

# Aggregating Inconsistent Information: Ranking and Clustering

NIR AILON

*Google Research, New York, NY*

MOSES CHARIKAR

*Princeton University, Princeton, NJ*

AND

ALANTHA NEWMAN

*DIMACS, Rutgers University, New Brunswick, NJ*

**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  $NP \subseteq BPP$ , there is no polynomial time algorithm for the problem of minimum feedback arc set in tournaments.

Categories and Subject Descriptors: F.2 [**Theory of Computation**]: Analysis of Algorithms and Problem Complexity; G.2 [**Mathematics of Computing**]: Discrete Mathematics

---

The work of N. Ailon was done while he was at Princeton University, partly supported by a Princeton University Honorable Fellowship.

M. Charikar was supported by a National Science Foundation (NSF) ITR grant CCR-0205594, DOE Early Career Principal Investigator award DE-FG02-02ER25540, NSF CAREER award CCR-0237113, and Alfred P. Sloan fellowship and a Howard B. Wentz Jr. Junior Faculty award.

The work of A. Newman was done while she was visiting Princeton University, supported by M. Charikar's Alfred P. Sloan fellowship.

Authors' addresses: N. Ailon, Google NY, 4th Floor, 76 Ninth Avenue, New York NY 10011, e-mail: nailon@gmail.com; M. Charikar, Department of Computer Science, Princeton University, 35 Olden Street, Princeton, NJ 08540, e-mail: moses@cs.princeton.edu; A. Newman, DIMACS Center, CoRE Building, 4th Floor, Rutgers University, 96 Frelinghuysen Road, Piscataway, NJ 08854, e-mail: alantha.newman@gmail.com.

Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or direct commercial advantage and that copies show this notice on the first page or initial screen of a display along with the full citation. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, to republish, to post on servers, to redistribute to lists, or to use any component of this work in other works requires prior specific permission and/or a fee. Permissions may be requested from Publications Dept., ACM, Inc., 2 Penn Plaza, Suite 701, New York, NY 10121-0701 USA, fax +1 (212) 869-0481, or [permissions@acm.org](mailto:permissions@acm.org).  
© 2008 ACM 0004-5411/2008/10-ART23 \$5.00 DOI 10.1145/1411509.1411513 <http://doi.acm.org/10.1145/1411509.1411513>

General Terms: Algorithms, Theory

Additional Key Words and Phrases: Rank aggregation, consensus clustering, correlation clustering, minimum feedback arc-set, tournaments

**ACM Reference Format:**

Ailon, N., Charikar, M., and Newman, A. 2008. Aggregating inconsistent information: Ranking and clustering. *J. ACM* 55, 5, Article 23 (October 2008), 27 pages. DOI = 10.1145/1411509.1411513 <http://doi.acm.org/10.1145/1411509.1411513>

## 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 [Condorcet 1785; Borda 1781]. There are numerous applications in sports, databases, and statistics [Dwork et al. 2001a; Fagin et al. 2003] 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 [Filkov and Skiena 2003] or combine information from multiple classifiers [Strehl 2002].

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 [Kemeny 1959; Kemeny and Snell 1962].<sup>1</sup> Given  $n$  candidates and  $k$  permutations of the candidates,  $\{\pi_1, \pi_2, \dots, \pi_k\}$ , a *Kemeny optimal* ranking of the candidates is the ranking  $\pi$  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  $\pi_j$  and  $\pi_k$ .<sup>2</sup> 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 article 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 [Bartholdi et al. 1989] and remains NP-hard even when there are only four input lists to aggregate [Dwork et al. 2001a]. 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 [Diaconis and Graham 1977; Dwork

<sup>1</sup>Historically known as *Kemeny aggregation*.

<sup>2</sup>The distance function  $d(\cdot, \cdot)$  is in fact a distance function and is known as the *Kendall tau* distance.

et al. 2001a]. 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  $(i, j) \in A$  or  $(j, i) \in A$ . 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 article, 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) [Bansal et al. 2004]. A useful application of CORRELATION-CLUSTERING is optimally combining the output of different machine learning classifiers [Bansal et al. 2004; Strehl 2002]. Bansal et al. [2004] provide in-depth descriptions of other applications of CORRELATION-CLUSTERING. 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 [Filkov and Skiena 2003].

**1.1. PREVIOUS WORK.** The minimum feedback arc set problem can be approximated to within a factor of  $O(\log n \log \log n)$  in general graphs [Even et al. 1998; Seymour 1995] and has (at least) the same approximation hardness as the vertex cover problem [Karp 1972], which is 1.36 [Dinur and Safra 2002]. More than a decade ago, Bang-Jensen and Thomassen [1992] conjectured that FAS-TOURNAMENT is NP-hard. 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 [Speckenmeyer 1989] and is approximable to within  $5/2$  [Cai et al. 2001].

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 et al. [1996] and Frieze and Kannan [1999] 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 [Ailon et al. 2005], Kenyon-Mathieu and Schudy [2007] 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. Nevertheless, 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 sc 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 is 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  $\pi_i$  and  $\pi_j$  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 [Diaconis and Graham 1977] and can be computed in polynomial time via a minimum cost matching [Dwork et al. 2001a, 2001b]. 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 et al. [2004] gave the first constant-factor approximation for the problem of minimizing disagreements on the complete graph. This factor was improved to 4 by rounding a linear program Charikar et al. [2003]. The weighted version of CORRELATION-CLUSTERING, in which edges have fractional  $\pm$  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 [Gionis et al. 2005; Bansal et al. 2004]. When the edge weights satisfy the probability and the triangle inequality constraints (see Section 1.2), the best previous approximation factor was 3 [Gionis et al. 2005]. CORRELATION-CLUSTERING on complete graphs is MAX-SNP-hard [Charikar et al. 2003] and CONSENSUS-CLUSTERING is NP-hard [Wakabayashi 1998]. However, CONSENSUS-CLUSTERING is not known to be NP-hard if the number of input clusters is constant [Filkov and Skiena 2003].

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 KWIK-SORT, 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 article. 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 2.1), 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*:  $w_{ij} + w_{ji} = 1$  (respectively,  $w_{ij}^+ + w_{ij}^- = 1$ ) for all  $i, j \in V$ .
- (ii) *Triangle Inequality*:  $w_{ij} \leq w_{ik} + w_{kj}$  (respectively,  $w_{ij}^- + w_{jk}^- \leq w_{jk}^-$ ) for all  $i, j, k \in V$ .
- (iii) *Aggregation*: Edge weights are a convex combination of actual permutations (respectively, 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 I 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),

TABLE I. THE PREVIOUS BEST-KNOWN FACTORS ARE SHOWN IN PARENTHESES

	Ordering	Clustering	Ordering-LP	Clustering-LP
Unweighted Instances	3*	3 (4 <sup>†</sup> )	5/2	5/2
Probability Constraints (i)	5*	5(9 <sup>††</sup> )	5/2	5/2
Triangle Inequality (ii)	2*	N/A**		
Probability Constraints + Triangle Inequality (i,ii)	2*	2 (3 <sup>†*</sup> )	2	2
Aggregation (iii)	11/7 (2)	11/7 (2)	4/3	4/3

\*The best-known factor was the  $O(\log n \log \log n)$  algorithm [Even et al. 1998; Seymour 1995] for digraphs.

\*\*Our techniques cannot directly be applied to weighted CORRELATION-CLUSTERING with triangle inequality but no probability constraints.

<sup>†</sup>Charikar et al. [2003].

<sup>††</sup>Charikar et al. [2003], Bansal et al. [2004].

<sup>†\*</sup>Gionis et al. [2005].

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  $\text{NP} \not\subseteq \text{BPP}$ . The question of NP-hardness of FAS-TOURNAMENT has been a long-standing conjecture of Bang-Jensen and Thomassen [1992]. We show a randomized reduction from the problem of finding a 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 Alon [2006], 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 [Ailon et al. 2005], and finally, in Section 11, we discuss open problems.

## 2. Preliminaries and Definitions

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

—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  $\pi$  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  $\pi$ ).<sup>3</sup> 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  $\pi$  on  $V$  minimizing  $\sum_{i, j: i <_{\pi} j} w_{ji}$ . Clearly, the unweighted case can be encoded as a 0/1 weighted case.

<sup>3</sup>By  $i <_{\pi} j$  we mean that  $\pi$  ranks  $i$  before  $j$ .

- RANK-AGGREGATION. We are given a list of permutations (rankings) corresponding to  $k$  voters  $\pi_1, \dots, \pi_k$  on  $V$ . We want to find a permutation  $\pi$  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.  $E^-$ ) denote the set of pairs  $i \neq j$  that are (+)-related (resp. (−)-related). We want to find disjoint clusters  $C_1, \dots, C_m$  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  $\mathcal{C}_1, \dots, \mathcal{C}_k$  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 majority tournament, which is an unweighted tournament that corresponds to the input weighted tournament. Similarly, a weighted CORRELATION-CLUSTERING instance has a corresponding unweighted majority instance.

*Definition 2.1.* 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*  $(i \rightarrow j, j \rightarrow k, k \rightarrow i)$ . 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.

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 3.1.** *KWIKSORT is a randomized algorithm for FAS-TOURNAMENT with expected cost at most three times the optimal cost.*

```

KWIKSORT( $G = (V, A)$ )

If  $V = \emptyset$  then return empty-list
Set  $V_L \rightarrow \emptyset, V_R \rightarrow \emptyset$ .
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.)

```

PROOF. We want to show that  $E[C^{KS}] \leq 3C^{OPT}$ . 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  $E[C^{KS}] = \sum_{t \in T} p_t$ .

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 breaking 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

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

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  $e \in E$ ,

$$\sum_{t:e \in t} \frac{1}{3} p_t \leq 1. \quad (1)$$

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

#### 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  $\bar{w}(e) = w_{ji} = 1 - w(e) \leq w(e)$ . Fix an optimal solution  $\pi^*$ , 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  $c^*(e) = \bar{w}(e)$ . 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 4.1.** *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  $E[C^{KS}] \leq \alpha C^{OPT}$ , that is, KWIKSORT( $G_w$ ) is an expected  $\alpha$ -approximation 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  $\bar{w}(e)$ . 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,  $E[C^{KS}] = B^{KS} + F^{KS}$ , where

$$B^{KS} = \sum_{t \in T} \frac{1}{3} p_t w(t)$$

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

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  $\bar{w}(e) \leq c^*(e)$  for any  $e \in A_w$ . Therefore, if for some  $\alpha > 0$ ,  $w(t) \leq \alpha c^*(t)$  for all  $t$ , then  $E[C^{KS}] \leq \alpha C^{OPT}$  as required.  $\square$

**LEMMA 4.2.** *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  $\bar{w}(e) = 1 - w(e)$ . Fix a triangle  $t$  containing edges  $e_1, e_2, e_3$ , and assume

$$1/2 \leq w(e_1) \leq w(e_2) \leq w(e_3) \leq 1. \quad (2)$$

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,

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

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

In the conference version [Ailon et al. 2005], a weaker bound of 3 was proven for the triangle inequality constraints only case and 2 for the combined constraints. This improvement in Lemma 4.2 is due to Warren Schudy.

Combining Theorem 4.1 and Lemma 4.2, we get

**THEOREM 4.3.** *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  $\{\pi_1, \dots, \pi_k\}$  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 4.3 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( $\{\pi_1, \pi_2, \dots, \pi_k\}$ )  
 Output a permutation  $\pi_i$  chosen uniformly at random from the input permutations.

(In practice, we can pick the permutation  $\pi_i$  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  $z(e) = 2w(e)\bar{w}(e)$ , where  $w(e)$  and  $\bar{w}(e)$  are defined as in Section 4. We claim that

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

Indeed, edge  $e \in A_w$  becomes a backward (respectively, forward) edge with probability  $\bar{w}(e)$  (respectively,  $w(e)$ ), in which case it incurs the cost of  $w(e)$  (respectively,  $\bar{w}(e)$ ). 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 5.1.** *If there exist constants  $\beta \in [0, 1]$  and  $\gamma > 0$  such that*

$$\begin{aligned} \beta w(t) + (1 - \beta)z(t) &\leq \gamma c^*(t) \text{ for all } t \in T, \text{ and} \\ \beta \bar{w}(e) + (1 - \beta)z(e) &\leq \gamma c^*(e) \text{ for all } e \in A_w, \end{aligned}$$

*then the best of KWIKSORT and PICK-A-PERM is a  $\gamma$ -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  $E[C^{PAP}] = B^{PAP} + F^{PAP}$ , where

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

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

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

as required.  $\square$

LEMMA 5.2. *For all  $t \in T$ ,  $\frac{3}{7}w(t) + \frac{4}{7}z(t) \leq \frac{11}{7}c^*(t)$ , and for all  $e \in A_w$ ,  $\frac{3}{7}\bar{w}(e) + \frac{4}{7}z(e) \leq \frac{11}{7}c^*(e)$ .*

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

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

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.  $\square$

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 5.3 follows from Theorem 5.1 and Lemma 5.2, using  $\beta = 3/7$  and  $\gamma = 11/7$ :

THEOREM 5.3. *The best of KWIKSORT on  $G_w$  and PICK-A-PERM is an expected  $11/7$  approximation for RANK-AGGREGATION.*

In using Theorem 5.1 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 [Chaudhuri et al. 2006].

LEMMA 5.4. *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  $e \in A_w$ ,  $\frac{2}{5}\bar{w}(e) + \frac{3}{5}z(e) \leq \frac{6}{5}c^*(e)$ .*

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.  $\square$

Theorem 5.5 follows from Theorem 5.1 and Lemma 5.4, using  $\beta = 2/5$  and  $\gamma = 6/5$ .

**THEOREM 5.5.** *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  $C \cup \text{KWIKCLUSTER}(G')$  .

```

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.<sup>4</sup> 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*

<sup>4</sup>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.

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 6.1.** *Algorithm KWIKCLUSTER is a randomized expected 3-approximation algorithm for CORRELATION-CLUSTERING.*

Now let  $(V, w^+, w^-)$  be a weighted CORRELATION-CLUSTERING instance, where  $w^+, w^- \in (\mathbb{R}^+)^{\binom{V}{2}}$ . 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 6.2 is analogous to Theorem 4.3:

**THEOREM 6.2.** *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 4.3, 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 6.2. 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 6.3.** *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 [Potts 1980; Charikar et al. 2003]. 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 7.1.** *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,

<p>LP for weighted FAS-TOURNAMENT</p> $\min \sum_{i < j} (x_{ij} w_{ji} + x_{ji} w_{ij}) \text{ s.t.}$ $x_{ik} \leq x_{ij} + x_{jk} \quad \forall \text{ distinct } i, j, k$ $x_{ij} + x_{ji} = 1 \text{ for all } i \neq j$ $x_{ij} \geq 0 \text{ for all } i \neq j$	<p>LP for weighted CORRELATION-CLUSTERING</p> $\text{minimize } \sum_{i < j} (x_{ij}^+ w_{ji}^- + x_{ji}^- w_{ij}^+) \text{ s.t.}$ $x_{ik}^- \leq x_{ij}^- + x_{jk}^- \text{ for all distinct } i, j, k$ $x_{ij}^+ + x_{ji}^- = 1 \text{ for all } i \neq j$ $x_{ij}^-, x_{ij}^+ \geq 0 \text{ for all } i \neq j$
<p>LP-KWIKSORT(<math>V, x</math>)</p> <p><i>A recursive algorithm for rounding the LP for weighted FAS-TOURNAMENT.</i></p> <p><i>Given an LP solution:</i></p> <p><math>x = \{x_{ij}\}_{i, j \in V}</math>, returns an ordering on the vertices.</p> <p>If <math>V = \emptyset</math> then return empty-list</p> <p>Pick random pivot <math>i \in V</math>.</p> <p>Set <math>V_R = \emptyset, V_L = \emptyset</math>.</p> <p>For all <math>j \in V, j \neq i</math>:</p> <p style="padding-left: 2em;">With probability <math>x_{ji}</math></p> <p style="padding-left: 4em;">Add <math>j</math> to <math>V_L</math>.</p> <p style="padding-left: 2em;">Else (w/ prob. <math>x_{ij} = 1 - x_{ji}</math>)</p> <p style="padding-left: 4em;">Add <math>j</math> to <math>V_R</math>.</p> <p>Return order</p> <p style="padding-left: 2em;">LP-KWIKSORT(<math>V_L, x</math>), <math>i</math>,</p> <p style="padding-left: 4em;">LP-KWIKSORT(<math>V_R, x</math>)</p>	<p>LP-KWIKCLUSTER(<math>V, x^+, x^-</math>)</p> <p><i>A recursive algorithm for rounding the LP for weighted CORRELATION-CLUSTERING.</i></p> <p><i>Given an LP solution:</i></p> <p><math>x^+ = \{x_{ij}^+\}_{i &lt; j}, x^- = \{x_{ij}^-\}_{i &lt; j}</math>, returns a clustering of the vertices.</p> <p>If <math>V = \emptyset</math> then return <math>\emptyset</math></p> <p>Pick random pivot <math>i \in V</math>.</p> <p>Set <math>C = \{i\}, V' = \emptyset</math>.</p> <p>For all <math>j \in V, j \neq i</math>:</p> <p style="padding-left: 2em;">With probability <math>x_{ij}^+</math></p> <p style="padding-left: 4em;">Add <math>j</math> to <math>C</math>.</p> <p style="padding-left: 2em;">Else (w/ prob. <math>x_{ij}^- = 1 - x_{ij}^+</math>)</p> <p style="padding-left: 4em;">Add <math>j</math> to <math>V'</math>.</p> <p>Return clustering</p> <p style="padding-left: 2em;"><math>\{C\} \cup \text{LP-KWIKCLUSTER}(V', x^+, x^-)</math></p>

FIG. 1. Standard LP relaxations and their corresponding rounding algorithms.

—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 7.2.** *Our clustering LP rounding algorithm LP-KWIKCLUSTERING obtains the following approximation ratios on weighted CORRELATION-CLUSTERING 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 CONSENSUS-CLUSTERING.

The result for CONSENSUS-CLUSTERING is obtained by returning the better of LP-KWIKCLUSTER and PICK-A-CLUSTER.

The bounds in Theorems 7.1 and 7.2 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 feasibly solution to the FAS-TOURNAMENT LP that has value at most  $n/3$  and there exist tournaments with no minimum feedback arc set of size smaller than  $n(1/2 - \varepsilon)$ , where  $\varepsilon$  is arbitrarily small.

We now prove Theorems 7.1 and 7.2. 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 7.1).

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

$$x_{ij}w_{ji} + x_{ji}w_{ij}, \quad (6)$$

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  $B_{ij}^t$  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  $t = \{i, j, k\}$ ,

$$\Pr[A_t \wedge B_{ij}^t] = \Pr[A_t] \Pr[B_{ij}^t | A_t] = \frac{1}{3} p_t x_{ik} x_{kj}.$$

(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  $q_{ij}^t = \frac{1}{3} x_{ik} x_{kj}$ . So the total expected charge to a triplet  $t = \{i, j, k\}$  is  $p_t y(t)$ , where

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

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

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



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

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

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

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

$$B_{LP} = \sum_t p_t \sum_{\{i, j\} \subseteq t} (q_{ij}^t + q_{ji}^t) c_{ij}^*$$

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

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

**LEMMA 7.3.** *If the weight system satisfies the probability constraints (respectively, probability constraints and triangle inequality constraints), then for any  $t \in T$ ,*

$$y(t) \leq \tau \sum_{\{i, j\} \subseteq t} (q_{ij}^t + q_{ji}^t) c_{ij}^*,$$

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

Therefore, in this case,  $B^{LKS} \leq \tau B_{LP}$ . We conclude that  $E[C^{LKS}] \leq \tau C_{LP}$ . This concludes the proof of the first two items of Theorem 7.1.

We now prove the last item of Theorem 7.1 by coupling LP-KWIKSORT with PICK-A-PERM. Recall from Section 5 that the expected value of the PICK-A-PERM algorithm is

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

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

$$E[C^{PAP}] = B_{LP}^{PAP} + F_{LP}^{PAP},$$

where

$$B_{LP}^{PAP} = \sum_t p_t \sum_{\{i, j\} \subseteq t} (q_{ij}^t + q_{ji}^t) z_{ij}$$

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

It is easy to see that  $0 \leq F_{LP}^{PAP} \leq 2F_{LP}$  (because  $z_{ij} \leq 2c_{ij}^*$ , and  $\sum_{t: i, j \in t} p_t (q_{ij}^t + q_{ji}^t) \leq 1$ ). Along with  $F^{LKS} = F_{LP}$ , this implies that  $\frac{2}{3}F^{LKS} + \frac{1}{3}F_{LP}^{PAP} \leq \frac{4}{3}F_{LP}$ .

Likewise, in Lemma 7.4 (proof in Section 8), we bound a convex combination of  $B^{LKS}$  and  $B_{LP}^{PAP}$ .

LEMMA 7.4. *For all  $t = \{i, j, k\}$ ,*

$$\frac{2}{3}y(t) + \frac{1}{3} \sum_{\{i,j\} \subseteq t} (q_{ij}^t + q_{ji}^t) z_{ij} \leq \frac{4}{3} \sum_{\{i,j\} \subseteq t} (q_{ij}^t + q_{ji}^t) c_{ij}^*.$$

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

Combining, we conclude that

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

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 7.1.

We now prove Theorem 7.2, 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  $i \neq j$  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  $B_{\{i,j\}}^t$  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  $B_{\{i,j\}}^t$  conditioned on  $A_t$  is  $q_{\{i,j\}}^t = \frac{1}{3}x_{ki}^+ x_{kj}^-$ . Let  $B_{\{ij\}}^t$  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  $B_{\{ij\}}^t$  conditioned on  $A_t$  is  $q_{\{ij\}}^t = \frac{1}{3}x_{ki}^+ x_{kj}^+$ . Define  $y(t) = \sum_{\{i,j\} \subseteq t} (q_{\{i,j\}}^t + q_{\{ij\}}^t) w_{ij}^+ + q_{\{ij\}}^t w_{ij}^-$ .

For all  $i \neq j$ ,

$$\sum_{t:\{i,j\} \subseteq t} p_t (q_{\{i,j\}}^t + q_{\{ij\}}^t) \leq 1$$

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

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

LEMMA 7.5. *If the weight system satisfies the probability constraints (respectively, probability constraints and triangle inequality constraints), then for any  $t \in T$ ,*

$$y(t) \leq \tau \sum_{\{i,j\} \subseteq t} \left( q_{\{i\}j}^t + q_{\{j\}i}^t + q_{\{ij\}}^t \right) c_{ij}^*,$$

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 7.2.

For CONSENSUS-CLUSTERING, we let  $C_{LP}^{PAC}$  denote the value of PICK-A-CLUSTER. So  $E[C_{LP}^{PAC}] = B_{LP}^{PAC} + F_{LP}^{PAC}$ , where

$$\begin{aligned} B_{LP}^{PAC} &= \sum_t p_t \sum_{\{i,j\} \subseteq t} \left( q_{\{i\}j}^t + q_{\{j\}i}^t + q_{\{ij\}}^t \right) z_{ij} \\ F_{LP}^{PAC} &= \sum_{i < j} \left( 1 - \sum_{t: \{i,j\} \subseteq t} p_t (q_{\{i\}j}^t + q_{\{j\}i}^t + q_{\{ij\}}^t) \right) z_{ij} \geq 0. \\ z_{ij} &= 2w_{ij}^+ w_{ij}^- \end{aligned}$$

LEMMA 7.6. *For all  $t = \{i, j, k\}$ ,*

$$\frac{2}{3}y(t) + \frac{1}{3} \sum_{\{i,j\} \subseteq t} \left( q_{\{i\}j}^t + q_{\{j\}i}^t + q_{\{ij\}}^t \right) z_{ij} \leq \frac{4}{3} \sum_{\{i,j\} \subseteq t} \left( q_{\{i\}j}^t + q_{\{j\}i}^t + q_{\{ij\}}^t \right) c_{ij}^*.$$

Also, it is easy to see that  $z_{ij} \leq 2c_{ij}^*$ , so  $0 \leq F_{LP}^{PAC} \leq 2F_{LP}$  and consequently  $\frac{2}{3}F^{LKS} + \frac{1}{3}F_{LP}^{PAC} \leq \frac{4}{3}F_{LP}$ . Combining this with Lemma 7.6, we conclude that  $\frac{2}{3}C^{LKS} + \frac{1}{3}C_{LP}^{PAC} \leq \frac{4}{3}F_{LP}$ , as desired. The proofs of Lemmas 7.5 and 7.6 can be found in Section 8. This completes the proof of Theorem 7.2.

## 8. Proving Polynomial Inequalities in Polytopes

In this section, we prove Lemmas 7.3, 7.4, 7.5 and 7.6. 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 7.3 and 7.4), 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 7.5 and 7.6), 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 7.3 and 7.4) and clustering (Lemmas 7.5 and 7.6) 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,

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

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

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

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

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

Lemma 7.3 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  $\Pi$ , and similarly  $g$  obtains its maximum value on  $(\mathbf{x}, \mathbf{w})$  for  $\mathbf{w}$ , which is some vertex of  $\Delta$ . 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  $\tilde{f}, \hat{f}, \tilde{g} : \mathbb{R}^3 \rightarrow \mathbb{R}$  are bounded above by 0 on  $\Delta$ .
- (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  $\Delta$ , so these points  $\mathbf{x}$  can be ignored. The points that are left are  $\mathbf{x} \in \Delta$  such that 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  $\Delta_k$  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 II.

Lemma 7.4 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  $\mathbf{x} \in \Delta_2$ , 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 nondegeneracy, its unique global maximum is obtained when  $\nabla_{\mathbf{w}}h = 0$ , which can be easily verified

TABLE II. MAXIMA OF  $\tilde{f}$ ,  $\hat{f}$  AND  $\tilde{g}$  ON  $\Delta_1$ ,  $\Delta_2$ 

function\domain	$\Delta_1$	$\Delta_2$
$\tilde{f}$	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)

to be solved by  $\mathbf{w} = \mathbf{w}^* = (w_1^*, w_2^*, w_3^*)$  defined by

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

(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  $h(1, 1, 0, \mathbf{w}) = -2w_3^2/3 \leq 0$ , proving the desired assertion trivially). Since we are assuming  $x_1 + x_2 + x_3 = 2$ , we have that for any  $1 \leq i < j \leq 3$ ,  $x_i + x_j \geq 1$ , equivalently  $x_i x_j \geq (1 - x_i)(1 - x_j)$ . Therefore (9) implies  $w_i \geq \frac{1}{2} + 2x_i - 1$  for  $i = 1, 2, 3$ . Summing up, we obtain  $w_1 + w_2 + w_3 \geq -\frac{3}{2} + 2(x_1 + x_2 + x_3) = \frac{5}{2} > 2$ . In other words, (9) implies that  $\mathbf{w}^*$  and  $\Delta$  are strictly on different sides of  $H_2$ . Let  $\mathbf{w}' = (w'_1, w'_2, w'_3)$  be any point in  $\Delta$ . Consider the straight line  $\ell$  passing through  $\mathbf{w}'$  and  $\mathbf{w}^*$ , and let  $\mathbf{w}''$  the intersection of this line with  $H_2$ . Restricted to  $\ell$  (and for our fixed  $\mathbf{x} \in \Delta_2$ )  $h$  is a parabola, attaining its maximum on  $\mathbf{w}^*$ . Therefore  $h(\mathbf{x}, \mathbf{w}'') \geq h(\mathbf{x}, \mathbf{w}')$ , 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  $\tilde{h} : \mathbb{R}^4 \rightarrow \mathbb{R}$  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  $\tilde{h} \leq 0$  on  $\{x_1 \leq 1, x_2 \leq 1, x_1 + x_2 \geq 1\} \times \mathbb{R}^2$ . 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  $x_1 + x_2 \geq 1$  and  $x_1, x_2 \leq 1$ .

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,

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

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

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

TABLE III. MAXIMUM OF  $f_{000}, f_{001}, f_{011}, f_{111}, g_{000}, g_{011}, g_{111}$  ON  $\Delta_1, \Delta_2, \Delta_3$ 

function \ domain	$\Delta_1$	$\Delta_2$	$\Delta_3$
$f_{000}$	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)

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

Lemma 7.5 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 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  $\Delta$ , 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.  $\Delta_k$  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 III.

Lemma 7.6 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.

TABLE IV. MAXIMA OF  $f$  GIVEN DIFFERENT CONSTRAINTS ON  $\Delta_1, \Delta_2, \Delta_3$ 

constraint	$\Delta_1$	$\Delta_2$	$\Delta_3$
00****	0:000000	0:000000	0:000000
11****	0:11011 $\frac{1}{2}$	0:11011 $\frac{1}{2}$	infeasible
**01*	$-\frac{2}{3}$ :101011	0:011011	0:011011
0**0**	0:000000	0:000000	0:000000
1**0**	$-\frac{2}{3}$ :110011	$-\frac{2}{3}$ :110011	$-\frac{2}{3}$ :101011
0***0*	0:000000	0:000000	0:000000
1***0*	0:101101	$-\frac{2}{3}$ :110101	0:101101
$xx0ww^*$	0:000000	0:000000	0:000000
$xx^*ww0$	0:000000	0:000000	0:000000
01****	infeasible	0:011 $\frac{1}{2}$ 11	0:011 $\frac{1}{2}$ 11
***00*	0:000000	0:000000	0:000000
***11*	0:101111	0:11011 $\frac{1}{2}$	0:101111
0**1**	$-\frac{5}{4}$ :0001 $\frac{1}{4}$ $\frac{3}{4}$	0:011111	0:011111
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***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***1*	0:11011 $\frac{1}{2}$	0:11011 $\frac{1}{2}$	0:101111
$xx1ww^*$	infeasible	infeasible	0: $\frac{1}{2}$ $\frac{1}{2}$ 1 $\frac{1}{2}$ $\frac{1}{2}$ 1
$xx^*ww1$	0:110111	0:110111	0: $\frac{1}{2}$ $\frac{1}{2}$ 1 $\frac{1}{2}$ $\frac{1}{2}$ 1

The constraint 0\*\*0\*1 means, as an example,  $x_1 = 0, w_1 = 0, w_3 = 1$ . 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  $(x_1, x_2, x_3, w_1, w_2, w_3)$ .

*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 IV.

### 9. NP-Hardness of Feedback Arc Set on Tournaments

All the problems referred to in Table I in Section 1.2 were previously known to be NP-hard except for FAS-TOURNAMENT. In this section, we show:

**THEOREM 9.1.** *Unless  $NP \subseteq BPP$ , 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 [Karp 1972] (in fact, it is MAX-SNP-hard, see Håstad [2001], Newman [2000], and Newman and Vempala [2001]).

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  $\pi$  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  $\pi$  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 Newman [2000] and Newman and Vempala [2001]).

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

$$V^k = \bigcup_{v \in V} \{v_1, \dots, v_k\}$$

$$A^k = \{(u_i, v_j) \mid (u, v) \in A, i, j \in \{1, \dots, k\}\}.$$

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  $\pi$  on  $V^k$  that rank the vertices  $v_1, \dots, v_k$  as one block for all  $v \in V$  (as explained in Alon [2006], 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  $\pi$ ).

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  $\pi$  of  $V^k$ , let  $f_R(\pi)$  denote the number of feedback edges in  $A_R^k$  with respect to  $\pi$ . Denote by  $\mu$  the expected value of  $f_R(\pi)$ , which is the same for all  $\pi$ , and can be efficiently computed. We claim that for  $k = \text{poly}(n)$ , with probability at least  $2/3$ , all rankings  $\pi$  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  $\pi$ , set a random indicator variable  $X_{wz}^{\pi}$  for every nonedge  $\{w, z\}$  of  $G^k$  that equals 1 iff the edge between  $w$  and  $z$  in  $A_R^k$  is backward with respect to  $\pi$ . So  $f_R(\pi) = \sum X_{wz}^{\pi}$ . A simple application of Chernoff bounds [Alon and Spencer 1992] and union bound (over all possible  $(nk)!$  rankings) completes the proof of the claim. It follows that unless  $\text{FAS-DIGRAPH} \in \text{BPP}$ , we cannot solve  $\text{FAS-TOURNAMENT}$  in polynomial time.  $\square$

We wish to thank Noga Alon for ideas significantly simplifying the proof [Alon 2006]. Our initial hardness result was via max-SNP hardness of  $\text{FAS-DIGRAPH}$ , 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 [Ailon et al. 2005], there have been interesting developments in the field.

On the ranking side, Kenyon-Mathieu and Schudy [2007] presented a PTAS for  $\text{FAS-TOURNAMENT}$ , thus considerably improving the constant approximation guarantee presented here. Williamson and Van Zuylen [2007] derandomized the pivot algorithms introduced in this article for both ranking and clustering, with matching approximation guarantees. In addition, Coppersmith et al. [2006] showed that ordering a weighted tournament by in-degree is a 5-approximation for weighted  $\text{FAS-TOURNAMENT}$  with probability constraints, thus obtaining another natural constant factor approximation. Ailon [2008] 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 [2008], which is inspired by this work and improves a result by Balcan et al. [2007] (inspired by Coppersmith et al. [2006]).

On the clustering side, Ailon and Charikar [2005] extended results here to hierarchical clustering, a problem well studied in phylogeny. They generalize KWIK-CLUSTER 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 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  $n \rightarrow \infty$ . 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 [Dwork et al. 2001a; Dwork et al. 2001b]?
- Is CONSENSUS-CLUSTERING NP-Hard for a constant number of clusters [Wakabayashi 1998; Filkov and Skiena 2003]?
- Can we approximate weighted CORRELATION-CLUSTERING with triangle inequalities, but no probability constraints?

ACKNOWLEDGMENTS. 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 [Gionis et al. 2005], 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 for anonymous reviewers.

## REFERENCES

- AILON, N. 2008. Aggregation of partial rankings, p-ratings and top-m lists. *Algorithmica*. DOI 10.1007/s00453-008-9211-1.
- AILON, N., AND CHARIKAR, M. 2005. Fitting tree metrics: Hierarchical clustering and phylogeny. In *Proceedings of the 46th Annual IEEE Symposium on Foundations of Computer Science (FOCS)* (Pittsburgh, PA). IEEE Computer Society Press, Los Alamitos, CA, 73–82.

- AILON, N., CHARIKAR, M., AND NEWMAN, A. 2005. Aggregating inconsistent information: Ranking and clustering. In *Proceedings of the 37th Annual Symposium on the Theory of Computing (STOC)* (Boston, MA). ACM, New York, 684–693.
- AILON, N., AND MOHRI, M. 2008. An efficient reduction of ranking to classification. In *Conference on Learning Theory (COLT)* (Helsinki, Finland).
- ALON, N. 2006. Ranking tournaments. *SIAM J. Disc. Math.* 20, 1, 137–142.
- ALON, N., AND SPENCER, J. H. 1992. *The Probabilistic Method*. Wiley, New York.
- ARORA, S., FRIEZE, A., AND KAPLAN, H. 1996. A new rounding procedure for the assignment problem with applications to dense graph arrangement problems. In *Proceedings of the 37th Annual Symposium on the Foundations of Computer Science (FOCS)* (Burlington, VT). IEEE Computer Society Press, Los Alamitos, CA, 24–33.
- BALCAN, M.-F., BANSAL, N., BEYGELZIMER, A., COPPERSMITH, D., LANGFORD, J., AND SORKIN, G. B. 2007. Robust reductions from ranking to classification. In *Proceedings of the Conference on Learning Theory (COLT)*. Lecture Notes in Computer Science, vol. 4539. Springer-Verlag, New York, 604–619.
- BANG-JENSEN, J., AND THOMASSEN, C. 1992. A polynomial algorithm for the 2-path problem in semi-complete graphs. *SIAM J. Disc. Math.* 5, 3, 366–376.
- BANSAL, N., BLUM, A., AND CHAWLA, S. 2004. Correlation clustering. *Mach. Learn. J. (Special Issue on Theoretical Advances in Data Clustering)* 56, 1–3, 89–113. (Extended abstract appeared in FOCS 2002, pages 238–247.)
- BARTHOLDI, J., TOVEY, C. A., AND TRICK, M. 1989. Voting schemes for which it can be difficult to tell who won the election. *Social Choice Welf.* 6, 2, 157–165.
- BORDA, J. C. 1781. Mémoire sur les élections au scrutin. *Histoire de l'Académie Royale des Sciences*.
- CAI, M.-C., DENG, X., AND ZANG, W. 2001. An approximation algorithm for feedback vertex sets in tournaments. *SIAM J. Comput.* 30, 6, 1993–2007.
- CHARIKAR, M., GURUSWAMI, V., AND WIRTH, A. 2003. Clustering with qualitative information. In *Proceedings of the 44th Annual IEEE Symposium on Foundations of Computer Science (FOCS)* (Boston, MA). IEEE Computer Society Press, Los Alamitos, CA, 524–533.
- CHAUDHURI, K., CHEN, K., MIHAESCU, R., AND RAO, S. 2006. On the tandem duplication-random loss model of genome rearrangement. In *Proceedings of the 17th Annual ACM-SIAM Symposium on Discrete Algorithm (SODA'06)*. ACM, New York, 564–570.
- CONDORCET, M.-J. 1785. Éssai sur l'application de l'analyse à la probabilité des décisions rendues à la pluralité des voix.
- COPPERSMITH, D., FLEISCHER, L., AND RUDRA, A. 2006. Ordering by weighted number of wins gives a good ranking for weighted tournaments. In *Proceedings of the 17th Annual ACM-SIAM Symposium on Discrete Algorithm (SODA'06)*. ACM, New York, 776–782.
- DIACONIS, P., AND GRAHAM, R. 1977. Spearman's footrule as a measure of disarray. *J. Roy. Stat. Soc., Ser. B* 39, 2, 262–268.
- DINUR, I., AND SAFRA, S. 2002. On the importance of being biased. In *Proceedings of the 34th Annual Symposium on the Theory of Computing (STOC)*. ACM, New York, 33–42.
- DWORK, C., KUMAR, R., NAOR, M., AND SIVAKUMAR, D. 2001a. Rank aggregation methods for the web. In *Proceedings of the 10th International Conference on the World Wide Web (WWW10)* (Hong Kong, China), 613–622.
- DWORK, C., KUMAR, R., NAOR, M., AND SIVAKUMAR, D. 2001b. Rank aggregation revisited. Manuscript. (Available from: <http://www.eecs.harvard.edu/~michaelm/CS222/rank2.pdf>.)
- EVEN, G., NAOR, J. S., SUDAN, M., AND SCHIEBER, B. 1998. Approximating minimum feedback sets and multicuts in directed graphs. *Algorithmica* 20, 2, 151–174.
- FAGIN, R., KUMAR, R., AND SIVAKUMAR, D. 2003. Efficient similarity search and classification via rank aggregation. In *Proceedings of the 2003 ACM SIGMOD International Conference on Management of Data* (San Diego, CA). ACM, New York, 301–312.
- FILKOV, V., AND SKIENA, S. 2003. Integrating microarray data by consensus clustering. In *Proceedings of International Conference on Tools with Artificial Intelligence (ICTAI)* (Sacramento, CA), 418–425.
- FRIEZE, A., AND KANNAN, R. 1999. Quick approximations to matrices and applications. *Combinatorica* 19, 2, 175–220.
- GIONIS, A., MANNILA, H., AND TSAPARAS, P. 2005. Clustering aggregation. In *Proceedings of the 21st International Conference on Data Engineering (ICDE)* (Tokyo, Japan).
- HÄSTAD, J. 2001. Some optimal inapproximability results. *J. ACM* 48, 798–859.
- KARP, R. M. 1972. Reducibility among combinatorial problems. In *Complexity of Computer Computations*. Plenum Press, New York, 85–104.

- KEMENY, J. G. 1959. Mathematics without numbers. *Daedalus* 88, 571–591.
- KEMENY, J., AND SNELL, J. 1962. *Mathematical Models in the Social Sciences*. Blaisdell, New York. (Reprinted by MIT Press, Cambridge, 1972.)
- KENYON-MATHIEU, C., AND SCHUDY, W. 2007. How to rank with few errors. In *Proceedings of the 39th Annual ACM Symposium on Theory of Computing (STOC)* (New York, NY). ACM, New York, 95–103.
- NEWMAN, A. 2000. Approximating the maximum acyclic subgraph. M.S. thesis, Massachusetts Institute of Technology, Cambridge, MA.
- NEWMAN, A., AND VEMPALA, S. 2001. Fences are futile: On relaxations for the linear ordering problem. In *Proceedings of the 8th Conference on Integer Programming and Combinatorial Optimization (IPCO)*. 333–347.
- POTTS, C. N. 1980. An algorithm for the single machine sequencing problem with precedence constraints. *Math. Prog.* 13, 78–87.
- SEYMOUR, P. 1995. Packing directed circuits fractionally. *Combinatorica* 15, 281–288.
- SPECKENMEYER, E. 1989. On feedback problems in digraphs. *Graph Theoretic Concepts in Computer Science*, Lecture Notes in Computer Science, vol. 411, Springer-Verlag, New York, 218–231.
- STREHL, A. 2002. PhD dissertation. Ph.D. thesis, University of Texas at Austin.
- WAKABAYASHI, Y. 1998. The complexity of computing medians of relations. *Resenhas* 3, 3, 323–349.
- WILLIAMSON, D. P., AND VAN ZUYLEN, A. 2007. Deterministic algorithms for rank aggregation and other ranking and clustering problems. In *Proceedings of the 5th Workshop on Approximation and Online Algorithms (WAOA)*.

RECEIVED JANUARY 2006; REVISED MAY 2008; ACCEPTED AUGUST 2008