On Rank Aggregation of Multiple Orderings in Network Design

Hadas Shachnai, *Computer Science Dept., Technion, Haifa 32000, Israel**

Lisa Zhang, *Bell Labs, Lucent Technologies, 600 Mountain Ave., Murray Hill, NJ 07974*†

Tomomi Matsui, *Dept. of Information and System Engineering, Chuo University, Tokyo 112-8551, Japan*‡

Abstract

In the classic *rank aggregation (RA)* problem, we are given L input lists with potentially inconsistent orders of n elements; our goal is to find a single order of all elements that minimizes the total number of disagreements with the given orders. The problem is well known to be NP-hard, already for $L = 4$. We consider a generalization of RA, where each list is associated with a *set of orderings*, and our goal is to choose one ordering per list and to find a permutation of the elements that minimizes the total disagreements with the chosen orderings. For the case in which the lists *completely overlap*, i.e. each list contains all n elements, we show that a simple Greedy algorithm yields a $(2 - 2/L)$-approximation for generalized RA. The case in which the lists only *partially overlap*, i.e. each list contains a subset of the n elements, is much harder to approximate. In fact, we show that RA with multiple orderings per list and partial overlaps cannot be approximated within *any* bounded ratio.

Keywords: generalized rank aggregation, master ring, approximation algorithms

1. Introduction

The problem of aggregating inconsistent information from many different sources arises in numerous contexts and disciplines. This includes the ancient problem of ranking a set of contestants or a set of alternatives based on possibly conflicting preferences, and the more recent problem of aggregating lists of Web pages output by different search engines (see, e.g., in [7]). Informally, the rank aggregation problem (RA) is defined as follows. Given L lists, each containing a permutation of the n elements, we seek a permutation π of the elements that has *minimum total distance* to the L permutations. The distance between two permutations can be measured using various functions. We focus here on the Kemeny measure (see, e.g., [7, 3]). The Kemeny distance between the two orderings is the number of pairs of elements that appear in different orders in two orderings.

In this paper, we study a generalization of the rank aggregation problem (GRA), where each list has a set of multiple orderings. For $1 \leq \ell \leq L$, let S_ℓ be the ℓth list and let P_ℓ be the set of orderings for the list where each $\sigma \in P_\ell$ is a permutation of the elements in S_ℓ. Let π be a permutation of all the elements in the input, $\cup_\ell S_\ell$. We define the distance between π and the list S_ℓ to be the minimum distance between π and any $\sigma \in P_\ell$. As before, our objective is to find the optimal π that minimizes the total distance to the L lists. We consider two situations: *complete overlap*, where each list contains all n elements, and *partial overlap*, where each list contains a subset of the n elements. For the latter, Kemeny distance between two orderings is defined on elements common to both orderings. Note that a solution to GRA must be a permutation of *all* elements.

Aggregation of partially overlapping lists has many applications, e.g. aggregating the rankings of Web pages where the rankings do not necessarily contain the same sets of pages [7]. In the following we give another example, the *master ring (MR)* problem that arises in optical network design. MR motivates the generalization of rank aggregation to partial overlaps as well as multiple orderings per list.

Optical networks often consist of a collection of interconnecting rings. A *master ring* contains every node in the network exactly once and respects the node ordering of every individual ring, either in the clockwise

*E-mail: hadas@cs.technion.ac.il. Work partially supported by funding for DIMACS visitors.

†E-mail: ylz@research.bell-labs.com

‡E-mail: matsui@ise.chuo-u.ac.jp

or counter-clockwise direction. The master ring problem is to find such a ring, whenever it exists. Master rings are used in network design to replace a complicated topology resulting from ad-hoc expansions of the network; a master ring serves also as a simple backup topology for routine maintenance (see, e.g., in [1, 2, 9]). Figure 1 shows an instance of the MR problem. The network consists of 3 rings, which contain nodes $abcdef$, $achg$ and $ghcdi$ respectively. A possible master ring is $abghcdefi$. Given a network consisting of $L > 1$

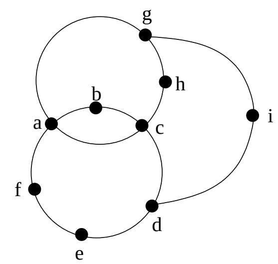

Figure 1: An instance of MR.

interconnecting rings, MR can be stated in terms of GRA. Each ring S_ℓ, $1 \leq \ell \leq L$, has n_ℓ nodes and is a list of n_ℓ elements with $2n_\ell$ possible orderings, n_ℓ clockwise orderings of the nodes and n_ℓ counter-clockwise orderings. A master ring exists if there is a permutation π of all the nodes such that the total Kemeny distance between π and the L lists is zero.

In this paper we use RA to refer to rank aggregation on lists with *single* orderings, and use GRA to refer to rank aggregation on lists with *multiple* orderings. Both problems have complete-overlap and partial overlap instances. It is well known that RA is NP-hard, even when there are only four input lists to aggregate [5, 7]. Most of the previous work on RA focused on complete-overlap instances, where all lists contain the same set of elements. Several deterministic algorithms achieve the approximation ratio of 2 (see in [3]). Ailon et al. [3, 4] gave randomized algorithms, the best of which achieves an expected ratio of $4/3$ to the optimal. A comprehensive survey of other previous work on RA is given in [6] (see also in [12]). The RA problem with partial overlaps was considered in [7]. The paper gives some heuristic solutions. Referring to aggregation that is footrule-optimal, the authors observe that the problem is equivalent to the problem of computing the number of edges to delete to convert a directed graph into a directed acyclic graph (DAG). The master ring problem was studied earlier in [9, 2, 11]. The paper [11] gives an exact algorithm for MR which improves the running time of a brute-force algorithm for the problem.

We focus here on GRA. In Section 2 we give a Greedy algorithm for complete-overlap instances of GRA and two analyses that lead to approximation ratios of 2 and $2 - 2/L$. Rank aggregation with partial overlaps can be extremely hard. In Section 3 we show that already for partial-overlap instances in which each list has 3 orderings and 6 elements, GRA cannot be approximated within *any* bounded ratio.

2. Approximation of GRA with Complete Overlaps

In this section we consider complete-overlap instances of GRA, where each list has multiple orderings and our goal is to find a permutation π that minimizes the total distance to the L lists. Recall that the distance between π and each list S_ℓ is the minimum Kemeny distance between π and any ordering in the set P_ℓ. Let $d(p,q)$ be the Kemeny distance between two orderings p and q. We observe that such instances satisfy the triangle inequality, namely,

Lemma 1 *$d(p,r) + d(q,r) \geq d(p,q)$ for any complete-overlap orderings p, q and r.*

Proof: We use $d_{a,b}(p,q)$ to denote the Kemeny distance between orderings p and q when restricted to elements a and b. In particular, $d_{a,b}(p,q) = 0$ if a and b appear in the same order in orderings p and q;

$d_{a,b}(p,q) = 1$ otherwise. Consider any two elements a and b. If $d_{a,b}(p,q) = 0$ then $d_{a,b}(p,r) + d_{a,b}(q,r) \geq d_{a,b}(p,q)$ trivially. Otherwise, either $d_{a,b}(p,r) = 1$ or $d_{a,b}(q,r) = 1$. Therefore, $d_{a,b}(p,r) + d_{a,b}(q,r) \geq d_{a,b}(p,q)$. Summing over all pairs of elements, we have $d(p,r) + d(q,r) \geq d(p,q)$. ∎

Recall that P_ℓ is the set of orderings for the list S_ℓ, and consider the Greedy algorithm that selects the best ordering in $\cup_\ell P_\ell$ as a solution for GRA.

Theorem 2 *Given a complete-overlap instance of GRA, the best ordering in the set $\cup_\ell P_\ell$ guarantees a 2-approximation to the optimal.*

Proof: For a permutation q, let $D(q, S_\ell) = \min_{\sigma \in P_\ell} d(\sigma, q)$ be the distance between q and the list S_ℓ; and let $D(q) = \sum_\ell D(q, S_\ell)$ be the total distance. We use π to denote the optimal solution, and use p to denote the ordering in $\cup_\ell P_\ell$ that minimizes $d(p,\pi)$. Then,

$$\begin{aligned} D(p) &= \sum_\ell \min_{\sigma \in P_\ell} d(p,\sigma) \\ &\leq \sum_\ell d(p,\pi) + \min_{\sigma \in P_\ell} d(\pi,\sigma) \\ &\leq \sum_\ell 2 \min_{\sigma \in P_\ell} d(\pi,\sigma) \\ &= 2D(\pi) \end{aligned}$$

The first inequality follows from the triangle inequality; the second inequality holds since the definition of $d(\pi,p)$ implies $d(\pi,p) \leq d(\pi,\sigma)$ for all $\sigma \in \cup_\ell P_\ell$. We do not know which ordering is the p that minimizes $d(\pi,p)$. We go through all the orderings $q \in \cup_\ell P_\ell$, and the one that minimizes $D(q)$ gives a 2 approximation. ∎

A different analysis gives an improved approximation ratio for picking the best ordering. We note that, as a special case, the improvement applies to RA (i.e., single ordering per list) and is of interest for small values of L especially since RA is NP hard for $L \geq 4$.

Theorem 3 *Given a complete-overlap instance of GRA, the best ordering in the set $\cup_\ell P_\ell$ guarantees a $(2 - \frac{2}{L})$-approximation to the optimal.*

Proof: As before, we use π to denote the optimal solution. Also, let $\pi_\ell \in P_\ell$ be the ordering that defines $D(\pi, S_\ell)$, i.e. π_ℓ has the minimum distance to π. Given $\Pi = \{\pi_1, \ldots, \pi_L\}$, let us consider the RA problem with one ordering, namely π_ℓ per list. Let $q \in \Pi$ be the ordering that the Greedy algorithm chooses. As in [3], we define the "majority fraction" w_{ij} for every pair of elements i and j. Let f_{ij} be the fraction of the orderings in Π such that i is before j. If $f_{ij} \geq 1/2$, then $w_{ij} = f_{ij}$; otherwise $w_{ij} = 1 - f_{ij}$. Given as input the set of orderings in Π, consider the algorithm Pick-A-Perm (PAP) that selects as a solution randomly and uniformly an ordering in Π; then, as shown in [3], the expected cost of Pick-A-Perm is $E[C_{PAP}] = 2\sum_{ij} w_{ij}(1-w_{ij})$. Since the cost of PAP is a random variable, there exists an ordering whose total cost is at most $E[C_{PAP}]$, and since the order q selected by the Greedy algorithm is the best possible, it follows that

$$\sum_\ell d(q,\pi_\ell) \leq 2\sum_{ij} w_{ij}(1-w_{ij}). \tag{1}$$

Let p be the ordering that the Greedy algorithm chooses for the original GRA instance. We have

$$\begin{aligned} D(p) &= \sum_\ell \min_{\sigma \in P_\ell} d(p,\sigma) \\ &\leq \sum_\ell d(q,\pi_\ell) \end{aligned}$$

$$
\begin{aligned}
&\leq 2\sum_{ij} w_{ij}(1-w_{ij}) \\
&= 2\sum_{ij:w_{ij}<1} w_{ij}(1-w_{ij}) \\
&\leq (2-\frac{2}{L})\sum_{ij}(1-w_{ij}) \\
&\leq (2-\frac{2}{L})D(\pi)
\end{aligned}
$$

The first inequality follows from the definition of p and q and in particular p is the best among all orderings that include Π. The second inequality follows from (1). The third inequality holds since $w_{ij} \leq 1 - \frac{1}{L}$ for $w_{ij} \neq 1$. And finally, the last inequality follows from the fact that the optimal solution has to pay at least $1 - w_{ij}$ for all pairs i and j. ■

3. Hardness of GRA with Partial Overlaps

In the previous section we derived approximation ratios for the complete-overlap instances of GRA. In this section, we give some evidence as to why partial-overlap instances are hard to handle.

We first consider partial-overlap instances of RA, i.e. each list has a single ordering and consists of a subset of all elements. If each list S_ℓ contains two elements only, then RA is equivalent to the feedback arc set problem and is therefore NP hard [7].

In network design, it is often the case that if two rings intersect then they have at least two nodes in common. This construction with two overlapping elements allows tolerance of one node failure when supporting inter-ring traffic. In general, we say that an instance of RA is *K-overlapping* if any two lists are either disjoint or have at least K elements in common. A natural question is whether having the constraint of K-overlappings makes things easier. In the following we consider the case $K = 2$ and show that RA becomes easier only on a very restricted subclass of 2-overlapping instances, while on other instances it remains hard to solve.

Theorem 4 *Consider a 2-overlapping instance of RA. If each list has at most 3 elements, RA is polynomially solvable; If each list has at most 4 elements, RA is NP-hard.*

Proof: We distinguish between two cases:

- *Each list has at most 3 elements.* We create an intersection graph for the lists. Each vertex of the intersection graph corresponds to a list and an edge connects two overlapping lists. There are two cases to consider. Either every list in a connected component has two common elements, say a and b; or the component has at most four distinct elements. In the former case, we try both a before b and b before a in the optimal permutation. Given the relative ordering of a and b, we find the best ordering of the third element in each list relative to a and b. The relative orderings of the third elements do not matter. The validity of the latter case follows from the 2-overlap property applied to lists of at most 3 elements. Therefore, finding the optimal permutation for a component involves trying at most 4! permutations. We also note that no common element appears in two different components. Therefore, we can find the optimal permutation for each component independently.
- *Each list has at most 4 elements.* We show that the problem becomes NP-hard using the following reduction. Given an instance of lists with two elements, we create a 2-overlapping instance of lists with at most four elements. We order the elements in an arbitrary order. We attach a new element x_i to all strings that contains the ith element. Therefore, at most 2 new elements are attached to each original list. Any pair of the original lists have a common new element attached if and only if they

share a common element to begin with. Since rank aggregation with lists of 2 elements is equivalent to the feedback arc set and is NP-hard, when lists have 4 elements and are 2-overlapping the problem is NP-hard as well.

■

Now suppose that each list may have multiple orderings. We show that GRA with partial overlaps and multiple orderings cannot have bounded approximation ratio.

Theorem 5 *If P≠NP, there is no polynomial time algorithm that guarantees a bounded approximation ratio for GRA, even if each list contains exactly 6 elements and 3 orderings.*

Proof: The proof is by reduction from *One-in-Three 3SAT*.
One-in-Three 3SAT (e.g., see [8])
Input: A $p \times q$ matrix A such that (i) each entry is 0 or 1, and (ii) every row contains exactly three 1s.
Question: Is there a vector $z \in \{0,1\}^q$ satisfying $Az = 1_p$? (The vector 1_p denotes the p-dimensional all one vector.)

We denote the index sets of rows and columns of A by $\text{row}(A) = \{1, 2, \ldots, p\}$ and $\text{col}(A) = \{1, 2, \ldots, q\}$, respectively. Given an instance, a $p \times q$ matrix A, of One-in-Three 3SAT, we construct an instance of GRA with $2q$ elements and p lists as follows. First, we introduce a set of $2q$ elements defined by $\{\alpha_i \mid i \in \text{col}(A)\} \cup \{\beta_i \mid i \in \text{col}(A)\}$. For each row-index $h \in \text{row}(A)$, we introduce a subset S_h of 6 elements and a list P_h, defined as follows. Since each row of A has three 1s, there exist three column-indices $i, j, k \in \text{col}(A)$ satisfying $i < j < k$ and $a_{hi} = a_{hj} = a_{hk} = 1$. We put $S_h = \{\alpha_i, \beta_i, \alpha_j, \beta_j, \alpha_k, \beta_k\}$ and introduce a set of orderings of S_h defined by

$$P_h = \{(\beta_i, \alpha_i, \alpha_j, \beta_j, \alpha_k, \beta_k), (\alpha_i, \beta_i, \beta_j, \alpha_j, \alpha_k, \beta_k), (\alpha_i, \beta_i, \alpha_j, \beta_j, \beta_k, \alpha_k)\}.$$

Next, we show that if there exists an ordering π of $2q$ elements such that each list contains a subsequence of π, then the equality system $Az = 1_p$ has a 0-1 valued solution. We define a vector $z' \in \{0,1\}^q$ by setting $z'_{i'} = 1$ if $(\beta_{i'}, \alpha_{i'})$ is a subsequence of π, and $z'_{i'} = 0$ if $(\alpha_{i'}, \beta_{i'})$ is a subsequence of π. Let $h \in \text{row}(A)$ be any row-index of the matrix A, and $i, j, k \in \text{col}(A)$ be column-indices satisfying $i < j < k$ and $a_{hi} = a_{hj} = a_{hk} = 1$. Then, we have that

$$(z'_i, z'_j, z'_k) = \begin{cases} (1,0,0) & (\text{if } (\beta_i, \alpha_i, \alpha_j, \beta_j, \alpha_k, \beta_k) \text{ is a subsequence of } \pi), \\ (0,1,0) & (\text{if } (\alpha_i, \beta_i, \beta_j, \alpha_j, \alpha_k, \beta_k) \text{ is a subsequence of } \pi), \\ (0,0,1) & (\text{if } (\alpha_i, \beta_i, \alpha_j, \beta_j, \beta_k, \alpha_k) \text{ is a subsequence of } \pi), \end{cases}$$

and thus the equality $z'_i + z'_j + z'_k = 1$ holds.

Lastly, we show the inverse implication. Assume that the equality system $Az = 1_p$ has a 0-1 valued solution $z' \in \{0,1\}^q$. We introduce an ordering $\pi = (s_1, t_1, s_2, t_2, \ldots, s_q, t_q)$ of $2q$ elements satisfying that

$$\text{for all } i \in \text{col}(A), \quad (s_i, t_i) = \begin{cases} (\alpha_i, \beta_i) & (\text{if } z'_i = 0), \\ (\beta_i, \alpha_i) & (\text{if } z'_i = 1). \end{cases}$$

It is clear that each list P_h includes a subsequence of π, since z' satisfies the equality indexed by h.

From the above, the obtained GRA instance has a solution whose objective value is equal to 0 if and only if a given One-in-Three 3SAT instance has a 0-1 valued solution. Thus, the existence of a polynomial time algorithm that guarantees a bounded approximation ratio for GRA yields P=NP. ■

The next result shows that we can easily construct K-overlapping instances of GRA that are hard to approximate.

Theorem 6 *If P≠NP, there is no polynomial time algorithm that guarantees a bounded approximation ratio for K-overlap instances of GRA, even if each list contains exactly $6 + K$ elements and 3 orderings.*

Proof: We can construct a K-overlapping instance from the instance defined in the proof of Theorem 5 as follows. We introduce a set of K artificial elements $\{\gamma_1, \ldots, \gamma_K\}$ and for each ordering of the lists, we add the sequence $(\gamma_1, \ldots, \gamma_K)$ to the head. Thus, we transformed the given instance to an essentially equivalent K-overlapping instance. ∎

References

[1] S. Acharya, B. Gupta, P. Risbood, A. Srivastava. *Hitless Network Engineering of SONET Rings*, Globecom 2003.

[2] S. Acharya, B. Gupta, P. Risbood, A. Srivastava. *In-service Optimization of stacked SONET Rings*. Manuscript, 2004.

[3] N. Ailon, M. Charikar and A. Newman, Aggregating Inconsistent Information: Ranking and Clustering In *Proc. of STOC'05*.

[4] N. Ailon, M. Charikar and A. Newman, Proofs of Conjectures in "Aggregating Inconsistent Information: Ranking and Clustering" TR-719-05, Department of Computer Science, Princeton Univ., Jan. 2005.

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

[6] D. Coppersmith, L. Fleischer and A. Rudra, Ordering by weighted number of wins gives a good ranking for weighted tournaments. In *Proc. of SODA*, 2006.

[7] C. Dwork, R. Kumar, M. Naor, and D. Sivakumar. Rank aggregation methods for the web. In *WWW10*, Hong Kong, May 2001.

[8] M.R. Garey and D.S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W.H. Freeman, 1979.

[9] Mobius network management and optimization systems. Lucent Technologies Proprietary. Internal website: http://www-zoo.research.bell-labs.com/∼mobius/.

[10] R. Ramaswami and K. Sivarajan. *Optical networks: a practical perspective*. (Morgan Kaufmann Publishers Inc., San Francisco, 1998).

[11] H. Shachnai, L. Zhang, and T. Matsui. The master ring problem. *Submitted*, 2005.

[12] A. van Zuylen. Deterministic approximation algorithms for ranking and clusterings. Cornell ORIE Tech Report No. 1431, 2005.