $w(t) \leq 5c^*(t)$. Consequently, KwikSort has an expected approximation ratio of at most 5 on weighted tournament instances with probability constraints on the weights.

Now we assume that the edge weights satisfy the triangle inequality. Fix $t \in T$ with edge weights $w(e_1), w(e_2), w(e_3)$. By the triangle inequality,

$$\begin{aligned} w(e_3) &\leq \overline{w}(e_1) + \overline{w}(e_2) \\ w(e_1) &\leq \overline{w}(e_2) + \overline{w}(e_3) \\ w(e_2) &\leq \overline{w}(e_3) + \overline{w}(e_1) \end{aligned} \tag{3}$$

Summing up, we get $w(t) \leq 2(\overline{w}(e_1) + \overline{w}(e_2) + \overline{w}(e_3))$. But $c^*(t) \geq \overline{w}(e_1) + \overline{w}(e_2) + \overline{w}(e_3)$, because the optimal solution must at least pay the lower cost at each edge. This concludes the proof. □

In the conference version [ACN05], a weaker bound of 3 was proven for the triangle inequality constraints only case and 2 for the combined constraints. This improvement in Lemma 4 is due to Warren Schudy.

Combining Theorem 3 and Lemma 4, we get

Theorem 5 *Running algorithm* KwikSort *on G_w gives an expected* 5 *and* 2 *approximation for the probability constraints case and the triangle inequality constraints case, respectively.*

5 An Improved Approximation Ratio for Rank Aggregation

Let $\{\pi_1, \ldots, \pi_k\}$ be a Rank-Aggregation instance over some V. Consider the corresponding equivalent weighted Fas-Tournament instance (V, w) (where w_{ij} is the fraction of inputs ranking i before j). Clearly, this weight system $\{w_{ij}\}$ is a convex combination of acyclic tournaments. Therefore, by linearity, the edge weights obey the probability constraints and the triangle inequality constraints. Theorem 5 shows that we get a 2 approximation for this case, but the additional structure in these instances allows us to improve upon this factor. As stated in the introduction, there already exists a well known 2-approximation algorithm for Rank-Aggregation:

```
Pick-A-Perm($\{\pi_1, \pi_2, \ldots \pi_k\}$)
  Output a permutation $\pi_i$ chosen uniformly at random
  from the input permutations.
```

(In practice, we can pick the permutation π_i that minimizes the cost, but we use the randomized version for the analysis.) Let C^{PAP} denote the cost of Pick-A-Perm on the Rank-Aggregation instance. Let $G_w = (V, A_w)$ be the corresponding unweighted *majority* tournament. Let $z(e) = 2w(e)\overline{w}(e)$, where $w(e)$ and $\overline{w}(e)$ are defined as in Section 4. We claim that

$$E\left[C^{PAP}\right] = \sum_{e \in A_w} z(e)\ . \tag{4}$$

Indeed, edge $e \in A_w$ becomes a backward (respectively, forward) edge with probability $\overline{w}(e)$ (respectively, $w(e)$), in which case it incurs the cost of $w(e)$ (respectively, $\overline{w}(e)$). For a directed triangle $t = (e_1, e_2, e_3) \in T$, we let $z(t) = z(e_1) + z(e_2) + z(e_3)$. The following theorem shows how to analyze a "convex combination" of KwikSort and Pick-A-Perm:

Theorem 6 *If there exist constants $\beta \in [0, 1]$ and $\gamma > 0$ such that*

$$\beta w(t) + (1-\beta) z(t) \le \gamma c^*(t) \text{ for all } t \in T, \text{and}$$
$$\beta \overline{w}(e) + (1-\beta) z(e) \le \gamma c^*(e) \text{ for all } e \in A_w,$$

then the best of KwikSort *and* Pick-A-Perm *is a γ-approximation for* Rank-Aggregation.

Proof: We use the notation $C^{OPT}, F^{OPT}, B^{OPT}, c^*(e), c^*(t)$ defined in Section 4. We rearrange (4) as $E\left[C^{PAP}\right] = B^{PAP} + F^{PAP}$, where

$$B^{PAP} = \sum_{t \in T} \frac{1}{3} p_t z(t), \qquad F^{PAP} = \sum_{e \in A_w} \left(1 - \sum_{t : e \in t} \frac{1}{3} p_t\right) z(e)\ .$$

If we now have β, γ as in the statement of the theorem, then (keeping in mind the crucial fact that $\left(1 - \sum_{t:e\in t} \frac{1}{3}p_t\right) \geq 0$ for all $e \in A_w$),

$$\begin{aligned}
\beta E\left[C^{KS}\right] + (1-\beta)E\left[C^{PAP}\right] &= \beta B^{KS} + (1-\beta)B^{PAP} + \beta F^{KS} + (1-\beta)F^{PAP} \\
&= \sum_{t\in T} \frac{1}{3}p_t \left(\beta w(t) + (1-\beta)z(t)\right) \\
&\qquad + \sum_{e\in A_w} \left(1 - \sum_{t:e\in t} \frac{1}{3}p_t\right) \left(\beta \overline{w}(e) + (1-\beta)z(e)\right) \\
&\leq \sum_{t\in T} \frac{1}{3}p_t \gamma c^*(t) + \sum_{e\in A_w} \left(1 - \sum_{t:e\in t} \frac{1}{3}p_t\right) \gamma c^*(e) \\
&= \gamma C^{OPT} \ ,
\end{aligned}$$

as required. □

Lemma 7 *For all $t \in T$, $\frac{3}{7}w(t) + \frac{4}{7}z(t) \leq \frac{11}{7}c^*(t)$, and for all $e \in A_w$, $\frac{3}{7}\overline{w}(e) + \frac{4}{7}z(e) \leq \frac{11}{7}c^*(e)$.*

Proof: The second inequality in the lemma is obtained by verifying the simple fact that $\overline{w}(e) \leq c^*(e)$ and $z(e) \leq 2c^*(e)$ for all $e \in A_w$. To prove the first inequality, we want to show that

$$f(t) = \frac{3}{7}w(t) + \frac{4}{7}z(t) - \frac{11}{7}c^*(t) \leq 0, \tag{5}$$

where (slightly changing notation) $t = (w_1, w_2, w_3)$ and

$$\begin{aligned}
w(t) &= w_1 + w_2 + w_3 \\
z(t) &= 2w_1(1-w_1) + 2w_2(1-w_2) + 2w_3(1-w_3) \\
c^*(t) &= 1 - w_2 + 1 - w_3 + w_1 \\
1/2 &\leq w_1 \leq w_j \leq 1 \text{ for } j = 2, 3 \\
w_1 &+ w_2 + w_3 \leq 2
\end{aligned}$$

The proof can be completed by finding the global maximum of $f(t)$ on the defined polytope using standard techniques of multivariate calculus. □

Note that for $(w_1, w_2, w_3) = (1/2,\ 3/4,\ 3/4)$ we obtain $w(t) = 2$, $z(t) = 5/4$ and $c^*(t) = 1$, so (5) is tight. Theorem 8 follows from Theorem 6 and Lemma 7, using $\beta = 3/7$ and $\gamma = 11/7$:

Theorem 8 *The best of* KwikSort *on G_w and* Pick-A-Perm *is an expected* 11/7 *approximation for* Rank-Aggregation.

In using Theorem 6 to derive bounds we can also take advantage of a priori knowledge of the system of weights w. We illustrate this using the special case of only $k = 3$ voters, a case of independent interest in applications [CCMR06]:

Lemma 9 *If $k = 3$, then for all $t \in T$, $\frac{2}{5}w(t) + \frac{3}{5}z(t) \leq \frac{6}{5}c^*(t)$ and for all $e \in A_w$, $\frac{2}{5}\overline{w}(e) + \frac{3}{5}z(e) \leq \frac{6}{5}c^*(e)$.*

Proof: In this special case, we have that $w(e) \in \{2/3, 1\}$ for all $e \in A_w$, and $w(e_1) = w(e_2) = w(e_3) = 2/3$ for all $t = (e_1, e_2, e_3) \in T$, therefore $w(t) = 2, z(t) = 4/3$ and $c^*(t) \geq 4/3$. The inequalities can now be easily verified. □

Theorem 10 follows from Theorem 6 and Lemma 9, using $\beta = 2/5$ and $\gamma = 6/5$:

Theorem 10 *The best of* KwikSort *on* G_w *and* Pick-A-Perm *is an expected* $6/5$ *approximation for* Rank-Aggregation *when there are* $k = 3$ *voters.*

6 Correlation Clustering and Consensus Clustering

In this section, we show how to apply the techniques presented in Section 3 to Correlation-Clustering and Consensus-Clustering. Recall that our goal is to minimize disagreements. In Fas-Tournament, we used "bad triangles" in tournaments to charge the disagreements in our solution. In Correlation-Clustering, disagreements in the solution can also be charged to "bad triplets", which will be defined shortly. Thus, the bad triplets replace the role taken by the directed triangles in tournaments. Let (V, E^+, E^-) be a Correlation-Clustering instance. Our algorithm KwikCluster, which is an analog of KwikSort, is defined as follows:

```
KwikCluster($G = (V, E^+, E^-)$)

  If $V = \emptyset$ then return $\emptyset$
  Pick random pivot $i \in V$.
  Set $C = \{i\}, V' = \emptyset$.

  For all $j \in V, j \neq i$:
      If $(i, j) \in E^+$ then
          Add $j$ to $C$
      Else (If $(i, j) \in E^-$)
          Add $j$ to $V'$

  Let $G'$ be the subgraph induced by $V'$.

  Return $C \cup$ KwikCluster($G'$) .
```

As in the analysis of KwikSort, a pair i, j incurs a unit cost if a third vertex k is chosen as pivot when the triplet (i, j, k) is in the same recursive call, and there are two "+" and one "−" relations among i, j, k (doesn't matter in which order). A triplet (i, j, k) is therefore a *bad triplet* if it has two "+" and one "−" relations.[4] Let T denote the set of (not necessarily disjoint) bad triplets. For each $t = (i, j, k) \in T$ we define A_t as the event that *all three* i, j, k *are in the same recursive call when the first one among them was chosen as pivot.* Let p_t denote the probability of A_t. The analysis continues identically to that of KwikSort.

Theorem 11 *Algorithm* KwikCluster *is a randomized expected* 3*-approximation algorithm for* Correlation-Clustering.

[4] A Correlation-Clustering instance with no bad triplets induces a consistent clustering, just as a tournament with no 3-cycles is acyclic. Our algorithms have an optimal cost of 0 on these instances.

Now let (V, w^+, w^-) be a weighted Correlation-Clustering instance, where $w^+, w^- \in (\mathbb{R}^+)^{\binom{n}{2}}$. Unlike weighted Fas-Tournament, we will *only* consider weight systems that satisfy the probability constraints $w^+_{ij} + w^-_{ij} = 1$. We create the unweighted majority Correlation-Clustering instance $G_w = (V, E^+_w, E^-_w)$ and return the clustering generated by KwikCluster(G_w).

Triangle inequality constraints in weighted Correlation-Clustering have the following form: for all i, j, k, $w^+_{ij} + w^+_{jk} + w^-_{ik} \leq 2$. (Equivalently, $w^-_{ik} \leq w^-_{ij} + w^-_{jk}$.) Theorem 12 is analogous to Theorem 5:

Theorem 12 *Algorithm* KwikCluster *on G_w is a 5 (respectively, 2) approximation for weighted* Correlation-Clustering *with probability constraints (respectively, with probability and triangle inequality constraints combined).*

The proof is almost identical to that of Theorem 5, with "$+ + -$" (bad) triplets in G_w replacing the role of directed (bad) triangles in tournaments.

Solving Consensus-Clustering is equivalent to solving weighted Correlation-Clustering with w^+_{ij} (respectively, w^-_{ij}) as the fractional number of input clusters with a "+" (respectively, "−") relation between i and j. This weighted Correlation-Clustering instance obeys both the probability constraints and the triangle inequality constraints, but we can do better than the 2 approximation guaranteed by Theorem 12. Analysis almost identical to the one in Section 5 gives an expected 11/7 approximation for this case. The KwikCluster is coupled with Pick-A-Cluster, which is defined analogously to Pick-A-Perm: Simply return a cluster chosen uniformly at random from the list.

Theorem 13 *The best of* KwikCluster *on G_w and* Pick-A-Cluster *has an expected approximation ratio of at most $\frac{11}{7}$ for* Consensus-Clustering.

7 Using the Pivot Scheme for Rounding the LP

We show how the techniques introduced above can be used for rounding the LP's for Fas-Tournament and Correlation-Clustering. We consider the LP's given in Figure 1 [Pot80, CGW03]. Given a solution to the LP, we consider algorithms LP-KwikSort and LP-KwikCluster (Figure 1) for rounding the solutions for Fas-Tournament and Correlation-Clustering, respectively. The main idea of these algorithms is that, after we choose some pivot, we use the LP solution variables to randomly decide where to put all other vertices, instead of deciding greedily. We note that our LP-based algorithms only solve the LP once and use the same LP solution in all recursive calls.

Theorem 14 *Our sorting LP rounding algorithm* LP-KwikSort *obtains the following approximation ratios on weighted* Fas-Tournament*instances:*

- 5/2 *when the weights satisfy the probability constraints,*
- 2 *when the weights satisfy the probability and the triangle inequality constraints, and*
- 4/3 *for* Rank-Aggregation.

The result for Rank-Aggregation *is obtained by returning the better of* LP-KwikSort *and* Pick-A-Perm.

Theorem 15 *Our clustering LP rounding algorithm* LP-KwikClustering *obtains the following approximation ratios on weighted* Correlation-Clustering *instances:*

LP for weighted Fas-Tournament

$$\begin{gathered}\text{minimize } \sum_{i<j}(x_{ij}w_{ji} + x_{ji}w_{ij}) \text{ s.t.}\\ x_{ik} \le x_{ij} + x_{jk} \text{ for all distinct } i, j, k\\ x_{ij} + x_{ji} = 1 \text{ for all } i \ne j\\ x_{ij} \ge 0 \text{ for all } i \ne j\end{gathered}$$

LP for weighted Correlation-Clustering

$$\begin{gathered}\text{minimize } \sum_{i<j}(x^+_{ij}w^-_{ji} + x^-_{ji}w^+_{ij}) \text{ s.t.}\\ x^-_{ik} \le x^-_{ij} + x^-_{jk} \text{ for all distinct } i, j, k\\ x^+_{ij} + x^-_{ij} = 1 \text{ for all } i \ne j\\ x^-_{ij}, x^+_{ij} \ge 0 \text{ for all } i \ne j\end{gathered}$$

LP-KwikSort(V, x)

A recursive algorithm for rounding the LP for weighted Fas-Tournament*. Given an LP solution $x = \{x_{ij}\}_{i,j \in V}$, returns an ordering on the vertices.*

```
If $V = \emptyset$ then return empty-list
Pick random pivot $i \in V$.
Set $V_R = \emptyset, V_L = \emptyset$.

For all $j \in V, j \ne i$:
    With probability $x_{ji}$
        Add $j$ to $V_L$.
    Else (With probability $x_{ij} = 1 - x_{ji}$)
        Add $j$ to $V_R$.

Return order
  LP-KwikSort$(V_L, x)$, $i$, LP-KwikSort$(V_R, x)$
```

LP-KwikCluster(V, x^+, x^-)

A recursive algorithm for rounding the LP for weighted Correlation-Clustering*. Given an LP solution $x^+ = \{x^+_{ij}\}_{i<j}$, $x^- = \{x^-_{ij}\}_{i<j}$, returns a clustering of the vertices*

```
If $V = \emptyset$ then return $\emptyset$
Pick random pivot $i \in V$.
Set $C = \{i\}, V' = \emptyset$.

For all $j \in V, j \ne i$:
    With probability $x^+_{ij}$
        Add $j$ to $C$.
    Else (With probability $x^-_{ij} = 1 - x^+_{ij}$)
        Add $j$ to $V'$.

Return clustering
  $\{C\} \cup$ LP-KwikCluster$(V', x^+, x^-)$.
```

Figure 1: Standard LP relaxations and their corresponding rounding algorithms.

- 5/2 *when the weights satisfy the probability constraints,*
- 2 *when the weights satisfy the probability and the triangle inequality constraints, and*
- 4/3 *for* Consensus-Clustering.

The result for Consensus-Clustering *is obtained by returning the better of* LP-KwikCluster *and* Pick-A-Cluster.

The bounds in Theorems 14 and 15 are obtained with respect to the optimal corresponding LP solution, and hence imply bounds on their integrality gaps. We further remark that the integrality gap of the Fas-Tournament LP can be lower bounded by 3/2. This follows from the fact that for any tournament on n vertices, there is a feasible solution to the Fas-Tournament LP that has value at most $m/3$ and there exist tournaments with no minimum feedback arc set of size smaller than $m(1/2 - \varepsilon)$, where ε is arbitrarily small, and where m is the number of edges.

We now prove Theorems 14 and 15. The common technique will be to reduce the problem to proving global bounds of certain multinomials in high dimensional polytopes. We start with the analysis of LP-KwikSort (Theorem 14).

Let C^{LKS} denote the cost of the ordering returned by the rounding algorithm LP-KwikSort. We divide all pairs i, j into those that are charged *dangerously* and those that are charged *safely* by the algorithm. The safe edges are charged when one of their endpoints is chosen as pivot, and the other endpoint is in the same recursive call. The expected cost of pairs that are charged safely in LP-KwikSort is

$$x_{ij}w_{ji} + x_{ji}w_{ij} \ , \tag{6}$$

which is exactly the contribution to the LP solution. We let c^*_{ij} denote expression (6). So the value of the LP solution is $C_{LP} = \sum_{i<j} c^*_{ij}$.

A pair i, j is charged dangerously when a third vertex k is chosen as pivot, all three i, j, k are in the same recursive call, and i, j are placed on opposite sides of k. The charge is w_{ij} (respectively, w_{ji}) if j (respectively, i) is placed on the left side of k and i (respectively, j) on its right. In either case, we charge this cost to the triplet i, j, k. We let T denote the set of all triplets of distinct vertices, and for any $t = \{i, j, k\} \in T$ we denote by A_t the event that all of i, j, k are in the same recursive call when the first one among them is chosen as pivot. Let p_t denote the probability of A_t. Let B^t_{ij} denote the event that (i, j) is dangerously charged to triangle t, in that order (i to the left, j to the right). Then we have for any $t = \{i, j, k\}$,

$$\Pr\left[A_t \wedge B^t_{ij}\right] = \Pr\left[A_t\right] \Pr\left[B^t_{ij} | A_t\right] = \frac{1}{3} p_t x_{ik} x_{kj} .$$

(The 1/3 comes from the fact that conditioned on A_t, each one of i, j, k was equally likely to be the pivot vertex.) Denote $q^t_{ij} = \frac{1}{3} x_{ik} x_{kj}$. So the total expected charge to a triplet $t = \{i, j, k\}$ is $p_t y(t)$, where

$$y(t) = q^t_{ij} w_{ji} + q^t_{ji} w_{ij} + q^t_{jk} w_{kj} + q^t_{kj} w_{jk} + q^t_{ki} w_{ik} + q^t_{ik} w_{ki} .$$

Now we notice that for any $t = \{i, j, k\}$ and $t' = \{i, j, k'\}$ (two triplets sharing a pair i, j), the events $A_t \wedge (B^t_{ij} \vee B^t_{ji})$ and $A_{t'} \wedge (B^{t'}_{ij} \vee B^{t'}_{ji})$ are disjoint, because a pair i, j can be split into two different recursion branches only once. Thus,

$$\sum_{t: i,j \in t} p_t (q^t_{ij} + q^t_{ji}) \leq 1 \ .$$

The above expression is exactly the probability that the pair i, j is dangerously charged. Therefore, the total expected cost of LP-KwikSort is $E\left[C^{LKS}\right] = B^{LKS} + F^{LKS}$, where

$$B^{LKS} = \sum_t p_t y(t)$$

$$F^{LKS} = \sum_{i<j} \left(1 - \sum_{t:i,j\in t} p_t(q^t_{ij} + q^t_{ji})\right) c^*_{ij} \ .$$

The following expression is a rearrangement of the sum $C_{LP} = \sum_{i<j} c^*_{ij}$: $C_{LP} = B_{LP} + F_{LP}$, where

$$B_{LP} = \sum_t p_t \sum_{\{i,j\}\subseteq t} (q^t_{ij} + q^t_{ji}) c^*_{ij}$$

$$F_{LP} = \sum_{i<j} \left(1 - \sum_{t:i,j\in t} p_t(q^t_{ij} + q^t_{ji})\right) c^*_{ij} \ .$$

So $F_{LP} = F^{LKS} \geq 0$. We have the following lemma. We defer the proof to Section 8.

Lemma 16 *If the weight system satisfies the probability constraints (respectively, probability constraints and triangle inequality constraints), then for any $t \in T$,*

$$y(t) \leq \tau \sum_{\{i,j\}\subseteq t} (q^t_{ij} + q^t_{ji}) c^*_{ij},$$

where $\tau = 5/2$ (respectively, $\tau = 2$).

Therefore, in this case, $B^{LKS} \leq \tau B_{LP}$. We conclude that $E\left[C^{LKS}\right] \leq \tau C_{LP}$. This concludes the proof of the first two items of Theorem 14.

We now prove the last item of Theorem 14 by coupling LP-KwikSort with Pick-A-Perm. Recall from Section 5 that the expected value of the Pick-A-Perm algorithm is

$$E\left[C^{PAP}\right] = \sum_{i<j} z_{ij},$$

where $z_{ij} = 2w_{ij}(1 - w_{ij})$. We rearrange this sum as follows:

$$E\left[C^{PAP}\right] = B^{PAP}_{LP} + F^{PAP}_{LP},$$

where

$$B^{PAP}_{LP} = \sum_t p_t \sum_{\{i,j\}\subseteq t} (q^t_{ij} + q^t_{ji}) z_{ij}$$

$$F^{PAP}_{LP} = \sum_{i<j} \left(1 - \sum_{t:\{i,j\}\subseteq t} p_t(q^t_{ij} + q^t_{ji})\right) z_{ij} \ .$$

It is easy to see that $0 \leq F^{PAP}_{LP} \leq 2F_{LP}$ (because $z_{ij} \leq 2c^*_{ij}$, and $\sum_{t:i,j\in t} p_t(q^t_{ij} + q^t_{ji}) \leq 1$). Along with $F^{LKS} = F_{LP}$, this implies that $\frac{2}{3}F^{LKS} + \frac{1}{3}F^{PAP}_{LP} \leq \frac{4}{3}F_{LP}$. Likewise, in Lemma 17 (proof in Section 8), we bound a convex combination of B^{LKS} and B^{PAP}_{LP}.

Lemma 17 *For all $t = \{i, j, k\}$,*

$$\frac{2}{3}y(t) + \frac{1}{3}\sum_{\{i,j\}\subseteq t}\left(q^t_{ij} + q^t_{ji}\right) z_{ij} \le \frac{4}{3}\sum_{\{i,j\}\subseteq t}\left(q^t_{ij} + q^t_{ji}\right) c^*_{ij}.$$

As a consequence, $\frac{2}{3}B^{LKS} + \frac{1}{3}B^{PAP}_{LP} \le \frac{4}{3}B_{LP}$.

Combining, we conclude that

$$\frac{2}{3}E\left[C^{LKS}\right] + \frac{1}{3}E\left[C^{PAP}_{LP}\right] \le \frac{4}{3}C_{LP}\ .$$

This means, in particular, that the best of LP-KwikSort and Pick-A-Perm has an expected approximation ratio of at most $\frac{4}{3}$ with respect to the LP cost. This concludes the proof of Theorem 14.

We now prove Theorem 15, by analyzing the output of LP-KwikCluster on Correlation-Clustering and Consensus-Clustering instances. Define $c^*_{ij} = x^+_{ij}w^-_{ij} + x^-_{ij}w^+_{ij}$. This is the LP contribution as well as the expected charge of the safe pairs, which are defined as above: these are pairs of vertices $i \ne j$ such that one was chosen as pivot when the other was in the same recursive call to LP-KwikCluster. All other pairs are *dangerously charged*.

For a triplet $t = (i, j, k)$ of disjoint vertices, as usual, we let A_t denote the event that one of i, j, k was chosen as pivot when the other two vertices are in the same recursive call to LP-KwikCluster. Let $p_t = \Pr[A_t]$. Let $B^t_{\{i\}j}$ denote the event that i, j was dangerously charged to t, because k is the pivot, i is taken in k's cluster and j is placed aside (the charge to t is then w^+_{ij}). The probability of $B^t_{\{i\}j}$ conditioned on A_t is $q^t_{\{i\}j} = \frac{1}{3}x^+_{ki}x^-_{kj}$. Let $B^t_{\{ij\}}$ denote the event that i, j was dangerously charged to t, because k is the pivot, and both i and j are taken in k's cluster (the charge is w^-_{ij}). The probability of $B^t_{\{ij\}}$ conditioned on A_t is $q^t_{\{ij\}} = \frac{1}{3}x^+_{ki}x^+_{kj}$. Define $y(t) = \sum_{\{i,j\}\subseteq t}(q^t_{\{i\}j} + q^t_{\{j\}i})w^+_{ij} + q^t_{\{ij\}}w^-_{ij}$.

For all $i \ne j$,

$$\sum_{t:\{i,j\}\subseteq t} p_t(q^t_{\{i\}j} + q^t_{\{j\}i} + q^t_{\{ij\}}) \le 1$$

by disjointness of events. As before, we decompose $E\left[C^{LKS}\right] = B^{LKS} + F^{LKS}$ and $C_{LP} = F_{LP} + B_{LP}$, where

$$\begin{aligned}
B^{LKS} &= \sum_t p_t y(t) \\
F^{LKS} &= \sum_{i<j}\left(1 - \sum_{t:\{i,j\}\subseteq t} p_t(q^t_{\{i\}j} + q^t_{\{j\}i} + q^t_{\{ij\}})\right) c^*_{ij}. \\
B_{LP} &= \sum_t p_t \sum_{\{i,j\}\subseteq t}\left(q^t_{\{i\}j} + q^t_{\{j\}i} + q^t_{\{ij\}}\right) c^*_{ij} \\
F_{LP} &= \sum_{i<j}\left(1 - \sum_{t:\{i,j\}\subseteq t} p_t(q^t_{\{i\}j} + q^t_{\{j\}i} + q^t_{\{ij\}})\right) c^*_{ij}\ .
\end{aligned}$$

Lemma 18 *If the weight system satisfies the probability constraints (respectively, probability constraints and triangle inequality constraints), then for any $t \in T$,*

$$y(t) \le \tau \sum_{\{i,j\}\subseteq t} \left(q^t_{\{i\}j} + q^t_{\{j\}i} + q^t_{\{ji\}} \right) c^*_{ij} \ ,$$

where $\tau = 5/2$ (respectively, $\tau = 2$).

As a result, we get a 5/2 approximation for the probability constraints case, and a 2 approximation for the probability and triangle inequality constraints case. This proves the first two items of Theorem 15.

For Consensus-Clustering, we let C^{PAC}_{LP} denote the value of Pick-A-Cluster. So $E\left[C^{PAC}_{LP}\right] = B^{PAC}_{LP} + F^{PAC}_{LP}$, where

$$\begin{aligned}
B^{PAC}_{LP} &= \sum_t p_t \sum_{\{i,j\}\subseteq t} \left(q^t_{\{i\}j} + q^t_{\{j\}i} + q^t_{\{ij\}} \right) z_{ij} \\
F^{PAC}_{LP} &= \sum_{i<j} \left(1 - \sum_{t:\{i,j\}\subseteq t} p_t(q^t_{\{i\}j} + q^t_{\{j\}i} + q^t_{\{ij\}}) \right) z_{ij} \ge 0. \\
z_{ij} &= 2w^+_{ij} w^-_{ij}
\end{aligned}$$

Lemma 19 *For all $t = \{i,j,k\}$,*

$$\frac{2}{3} y(t) + \frac{1}{3} \sum_{\{i,j\}\subseteq t} \left(q^t_{\{i\}j} + q^t_{\{j\}i} + q^t_{\{ij\}} \right) z_{ij} \le \frac{4}{3} \sum_{\{i,j\}\subseteq t} \left(q^t_{\{i\}j} + q^t_{\{j\}i} + q^t_{\{ij\}} \right) c^*_{ij} \ .$$

Also, it is easy to see that $z_{ij} \le 2c^*_{ij}$, so $0 \le F^{PAC}_{LP} \le 2F_{LP}$ and consequently $\frac{2}{3}F^{LKS} + \frac{1}{3}F^{PAC}_{LP} \le \frac{4}{3}F_{LP}$. Combining this with Lemma 19, we conclude that $\frac{2}{3}C^{LKS} + \frac{1}{3}C^{PAC}_{LP} \le \frac{4}{3}F_{LP}$, as desired. The proofs of Lemmas 18 and 19 can be found in Section 8. This completes the proof of Theorem 15. □

8 Proving Polynomial Inequalities in Polytopes

In this section we prove Lemmas 16,17,18 and 19. All these lemmas are equivalent to proving certain inequalities on polynomials in $\mathbb{R}^6$. We restate these inequalities for the sake of clarity, and slightly change notation to reduce indexing. In what follows we fix a triplet t consisting of three arbitrary vertices, $t = (1,2,3) \subseteq V$. For the ranking proofs (Lemmas 16 and 17) we let $x_1 = x_{23}, x_2 = x_{31}, x_3 = x_{12}$ and $w_1 = w_{23}, w_2 = w_{31}, w_3 = w_{12}$. For the clustering proofs (Lemmas 18 and 19), we let $x_1 = x^-_{23}, x_2 = x^-_{31}, x_3 = x^-_{12}$ and $w_1 = w^-_{23}, w_2 = w^-_{31}, w_3 = w^-_{12}$. We use $\mathbf{x} \in \mathbb{R}^3$ as shorthand for (x_1, x_2, x_3) and $\mathbf{w} \in \mathbb{R}^3$ as shorthand for (w_1, w_2, w_3). We will use the same symbols to denote parallel objects in the ranking (Lemmas 16 and 17) and clustering (Lemmas 18 and 19) proofs. To avoid confusion, we now separate between the two.

8.1 Polyhedral Inequalities for Ranking

Let $\Pi \subseteq \mathbb{R}^3$ denote the *probability constraints* polytope, that is,

$$\Pi = \{(a_1, a_2, a_3) \ : \ 0 \le a_i \le 1, \ i = 1,2,3\} \ . \tag{7}$$

Let $\Delta \subseteq \Pi$ denote the *triangle inequality and probability constraints* for ranking polytope, that is

$$\Delta = \{(a_1, a_2, a_3) \in \Pi \ : \ 1 \le a_1 + a_2 + a_3 \le 2\} \ .$$

We define three functions, $piv, pap, lp : \mathbb{R}^6 \to \mathbb{R}$, as follows:

$$\begin{aligned}
piv(\mathbf{x}, \mathbf{w}) &= x_1x_2w_3 + (1-x_1)(1-x_2)(1-w_3) \\
&\quad + x_2x_3w_1 + (1-x_2)(1-x_3)(1-w_1) \\
&\quad + x_3x_1w_2 + (1-x_3)(1-x_1)(1-w_2) \\
\\
pap(\mathbf{x}, \mathbf{w}) &= (x_1x_2 + (1-x_1)(1-x_2))2w_3(1-w_3) \\
&\quad + (x_2x_3 + (1-x_2)(1-x_3))2w_1(1-w_1) \qquad (8) \\
&\quad + (x_3x_1 + (1-x_3)(1-x_1))2w_2(1-w_2) \\
\\
lp(\mathbf{x}, \mathbf{w}) &= (x_1x_2 + (1-x_1)(1-x_2))(x_3(1-w_3) + (1-x_3)w_3) \\
&\quad + (x_2x_3 + (1-x_2)(1-x_3))(x_1(1-w_1) + (1-x_1)w_1) \\
&\quad + (x_3x_1 + (1-x_3)(1-x_1))(x_2(1-w_2) + (1-x_2)w_2)
\end{aligned}$$

Lemma 16 is equivalent to showing that $f = piv - \frac{5}{2}lp \leq 0$ for all $(\mathbf{x}, \mathbf{w}) \in \Delta \times \Pi$ and $g = piv - 2lp \leq 0$ for all $(\mathbf{x}, \mathbf{w}) \in \Delta \times \Delta$. We make two simplification steps.

1. *Linearity in* $\mathbf{w}$: The functions f and g are linear in $\mathbf{w}$ (for $\mathbf{x}$ fixed). Therefore f obtains its maximum on $(\mathbf{x}, \mathbf{w})$ for $\mathbf{w}$ which is some vertex of Π, and similarly g obtains its maximum value on $(\mathbf{x}, \mathbf{w})$ for $\mathbf{w}$ which is some vertex of Δ. For f it suffices to check $\mathbf{w} = (0,0,0)$ and $\mathbf{w} = (0,0,1)$ (due to symmetry), and for g it suffices to check $\mathbf{w} = (0,0,1)$. Let $\tilde{f}(\mathbf{x}) = f(\mathbf{x}, 0, 0, 0)$, $\hat{f}(\mathbf{x}) = f(\mathbf{x}, 0, 0, 1)$ and $\tilde{g}(\mathbf{x}) = g(\mathbf{x}, 0, 0, 1)$. It remains to show that $\tilde{f}, \hat{f}, \tilde{g} : \mathbb{R}^3 \to \mathbb{R}$ are bounded above by 0 on Δ.

2. *Trilinearity in* $\mathbf{x}$: For $i = 1, 2, 3$ the functions $\tilde{f}, \hat{f}$ and $\tilde{g}$ are linear in x_i when x_j's are fixed for $j \in \{1,2,3\} \setminus \{i\}$. This means that any point $\mathbf{x} \in \Delta$ such that $\mathbf{x} + t\mathbf{e}_i \in \Delta$ for all $t \in [-\varepsilon, \varepsilon]$ for some $\varepsilon > 0$ and some $i \in \{1,2,3\}$ (where $\mathbf{e}_i$ is a standard basis element of $\mathbb{R}^3$) is *not* a strict local maximum of $\tilde{f}, \hat{f}$ and $\tilde{g}$ in Δ, so these points $\mathbf{x}$ can be ignored. The points that are left are $\mathbf{x} \in \Delta$ s.t. that $x_1 + x_2 + x_3 = 1$ or $x_1 + x_2 + x_3 = 2$.

Let $H_k \subseteq \mathbb{R}^3$ denote the hyperplane $x_1 + x_2 + x_3 = k$ for $k = 1, 2$, and let $\Delta_k = \Delta \cap H_k$. The closed polytopes Δ_k are two dimensional and the polynomials $\tilde{f}, \hat{f}$ and $\tilde{g}$ are of total degree 3 and maximal degree 2 in each variable. It is tedious yet elementary to verify that the maxima are obtained in accordance with Table 2.

function \ domain	Δ_1	Δ_2
$\tilde{f}$	0 at $(1/2, 0, 1/2)$	0 at $(1,0,1)$
$\hat{f}$	0 at $(0,0,1)$	0 at $(1,0,1)$
$\tilde{g}$	0 at $(0,0,1)$	0 at $(1,0,1)$

Table 2: Maxima of $\tilde{f}, \hat{f}$ and $\tilde{g}$ on Δ_1, Δ_2.

Lemma 17 is equivalent to proving that $h = 2piv/3 + pap/3 - 4lp/3 \leq 0$ for all $(\mathbf{x}, \mathbf{w}) \in \Delta \times \Delta$. The trilinearity in $\mathbf{x}$ still holds true for h, so as before we can assume that either $\mathbf{x} \in \Delta_1$ or $\mathbf{x} \in \Delta_2$. We can assume without loss of generality (by symmetry) that $x \in \Delta_2$, that is, $x_1 + x_2 + x_3 = 2$. When $\mathbf{x}$ is fixed, then h is a (possibly degenerate) concave paraboloid in $\mathbf{w}$. In case of non-degeneracy, its unique global maximum is obtained when $\nabla_w h = 0$, which can be easily verified to be solved by $\mathbf{w} = \mathbf{w}^* = (w_1^*, w_2^*, w_3^*)$ defined by

$$\begin{aligned} w_1^* &= \frac{x_2x_3}{x_2x_3 + (1-x_2)(1-x_3)} + 2x_1 - 1 \\ w_2^* &= \frac{x_3x_1}{x_3x_1 + (1-x_3)(1-x_1)} + 2x_2 - 1 \\ w_3^* &= \frac{x_1x_2}{x_1x_2 + (1-x_1)(1-x_2)} + 2x_3 - 1 \end{aligned} \tag{9}$$

(the paraboloid in $\mathbf{w}$ is degenerate if and only if any of the denominators in (9) are 0, equivalently $x_i = 0$ and x_j=1 for some i, j. But this implies that after possibly permuting coordinates, $\mathbf{x} = (1, 1, 0)$. But $h(1, 1, 0, \mathbf{w}) = -2w_3^2/3 \le 0$, proving the desired assertion trivially). Since we are assuming $x_1 + x_2 + x_3 = 2$, we have that for any $1 \le i < j \le 3$, $x_i + x_j \ge 1$, equivalently $x_ix_j \ge (1 - x_i)(1 - x_j)$. Therefore (9) implies $w_i \ge \frac{1}{2} + 2x_i - 1$ for $i = 1, 2, 3$. Summing up, we obtain $w_1 + w_2 + w_3 \ge -\frac{3}{2} + 2(x_1 + x_2 + x_3) = \frac{5}{2} > 2$. In other words, (9) implies that $\mathbf{w}^*$ and Δ are strictly on different sides of H_2. Let $\mathbf{w}' = (w_1', w_2', w_3')$ be any point in Δ. Consider the straight line ℓ passing through $\mathbf{w}'$ and $\mathbf{w}^*$, and let $\mathbf{w}''$ the intersection of this line with H_2. Restricted to ℓ (and for our fixed $\mathbf{x} \in \Delta_2$) h is a parabola, attaining its maximum on $\mathbf{w}^*$. Therefore $h(\mathbf{x}, \mathbf{w}'') \ge h(\mathbf{x}, w')$, and we can assume in what follows that $\mathbf{w} = \mathbf{w}'' \in H_2$ (we must drop the assumption that $\mathbf{w} \in \Delta$ though). We change variables and let $\tilde{h} : \mathbb{R}^4 \to \mathbb{R}$ be defined by $\tilde{h}(x_1, x_2, w_1, w_2) = h(x_1, x_2, 2 - x_1 - x_2, w_1, w_2, 2 - w_1 - w_2)$. We reduced the problem to proving that $\tilde{h} \le 0$ on $\{x_1 \le 1,\ x_2 \le 1,\ x_1 + x_2 \ge 1\} \times \mathbb{R}^2$. It is elementary to verify, using vanishing derivatives, that for (x_1, x_2) fixed, the maximum of $\tilde{h}$ is obtained when $(w_1, w_2) = (x_1, x_2)$. Substituting, we get $\tilde{h}(x_1, x_2, x_1, x_2) = -2(-1+x_1)(-1+x_2)(-1+x_2+x_3)$ which is less than or equal to 0 because $x_1 + x_2 \ge 1$ and $x_1, x_2 \le 1$.

8.2 Polyhedral Inequalities for Clustering

Let $\Pi \subseteq \mathbb{R}^3$ denote the *probability constraints* polytope as defined in (7). Let $\Delta \subseteq \Pi$ denote the *triangle inequality and probability constraints* polytope for clustering, that is,

$$\Delta = \{(a_1, a_2, a_3) \in \Pi \ :\ a_3 \le a_1 + a_2,\ a_1 \le a_2 + a_3,\ a_2 \le a_3 + a_1\} \ .$$

We define three functions, $piv, pap, lp : \mathbb{R}^6 \to \mathbb{R}$, as follows:

$$\begin{aligned} piv(\mathbf{x}, \mathbf{w}) &= (1-x_1)(1-x_2)w_3 + (x_1(1-x_2) + (1-x_1)x_2)(1-w_3) \\ &\quad + (1-x_2)(1-x_3)w_1 + (x_2(1-x_3) + (1-x_2)x_3)(1-w_1) \\ &\quad + (1-x_3)(1-x_1)w_2 + (x_3(1-x_1) + (1-x_3)x_1)(1-w_2) \\ \\ pap(\mathbf{x}, \mathbf{w}) &= ((1-x_1)(1-x_2) + (1-x_1)x_2 + x_1(1-x_2))2w_3(1-w_3) \\ &\quad + ((1-x_2)(1-x_3) + (1-x_2)x_3 + x_2(1-x_3))2w_1(1-w_1) \\ &\quad + ((1-x_3)(1-x_1) + (1-x_3)x_1 + x_3(1-x_1))2w_2(1-w_2) \\ \\ lp(\mathbf{x}, \mathbf{w}) &= ((1-x_1)(1-x_2) + (1-x_1)x_2 + x_1(1-x_2))(x_3(1-w_3) + (1-x_3)w_3) \\ &\quad + ((1-x_2)(1-x_3) + (1-x_2)x_3 + x_2(1-x_3))(x_1(1-w_1) + (1-x_1)w_1) \\ &\quad + ((1-x_3)(1-x_1) + (1-x_3)x_1 + x_3(1-x_1))(x_2(1-w_2) + (1-x_2)w_2) \end{aligned} \tag{10}$$

Lemma 18 is equivalent to showing that $f = piv - \frac{5}{2}lp \le 0$ for all $(\mathbf{x}, \mathbf{w}) \in \Delta \times \Pi$ and $g = piv - 2lp \le 0$ for all $(\mathbf{x}, \mathbf{w}) \in \Delta \times \Delta$. We make the two simplification steps as before.

1. *Linearity in* $\mathbf{w}$: The functions f and g are linear in $\mathbf{w}$ (for $\mathbf{x}$ fixed). Arguing as before, it suffices to analyze f on $\mathbf{w} = (0,0,0)$, $\mathbf{w} = (0,0,1)$, $\mathbf{w} = (0,1,1)$ and $\mathbf{w} = (1,1,1)$, and g on $\mathbf{w} = (0,0,0), \mathbf{w} = (0,1,1), \mathbf{w} = (1,1,1)$. We denote the functions on $\mathbf{x}$ after substituting for $\mathbf{w}$ by $f_{000}, f_{001}, f_{011}, f_{111}$ and $g_{000}, g_{011}, g_{111}$ (with obvious correspondence).

2. *Trilinearity in* $\mathbf{x}$*:* For $i = 1,2,3$ the functions f and g are linear in x_i when x_j's are fixed for $j \in \{1,2,3\} \setminus \{i\}$. This means that any point $\mathbf{x} \in \Delta$ such that $\mathbf{x} + t\mathbf{e}_i \in \Delta$ for all $t \in [-\varepsilon, \varepsilon]$ for some $\varepsilon > 0$ and some $i \in \{1,2,3\}$ (where $\mathbf{e}_i$ is a standard basis element of $\mathbb{R}^3$) is *not* a strict local maximum of f, g in Δ, so these points $\mathbf{x}$ can be ignored. The points that are left are $\mathbf{x} \in \Delta_1 \cup \Delta_2 \cup \Delta_3$ where $\Delta_i = \Delta \cap H_i$ for $i = 1,2,3$ and $H_1 = \{(a_1,a_2,a_3) \in \mathbb{R}^3 : \ a_1 = a_2 + a_3\}$, $H_2 = \{(a_1,a_2,a_3) \in \mathbb{R}^3 : \ a_2 = a_3 + a_1\}$, $H_3 = \{(a_1,a_2,a_3) \in \mathbb{R}^3 : \ a_3 = a_1 + a_2\}$.

The functions f, g restricted to one of the finitely many "interesting" points $\mathbf{w}$ and to $\mathbf{x} \in \Delta_i$ for some $i \in \{1,2,3\}$ can be represented as polynomials of total degree 3 and maximal degree 2 in each variable. Δ_k are two dimensional and the polynomials $\tilde{f}, \hat{f}$ and $\tilde{g}$ are of total degree 3 and maximal degree 2 in each variable. It is tedious yet elementary to verify that the maxima are obtained in accordance with Table 3.

function \ domain	Δ_1	Δ_2	Δ_3
f_{000}	0 at $(0,0,0)$	0 at $(0,0,0)$	0 at $(0,0,0)$
f_{001}	0 at $(1/2,0,1/2)$	-1 at $(0,1/2,1/2)$	0 at $(1/2,1/2,1)$
f_{011}	$-3/2$ at $(1/2,0,1/2)$	0 at $(0,1,1)$	0 at $(0,1,1)$
f_{111}	0 at $(1,0,1)$	0 at $(1,1,0)$	0 at $(1,0,1)$
g_{000}	0 at $(0,0,0)$	0 at $(0,0,0)$	0 at $(0,0,0)$
g_{011}	-1 at $(1,0,1)$	0 at $(0,1,1)$	0 at $(0,1,1)$
g_{111}	0 at $(1,0,1)$	0 at $(1,1,0)$	0 at $(1,0,1)$

Table 3: Maximum of $f_{000}, f_{001}, f_{011}, f_{111}, g_{000}, g_{011}, g_{111}$ on $\Delta_1, \Delta_2, \Delta_3$.

Lemma 19 is equivalent to proving that $h = 2piv/3 + pap/3 - 4lp/3 \leq 0$ for all $(\mathbf{x}, \mathbf{w}) \in \Delta \times \Delta$. We prove this assertion as follows:

Using symmetries of h: Let $(\mathbf{x}, \mathbf{w})$ be some local maximum of h in $\Delta \times \Delta$. Assume there is an index $i \in \{1,2,3\}$ such that all of $x_i, x_{i+1}, w_i, w_{i+1} \notin \{0,1\}$ (the index arithmetic is modulo 3). Without loss of generality, assume that $x_1, x_2, w_1, w_2 \notin \{0,1\}$. Since $(\mathbf{x}, \mathbf{w})$ is a local maximum of h on $\Delta \times \Delta$, and since $x_1, x_2, w_1, w_2 \notin \{0,1\}$, the derivatives of h on the hyperplane $H = \{(\mathbf{x}, \mathbf{w}) + t(1,-1,0,0,0,0) + s(0,0,0,1,-1,0) | t, s \in \mathbb{R}\}$ must vanish at $t = s = 0$. One verifies that h is a polynomial of total degree 2 in t, s on H, and the derivatives vanish in the unique point $t = (x_2 - x_1)/2, s = (w_2 - w_1)/2$. Therefore, we may assume that $x_1 = x_2$ and $w_1 = w_2$. Now if in addition $x_3, w_3 \notin \{0,1\}$ then we use the same argument (switching the roles of the variables), and we can assume that $x_1 = x_2 = x_3, w_1 = w_2 = w_3$. It is trivial to show that $h \leq 0$ under this constraint.

Boundary cases: We can now assume that either: (1) at least two of $x_1, x_2, x_3, w_1, w_2, w_3$ are in $\{0,1\}$, or, (2) $x_1 = x_2, w_1 = w_2$ and at least one of x_3, w_3 are in $\{0,1\}$. In addition, the function h is trilinear in $\mathbf{x}$, so we may assume (as above) that $\mathbf{x} \in \Delta_1 \cup \Delta_2 \cup \Delta_3$. This reduces the problem to proving inequalities for polynomials of total degree at most 4 and maximal degree at most 3 (respectively, 2) in each x-variable (respectively, w-variable), in 3-dimensional polytopes. We summarize the analysis in Table 4.

constraint	Δ_1	Δ_2	Δ_3	constraint	Δ_1	Δ_2	Δ_3
00****	0:000000	0:000000	0:000000	01****	infeasible	$0{:}011\frac{1}{2}11$	$0{:}011\frac{1}{2}11$
11****	$0{:}11011\frac{1}{2}$	$0{:}11011\frac{1}{2}$	infeasible	***00*	0:000000	0:000000	0:000000
***01*	$-\frac{2}{3}{:}101011$	0:011011	0:011011	***11*	0:101111	$0{:}11011\frac{1}{2}$	0:101111
0**0**	0:000000	0:000000	0:000000	0**1**	$-\frac{5}{4}{:}0001\frac{1}{4}\frac{3}{4}$	0:011111	0:011111
1**0**	$-\frac{2}{3}{:}110011$	$-\frac{2}{3}{:}110011$	$-\frac{2}{3}{:}101011$	1**1**	$0{:}1\frac{59}{64}\frac{5}{64}1\frac{59}{64}\frac{5}{64}$	$0{:}11011\frac{1}{2}$	$0{:}1011\frac{1}{2}1$
0***0*	0:000000	0:000000	0:000000	0***1*	$-\frac{5}{3}{:}000\frac{1}{2}1\frac{1}{2}$	$0{:}011\frac{1}{2}11$	$0{:}011\frac{1}{2}11$
1***0*	0:101101	$-\frac{2}{3}{:}110101$	0:101101	1***1*	$0{:}11011\frac{1}{2}$	$0{:}11011\frac{1}{2}$	0:101111
$xx0ww^*$	0:000000	0:000000	0:000000	$xx1ww^*$	infeasible	infeasible	$0{:}\frac{1}{2}\frac{1}{2}1\frac{1}{2}\frac{1}{2}1$
xx^*ww0	0:000000	0:000000	0:000000	xx^*ww1	0:110111	0:110111	$0{:}\frac{1}{2}\frac{1}{2}1\frac{1}{2}\frac{1}{2}1$

Table 4: Maxima of f given different constraints on Δ_1, Δ_2, Δ_3. The constraint 0**0*1 means, as an example, $x1 = 0, w1 = 0, w3 = 1$. A constraint of the form $xx0ww^*$ means $x_1 = x_2, x_3 = 0, w_1 = w_2$. The maxima are denoted by $M{:}x_1x_2x_3w_1w_2w_3$, where M is the maximum value, attained at $(x_1, x_2, x_3, w_1, w_2, w_3)$.

9 NP-Hardness of Feedback Arc Set on Tournaments

All the problems referred to in Table 1 in Section 1.2 were previously known to be NP-hard except for Fas-Tournament. In this section we show:

Theorem 20 *Unless NP ⊆ BPP,* Fas-Tournament *has no polynomial time algorithm.*

Proof: We reduce to Fas-Tournament from Fas-DiGraph, which is the problem of finding a minimum feedback arc set in a general directed graph. Fas-DiGraph is NP-hard [Kar72] (in fact, it is MAX-SNP-hard, see [Hås01, New00, NV01]).

Let $G = (V, A)$ (with $|V| = n$) be an instance of Fas-Digraph. Suppose we could add a set of edges A_R to G such that $(V, A \cup A_R)$ is a tournament, and such that exactly half of A_R are backward in any ordering π of V. Then by solving Fas-Tournament we would be able to recover the feedback arc set of G. This is generally impossible. However, if we add the edges A_R randomly (i.e. for every i, j such the neither (i, j) nor (j, i) are in A add (i, j) or (j, i) to A_R with equal probability) then for any π the expected number of backward edges is half $|R|$. The variance makes this approach fail. By blowing up G and using a concentration property of the random variable counting the number of backward edges in A_R, we can use this construction (see similar random digraph constructions in [New00, NV01]).

We pick an integer $k = \text{poly}(n)$ (chosen later). The blow-up digraph $G^k = (V^k, A^k)$ is defined as follows:

$$V^k = \bigcup_{v \in V} \{v_1, \ldots, v_k\}$$
$$A^k = \{(u_i, v_j) | (u, v) \in A, i, j \in \{1, \ldots, k\}\} .$$

We observe that the minimum feedback arc set of G^k is exactly k^2 times the minimum feedback arc set of G. Indeed, it suffices to consider only rankings π on V^k that rank the vertices $v_1, \ldots, v_k$ as one block for all $v \in V$ (as explained in [Alo06], if $v_i <_\pi v_j$ are not adjacent in the ranking, then either moving v_i immediately to the left of v_j or moving v_j immediately to the right of v_i will result in a ranking inducing no more feedback edges than π).

Now we turn G^k into a tournament $T^k = \{V^k, A^k \cup A_R^k\}$ using the construction defined above. For a ranking π of V^k, let $f_R(\pi)$ denote the number of feedback edges in A_R^k with respect to π. Denote by μ the expected value of $f_R(\pi)$, which is the same for all π, and can be efficiently computed. We claim that for $k = \text{poly}(n)$, with probability at least 2/3, all rankings π satisfy $|f_R(\pi) - \mu| = O((nk)^{3/2}\sqrt{\log(nk)})$. This would imply, using the above observation, that for big enough $k = \text{poly}(n)$ the size of the minimum feedback arc set of T^k can be used to efficiently recover the size of the minimum feedback arc set of G, because $(nk)^{3/2}\sqrt{\log(nk)} = o(k^2)$. To prove the claim, for any fixed ranking π, set a random indicator variable X_{wz}^π for every non-edge $\{w, z\}$ of G^k which equals 1 iff the edge between w and z in A_R^k is backward w.r.t. π. So $f_R(\pi) = \sum X_{wz}^\pi$. A simple application of Chernoff bounds [AS92] and union bound (over all possible $(nk)!$ rankings) completes the proof of the claim. It follows that unless FAS-DIGRAPH $\in BPP$, we cannot solve FAS-TOURNAMENT in polynomial time. □

We wish to thank Noga Alon for ideas significantly simplifying the proof [Alo06]. Our initial hardness result was via max-SNP hardness of FAS-DIGRAPH, and Noga Alon pointed out that the same idea also works with the weaker NP-hardness.

10 Related Work

Since the publication of the conference version of this work [ACN05], there have been interesting developments in the field.

On the ranking side, Kenyon-Mathieu and Schudy [KMS07] present a PTAS for FAS-TOURNAMENT, thus considerably improving the constant approximation guarantee presented here. Williamson and Van Zuylen [vZW07] derandomized the pivot algorithms introduced in this paper for both ranking and clustering, with matching approximation guarantees. In addition, Coppersmith et al. [CFR06] showed that ordering a weighted tournament by in-degree is a 5-approximation for weighted FAS-TOURNAMENT with probability constraints, thus obtaining another natural constant factor approximation. Ailon [Ail07] extends this work to partial rankings, often found in information science applications. In the machine learning community, the problem of learning how to rank has been revisited in the context of reduction to binary preference learning. We refer the reader to a recent paper by Ailon and Mohri [AM08] which is inspired by this work, and improves a result by Balcan et al. [BBB$^+$07] (inspired by [CFR06]).

On the clustering side, Ailon and Charikar [AC05] extend results here to hierarchical clulstering, a problem well studied in phylogeny. They generalize KWIKCLUSTER to that setting and obtain constant factor approximation guarantees.

11 Open Problems

- KWIKSORT is in fact the well-known *quick-sort* algorithm for ordered data with transitivity violations. Can we use other standard sorting algorithms, such as *merge-sort* to obtain similar approximation algorithms?

- Finding tight examples for the algorithms presented in this work is an interesting problem. For weighted weighted FAS-TOURNAMENT and weighted CORRELATION-CLUSTERING with probability constraints, Warren Schudy communicated the following tight example for the KWIKSORT and KWIKCLUSTER, respectively. It suffices to consider unweighted instances (weights are $0, 1$). For the ranking problem, take an acyclic tournament and flip the edge connecting the lowest and the highest ranked vertices. The optimal solution pays 1. KWIKSORT pays $n-2$ if the lowest or highest ranked vertices are chosen as pivot in the first step, otherwise 1. Therefore, the expected ratio is $3(n-2)/n$, which tends to 3 as $n \to \infty$. For the clustering problem set all edges to "+" except for one which is set to "−". The optimal solution pays 1 by clustering all the vertices together. KwikCluster pays $n-2$ if one of the two vertices incident to the unique "−"-edge is chosen as pivot in the first step, otherwise the optimal cost of 1, giving an expected ratio of $3(n-2)/n$. Finding tight examples for the triangle inequality cases as well as for the aggregation problems remains an open problem.

- Is Rank-Aggregation NP-Hard for 3 permutations [DKNS01a, DKNS01b]?
- Is Consensus-Clustering NP-Hard for a constant number of clusters [Wak98, FS03]?
- Can we approximate weighted Correlation-Clustering with triangle inequalities, but no probability constraints?

12 Acknowledgements

We would like to thank Ravi Kumar and D. Sivakumar for several discussions on these problems. Thanks also to Shuchi Chawla and Tony Wirth for extensive discussions on consensus clustering, to Aristides Gionis for sending us a preprint of their paper [GMT05], to Noga Alon for discussions on the hardness result, to Warren Schudy for communicating his nice observations to us, to Anke van Zuylen and Lisa Fleischer for sending us preprints of their recent derandomization results and to the anonymous reviewers.

References

[AC05] Nir Ailon and Moses Charikar. Fitting tree metrics: Hierarchical clustering and phylogeny. In *FOCS*, pages 73–82. IEEE Computer Society, 2005.

[ACN05] Nir Ailon, Moses Charikar, and Alantha Newman. Aggregating inconsistent information: ranking and clustering. In Harold N. Gabow and Ronald Fagin, editors, *STOC*, pages 684–693. ACM, 2005.

[AFK96] Sanjeev Arora, Alan M. Frieze, and Haim Kaplan. A new rounding procedure for the assignment problem with applications to dense graph arrangement problems. In *FOCS*, pages 21–30, 1996.

[Ail07] Nir Ailon. Aggregation of partial rankings, -ratings and top- lists. In Nikhil Bansal, Kirk Pruhs, and Clifford Stein, editors, *SODA*, pages 415–424. SIAM, 2007.

[Alo06] Noga Alon. Ranking tournaments. *SIAM J. Discrete Math.*, 20(1):137–142, 2006.

[AM08] Nir Ailon and Mehryar Mohri. An efficient reduction of ranking to classification. In Rocco A. Servedio and Tong Zhang, editors, *COLT*, pages 87–98. Omnipress, 2008.

[AS92] N. Alon and J. H. Spencer. *The Probabilistic Method.* Wiley, New York, 1992.

[BBB$^+$07] Maria-Florina Balcan, Nikhil Bansal, Alina Beygelzimer, Don Coppersmith, John Langford, and Gregory B. Sorkin. Robust reductions from ranking to classification. In Nader H. Bshouty and Claudio Gentile, editors, *COLT*, volume 4539 of *Lecture Notes in Computer Science*, pages 604–619. Springer, 2007.

[BBC04] Nikhil Bansal, Avrim Blum, and Shuchi Chawla. Correlation clustering. *Machine Learning*, 56(1-3):89–113, 2004.

[BJT92] Jorgen Bang-Jensen and Carsten Thomassen. A polynomial algorithm for the 2-path problem in semicomplete graphs. *SIAM Journal of Discrete Mathematics*, 5(3):366–376, 1992.

[Bor81] Jean Charles Borda. Mémoire sur les élections au scrutin. *Histoire de l'Académie Royale des Sciences*, 1781.

[BTT89] J.J. Bartholdi, C. A. Tovey, and M.A. Trick. Voting schemes for which it can be difficult to tell who won the election. *Social Choice and Welfare*, 6(2):157–165, 1989.

[CCMR06] Kamalika Chaudhuri, Kevin Chen, Radu Mihaescu, and Satish Rao. On the tandem duplication-random loss model of genome rearrangement. In *SODA* [DBL06], pages 564–570.

[CDZ00] Mao-Cheng Cai, Xiaotie Deng, and Wenan Zang. An approximation algorithm for feedback vertex sets in tournaments. *SIAM J. Comput.*, 30(6):1993–2007, 2000.

[CFR06] Don Coppersmith, Lisa Fleischer, and Atri Rudra. Ordering by weighted number of wins gives a good ranking for weighted tournaments. In *SODA* [DBL06], pages 776–782.

[CGW03] Moses Charikar, Venkatesan Guruswami, and Anthony Wirth. Clustering with qualitative information. In *FOCS*, pages 524–533. IEEE Computer Society, 2003.

[Con85] M.-J. Condorcet. Éssai sur l'application de l'analyse à la probabilité des décisions rendues à la pluralité des voix. 1785.

[DBL06] *Proceedings of the Seventeenth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2006, Miami, Florida, USA, January 22-26, 2006*. ACM Press, 2006.

[DG77] Percy Diaconis and Ronald Graham. Spearman's footrule as a measure of disarray. *Journal of the Royal Statistical Society, Series B*, 39(2):262–268, 1977.

[DKNS01a] Cynthia Dwork, Ravi Kumar, Moni Naor, and D. Sivakumar. Rank aggregation methods for the web. In *WWW*, pages 613–622, 2001.

[DKNS01b] Cynthia Dwork, Ravi Kumar, Moni Naor, and D. Sivakumar. Rank aggregation revisited. 2001. Manuscript.

[DS02] Irit Dinur and Shmuel Safra. The importance of being biased. In *STOC*, pages 33–42, 2002.

[ENSS98] Guy Even, Joseph Naor, Baruch Schieber, and Madhu Sudan. Approximating minimum feedback sets and multicuts in directed graphs. *Algorithmica*, 20(2):151–174, 1998.

[FK99] Alan M. Frieze and Ravi Kannan. Quick approximation to matrices and applications. *Combinatorica*, 19(2):175–220, 1999.

[FKS03] Ronald Fagin, Ravi Kumar, and D. Sivakumar. Efficient similarity search and classification via rank aggregation. In Alon Y. Halevy, Zachary G. Ives, and AnHai Doan, editors, *SIGMOD Conference*, pages 301–312. ACM, 2003.

[FS03] Vladimir Filkov and Steven Skiena. Integrating microarray data by consensus clustering. In *ICTAI*, pages 418–425. IEEE Computer Society, 2003.

[GMT05] Aristides Gionis, Heikki Mannila, and Panayiotis Tsaparas. Clustering aggregation. In *ICDE*, pages 341–352. IEEE Computer Society, 2005.

[Hås01] Johan Håstad. Some optimal inapproximability results. *J. ACM*, 48(4):798–859, 2001.

[Kar72] R. M. Karp. Reducibility among combinatorial problems. In *Complexity of Computer Computations*, pages 85–104. Plenum Press, New York, 1972.

[Kem59] John G. Kemeny. Mathematics without numbers. *Daedalus*, 88:571–591, 1959.

[KMS07] Claire Kenyon-Mathieu and Warren Schudy. How to rank with few errors. In David S. Johnson and Uriel Feige, editors, *STOC*, pages 95–103. ACM, 2007.

[KS62] John G. Kemeny and J. Laurie Snell. *Mathematical Models in the Social Sciences*. Blaisdell, New York, 1962. Reprinted by MIT Press, Cambridge, 1972.

[New00] Alantha Newman. Approximating the maximum acyclic subgraph. Master's thesis, Massachusetts Institute of Technology, Cambridge, MA, June 2000.

[NV01] Alantha Newman and Santosh Vempala. Fences are futile: On relaxations for the linear ordering problem. In Karen Aardal and Bert Gerards, editors, *IPCO*, volume 2081 of *Lecture Notes in Computer Science*, pages 333–347. Springer, 2001.

[Pot80] C. N. Potts. An algorithm for the single machine sequencing problem with precedence constraints. *Mathematical Programming*, 13:78–87, 1980.

[Sey95] Paul D. Seymour. Packing directed circuits fractionally. *Combinatorica*, 15(2):281–288, 1995.

[Spe89] Ewald Speckenmeyer. On feedback problems in diagraphs. In Manfred Nagl, editor, *WG*, volume 411 of *Lecture Notes in Computer Science*, pages 218–231. Springer, 1989.

[Str02] A. Strehl. *Relationship-Based Clustering and Cluster Ensembles for High-Dimensional Data Mining*. PhD thesis, University of Texas at Austin, 2002.

[vZW07] Anke van Zuylen and David P. Williamson. Deterministic algorithms for rank aggregation and other ranking and clustering problems. In Christos Kaklamanis and Martin Skutella, editors, *WAOA*, volume 4927 of *Lecture Notes in Computer Science*, pages 260–273. Springer, 2007.

[Wak98] Yoshiko Wakabayashi. The complexity of computing medians of relations. *Resenhas*, 3(3):323–349, 1998.