Approximate and Dynamic Rank Aggregation

Francis Y.L. Chin [1]

Department of Computer Science and Information Systems, The University of Hong Kong, Pokfulam Road, Hong Kong

Xiaotie Deng [2,1]

83 Tat Chee Avenue, Department of Computer Science, City University of Hong Kong, Kowloon Tong, Hong Kong SAR

Qizhi Fang

Department of Mathematics, Ocean University of China, Qingdao 266071, Shandong, P. R. China

Shanfeng Zhu

83 Tat Chee Avenue, Department of Computer Science, City University of Hong Kong, Kowloon Tong, Hong Kong SAR

Abstract

Rank aggregation, originally an important issue in social choice theory, has become more and more important in information retrieval applications over the Internet, such as meta-search, recommendation system, etc. In this work, we consider an aggregation function using a weighted version of the normalized Kendall-τ distance. We propose a polynomial time approximation scheme, as well as a practical heuristic algorithm with the approximation ratio two for the NP-hard problem. In addition, we discuss issues and models for the dynamic rank aggregation problem.

Key words: Rank aggregation, Kendall-τ distance, coherence, weighted ECC

1 Introduction

The rank aggregation problem finds a "consensus" ranking on a set of alternatives, based on preferences of individual voters. The topic is the focus of social choice theory. Its applications have included elections, and most recently, the meta-search problem on the Internet. The on-line version is especially useful for such applications, in that the top few ranks may be constructed and be presented to the users before the orders of the rest of the candidates are sorted out.

Dwork, et al., [10] studied the rank aggregation problem in the context of Web searching with an eye toward reducing spam in meta-search. They applied the criterion of Kendall-τ distance to evaluate the aggregated rank. The Kendall-τ distance between two ranking lists is the total number of pairs of alternatives that are assigned to different relative orders in the two ranking lists. Given a collection of partial rankings $\tau_1, \tau_2, \cdots, \tau_k$ of alternative web pages, they are interested in the complete ranking π that minimizes the average of the Kendall-τ distance between π and τ_i $(i = 1, 2, \cdots, k)$. The problem was shown to be NP-hard for fixed even $k \geq 4$ (that is, even for aggregation of a small number of ranking lists) and an effective procedure "local Kemenization" was developed to obtain a local Kemeny optimal ranking which satisfies the extended Condorcet criterion. A 2-approximation algorithm was obtained for full list rank aggregation but no proven approximation algorithm was known for partial list rank aggregation [10].

In reality, however, different voters (search engines in our discussion) may not rank the same candidate list. The metric of Kendall-τ distance may not be the best for such cases of partial rankings. If two partial rankings overlap over a small number of alternatives (and thus their Kendall-τ distance is small), one may not have full confidence to conclude that the two rankings differ a little. Dwork, et al., further proposed a normalized Kendall-τ distance to deal with this problem with partial ranking lists. We follow the main idea embedded in this approach and propose to consider both Kendall-τ distance and the size of overlap of the partial ranking lists for an alternative measure for partial

Email addresses: chin@csis.hku.hk (Francis Y.L. Chin), csdeng@cityu.edu.hk (Xiaotie Deng), fangqizhi@public.qd.sd.cn (Qizhi Fang), zhusf@cs.cityu.edu.hk (Shanfeng Zhu).

URL: http://www.cs.cityu.edu.hk/~deng (Xiaotie Deng).

[1] This work described in this work was supported by a grant (CityU 1074/00E) from The Research Grants Council of the Hong Kong Special Administrative Region, China

[2] This work is supported by a joint research grant (N_CityU 102/01) of Hong Kong RGC and NNSFC of China, and by a research grant from CityU of Hong Kong (Project No. 70001545).

ranking aggregation. That is, our measure prorates the normalized Kendall-τ distance by the number of common elements in the two measured rank lists.

Therefore, for a given collection of partial rankings $\tau_1, \tau_2, \cdots, \tau_k$ with different ranking lengths, we are interested in finding a final ranking π of all the candidates such that the sum of $|N_{\tau_i} \cap N_\pi| \left(1 - \frac{D(\tau_i, \pi)}{\binom{|N_{\tau_i} \cap N_\pi|}{2}}\right)$ is maximized, where N_{τ_i} is the set of alternatives in τ_i, N_π is the set of alternatives in π and $D(\tau_i, \pi)$ is the Kendall-τ distance between π and τ_i $(i = 1, 2, \cdots, k)$. We use the convention that the above term will be evaluated to zero if the $N_{\tau_i} \cap N_\pi$ is empty. We comment that this problem is equivalent to Kemeny aggregation problem [10] in a weighted version. Here, the weight of each partial ranking is determined by its overlap with the final ranking.

A particular feature of the rank aggregation problem on the web is that the number of voters is much less than the number of alternatives. As of September 2002, there were only eleven major general purpose search engines, and on the other hand, it was estimated that Google [3] has indexed about 968 million web pages by March 2002 [4]. Secondly, each voter ranks a different set of alternatives, determined by the different coverage of web search engines. Therefore, we should focus on this case where the number of voters is bounded by a constant.

We focus on the new aggregation method (we call it the Coherence aggregation problem). In Section 2, we introduce the formal definitions. We generalize the extended Condorcet criterion (ECC) to the weighted case, and show that the Coherence optimal ranking for partial ranking aggregation satisfies the weighted ECC. In Section 3, we discuss the NP-hardness of the Coherence aggregation problem and present a heuristic algorithm with performance ratio 2, and with a proof that the heuristic solution satisfies the weighted ECC. We note that although the Kemeny aggregation problem and the Coherence aggregation problem are equivalent in the weighted case, they are not equal in approximation. There is no approximation algorithm with constant ratio for Kemeny aggregation for partial rankings. In Section 4, we derive a PTAS for the Coherence aggregation problem. Our approach is motivated by techniques developed in [1,2]. In [2], Arora, et al., presented a unified framework for designing polynomial time approximation schemes (PTASs) for "dense" instances of many NP-hard optimization problems. Their unified framework begins with the idea of exhaustive sampling: picking a small random set of elements, guessing where they go on the optimum solution, and then using their placement to determine placement of other elements. Arora, et al., [2] applied this technique to some 'smooth' assignment problems by shrinking the

[3] http://www.google.com
[4] http://www.searchengineshowdown.com

space of possible placements of the random sample. The unweighted version of our model can be reduced to the maximum acyclic subgraph problem and the required smoothness condition is satisfied. It follows that a polynomial time approximation scheme can be obtained by their general framework for the unweighted case. For the weighted case, the smoothness condition on the coefficients in their unified approach is not satisfied. Our solution further extends and exploits their general methodology and provides new insight into design and analysis of polynomial time approximation scheme.

In Section 5, we discuss the dynamic version which is interesting for application to the meta search problem over the Internet where rank aggregation is dynamic in nature and involves in a large corpus of data. We discuss various algorithmic issues here and propose interesting research problems. In Section 6, we conclude with remarks on our results and discussion on future directions.

2 Definitions

Given a set of alternatives $N = \{1, 2, \cdots, n\}$, a ranking π with respect to N is a permutation of some elements of N which represents a voter's or a judge's preference on these alternatives. If π orders all the elements in N, it is called a complete ranking; otherwise, a partial ranking. For a ranking π, let N_π denote the set of elements presented in π, $|\pi| = |N_\pi|$ denote the number of elements in π, or the length of π. For each $i \in N_\pi$, $\pi(i)$ denote the position of the element i in π, and for any two elements $i, j \in N_\pi$, $\pi(i) < \pi(j)$ implies that i is ranked higher than j by π.

The rank aggregation problem is to combine a number of different rank orderings on a set of alternatives, in order to obtain a 'better' ranking. The notion of 'better' depends on what objective we strive to optimize. Among numerous ranking criteria, the methods based on *Kendall-τ distance* are accepted and studied extensively[14,3–5,7]. The Kendall-τ distance between two rankings π and σ is defined as

$$D(\pi, \sigma) = |\{(i, j) : \pi(i) < \pi(j), \text{but } \sigma(i) > \sigma(j), \forall i, j \in N_\pi \cap N_\sigma\}|.$$

Given a collection of partial rankings $\tau_1, \tau_2, \cdots, \tau_k$, the Kemeny optimal aggregation is a complete ranking π with respect to the union of the elements of $\tau_1, \tau_2, \cdots, \tau_k$ which minimizes the total Kendall-τ distance $D(\pi; \tau_1, \cdots, \tau_k) = \sum_{i=1}^{k} D(\pi, \tau_i)$.

For two partial rankings, if it is not the case that the elements i and j appear in both rankings, the pair (i, j) contributes nothing to their Kendall-τ distance. This implies that Kendall-τ distance ignores the effect of the size of "overlap"

in the measure of the discrepancy of two partial rankings. In view of this, we consider another measurement based on the size of "overlap" and normalized Kendall-τ distance, called *coherence*, to further characterize the relationship of two rankings.

Definition 2.1 *For two partial rankings τ and σ with $|N_\tau \cap N_\sigma| \geq 2$, the coherence of τ and σ is defined as*

$$\Phi(\tau, \sigma) = |N_\tau \cap N_\sigma| \left(1 - \frac{D(\tau, \sigma)}{\binom{|N_\tau \cap N_\sigma|}{2}}\right).$$

When $|N_\tau \cap N_\sigma| \leq 1$, we define the coherence $\Phi(\tau, \sigma) = 0$.

Definition 2.2 *For a collection of partial rankings $\tau_1, \tau_2, \cdots, \tau_K$ and a complete ranking π with respect to $N = N_{\tau_1} \cup \cdots \cup N_{\tau_K}$, $|\tau_s| = n_s \geq 2$ $(s = 1, 2, \cdots, K)$, we denote the total coherence by*

$$\Phi(\pi; \tau_1, \cdots, \tau_K) = \sum_{s=1}^{K} \Phi(\pi, \tau_s) = \sum_{s=1}^{K} n_s \left(1 - \frac{D(\pi, \tau_s)}{\binom{n_s}{2}}\right).$$

The Coherence optimal aggregation is a complete ranking of the elements in N which maximizes the total coherence $\Phi(\pi; \tau_1, \cdots, \tau_K)$ over all complete rankings π. The problem of finding the Coherence optimal aggregations is called Coherence aggregation problem.

In the definition of coherence, the contribution of partial ranking τ_s $(s = 1, 2, \cdots, K)$ to the total coherence $\Phi(\pi; \tau_1, \cdots, \tau_K)$ is

$$n_s \left(1 - \frac{D(\tau_s, \pi)}{\binom{n_s}{2}}\right) = \frac{2}{n_s - 1}\left[\binom{n_s}{2} - D(\tau_s, \pi)\right].$$

Let

$$\omega_s = \frac{2}{n_s - 1}, \qquad s = 1, 2, \cdots, K.$$

If ω_s is considered as the weight of the corresponding ranking, the Coherence aggregation problem is equivalent to the Kemeny aggregation problem in the weighted version, where the weight of each partial ranking is determined by its overlap with the final ranking. When the lengths of all partial rankings are equal, the Coherence aggregation problem is equivalent to the Kemeny aggregation problem proposed by Dwork et al., [10]. Kemeny optimal rankings are of particular interest because they satisfy the extended Condorcet criterion (ECC): if there is a partition $(P, \bar{P})$ of the elements in N such that for any $i \in P$ and $j \in \bar{P}$, the majority prefers i to j, then i must be ranked higher than j. Recently, Dwork, et al., [10] studied the Kemeny optimal aggregation problem

in the context of the Web and showed that ECC has excellent "spam-fighting" properties in the context of meta-search. When the weights are imposed upon the rankings, we can generalize the ECC to the following weighted version.

Weighted Extended Condorcet Criterion (Weighted ECC):

Given partial rankings $\tau_1, \tau_2, \cdots, \tau_K$ and the corresponding weights $\alpha_1, \alpha_2, \cdots, \alpha_K$. Let π be a complete ranking of their aggregation. For any partition $(P, \bar{P})$ of the elements of N, and for all $i \in P$ and $j \in \bar{P}$, if we have $\sum_{s:\tau_s(i)<\tau_s(j)} \alpha_s > \sum_{s:\tau_s(i)>\tau_s(j)} \alpha_s$, then in the aggregation π, i is ranked higher than j. We call π satisfying the weighted extended Condorcet criterion (weighted ECC).

Proposition 2.1 *Let π be a coherence optimal aggregation for partial rankings $\tau_1, \tau_2, \cdots, \tau_K$. Then π satisfies the weighted extended Condorcet criterion with respect to $\tau_1, \tau_2, \cdots, \tau_K$ and their weights $\omega_1, \omega_2, \cdots, \omega_K$.*

Proof. Suppose that there is a partition $(P, \bar{P})$ of N such that for all $i \in P$ and $j \in \bar{P}$ we have that $\sum_{s:\tau_s(i)<\tau_s(j)} \omega_s > \sum_{s:\tau_s(i)>\tau_s(j)} \omega_s$, but there exist two elements $i^* \in P$ and $j^* \in \bar{P}$ such that $\pi(j^*) < \pi(i^*)$. Let (i^*, j^*) be an adjacent such pair in π. Let π' be the ranking obtained by transposing the positions of i^* and j^*. Then we have that

$$\Phi(\pi'; \tau_1, \cdots, \tau_K) - \Phi(\pi; \tau_1, \cdots, \tau_K) = \sum_{s:\tau_s(i^*)<\tau_s(j^*)} \omega_s - \sum_{s:\tau_s(j^*)<\tau_s(i^*)} \omega_s > 0,$$

which contradicts to the optimality of π. $\square$

3 Complexity and Heuristic Algorithm

For partial rankings of length 2, finding Coherence optimal aggregation is exactly the same problem as finding an acyclic subgraph with maximum weight in a weighted digraph, and hence is NP-hard [13]. Bartholdi, et al., [5] proved that the Kemeny aggregation problem is NP-hard for an unbounded number of complete rankings. Their proof can also derive the proof of NP-hardness for the Coherence aggregation problem for an unbounded number of partial rankings with unbounded length. On the other hand, Dwork, et al., [10] discussed the hardness in the setting of interest in meta-search: many alternatives and very few voters. They showed that computing a Kemeny optimal ranking is still NP-hard for any fixed even $K \geq 4$. Their result derives directly the NP-hardness of the Coherence aggregation problem for all integer $K \geq 4$, since odd number of partial rankings can be obtained from even number of complete rankings by splitting one complete ranking into two partial rankings.

Theorem 3.1 *The Coherence aggregation problem for a given collection of*

K partial rankings, for integer $K \geq 4$, is NP-hard.

In the rest of this paper, when the given collection of rankings $\{\tau_1, \cdots, \tau_K\}$ is clear from the context, we will denote $\Phi(\pi; \tau_1, \cdots, \tau_K)$ by $\Phi(\pi)$. The following proposition gives a relationship between an aggregation and its reversal for a given collection of partial rankings, which derives the performance ratio of our heuristic algorithm.

Proposition 3.2 *Let π and π^r be an aggregation total ranking and its reversal with respect to a collection of rankings $\tau_1, \tau_2, \cdots, \tau_K$, respectively. Then*

$$\Phi(\pi) + \Phi(\pi^r) = \sum_{s=1}^{K} n_s.$$

Proof. By the definition of coherence,

$$\begin{aligned}\Phi(\pi) + \Phi(\pi^r) &= \textstyle\sum_{s=1}^{K} n_s \left(1 - \frac{D(\pi,\tau_s)}{\binom{n_s}{2}}\right) + \sum_{s=1}^{K} n_s \left(1 - \frac{D(\pi^r,\tau_s)}{\binom{n_s}{2}}\right) \\ &= \textstyle\sum_{s=1}^{K} \frac{2}{n_s - 1}\left[2\binom{n_s}{2} - D(\pi,\tau_s) - D(\pi^r,\tau_s)\right] = \sum_{s=1}^{K} n_s\end{aligned}$$

where the last equality holds because $D(\pi, \tau_s) + D(\pi^r, \tau_s) = \binom{n_s}{2}$. $\square$

Followed from Proposition 3.2, for any aggregation π and its reversal π^r with respect to $\tau_1, \tau_2, \cdots, \tau_K$, a simple 2-approximation algorithm can be obtained by comparing the coherence values of π and π^r. In this section, we investigate heuristic procedures that construct a better aggregation while taking into account the data of the given instance of the problem. The algorithm consists of two parts: Initial Ranking and Adjustment.

Given a collection of partial rankings $\tau_1, \cdots, \tau_K$ with $|\tau_s| = n_s \geq 2$ ($s = 1, 2, \cdots, K$) and $N = N_{\tau_1} \cup \cdots \cup N_{\tau_K} = \{1, 2, \cdots, n\}$, the weight of each partial ranking is defined as $\omega_s = \dfrac{2}{n_s - 1}$, $s = 1, 2, \cdots, K$. For each ordered pair (i, j) ($i, j \in N$), we define the preference value r_{ij} as the sum of weights of the partial rankings which rank i higher than j, that is,

$$r_{ij} = \sum_{s:\tau_s(i)<\tau_s(j)} \omega_s.$$

Thus, the Coherence aggregation problem is to find a ranking π of N that maximizes the total Coherence $\Phi(\pi) = \displaystyle\sum_{(i,j):\pi(i)<\pi(j)} r_{ij}$.

For each element $i \in N$, denote

$$P(i) = \sum_{j:j\neq i} r_{ji} \quad \text{and} \quad Q(i) = \sum_{j:j\neq i} r_{ij}.$$

We note that $P(i)$ and $Q(i)$ are the contributions to the total Coherence by assigning element i in the lowest position and the highest position of the ranking, respectively. The main idea of Initial Ranking procedure is, in every iteration, to arrange some element to the lowest or highest position, according to their contributions $P(i)$ and $Q(i)$. In Adjustment procedure, if there are two adjacent ordered elements i_k and i_{k+1} such that $r_{i_k i_{k+1}} < r_{i_{k+1} i_k}$ in the ranking obtained already, we transpose the positions of them to get a better ranking.

Initial Ranking Procedure

1. Set $S \leftarrow N$, $u \leftarrow 1$ and $v \leftarrow n$.
2. Compute $\gamma = \max_{i\in S}\{|P(i)-Q(i)|\}$, and denote i^* the element with the largest γ. If $P(i^*) \leq Q(i^*)$, set $\pi(i^*) \leftarrow u,\ u \leftarrow u+1$; if $P(i^*) > Q(i^*)$, set $\pi(i^*) \leftarrow v,\ v \leftarrow v-1$. For each element $j \in S \setminus \{i^*\}$, let

$$P(j) \leftarrow P(j) - r_{i^* j} \quad \text{and} \quad Q(j) \leftarrow Q(j) - r_{j i^*}.$$

   And set $S \leftarrow S \setminus \{i^*\}$.
3. If $v > u$, go to Step 2; else, stop and output the ranking π.

Adjustment Procedure Given a ranking $\pi = i_1, i_2, \cdots, i_n$.

1. Set $\pi^* \leftarrow j_1 \leftarrow i_1$ and $l \leftarrow 1$.
2. Compute $k^* = \begin{cases} 0 & \forall 1 \leq k \leq l,\ r_{j_k i_{l+1}} \leq r_{i_{l+1} j_k} \\ \max\{k : 1 \leq k \leq l,\ r_{j_k i_{l+1}} > r_{i_{l+1} j_k}\} & \text{otherwise} \end{cases}$

   Insert element i_{l+1} at position k^*+1 and get a new ranking with $l+1$ elements:

   For $k \leq k^*$, set $j_k \leftarrow j_k$;

   For $k = k^*+1$, set $j_k \leftarrow i_{l+1}$;

   For $k^*+1 < k \leq l+1$, set $j_k \leftarrow j_{k-1}$.

   Set $\pi^* \leftarrow j_1, \cdots, j_{l+1}$, and $l \leftarrow l+1$.
3. If $l < n$, go to Step 2; else, stop and output the ranking π^*.

The coherence preserved by the Initial Ranking procedure is at least one half of the total value $\sum_{s=1}^{K} n_s$, since this property holds in every iteration with respect to the coherence incurred by the element i^*. We remark that there may be some other rules for choosing the element i^* in the Initial Ranking procedure for choosing and ranking the corresponding element, such as, according to the value (1) $\gamma = \max_{i\in S}\{P(i)\} = P(i^*)$; or (2) $\gamma = \max_{i\in S}\{Q(i)\} = Q(i^*)$. The main idea of our Adjustment procedure is similar to the Local Kemenization procedure investigated in [10], which computes a locally Kemeny optimal aggregation of $\tau_1, \tau_2, \cdots, \tau_K$ being maximally consistent with the initial ranking. Since in Adjustment procedure, insertion of a new element in

each iteration can be viewed as a number of consecutive swaps of neighboring elements in the original ranking, following from the definition of weighted ECC and Proposition 2.1, we have

Proposition 3.3 *Let π^* be a ranking obtained from Adjustment procedure with respect to $\tau_1, \tau_2, \cdots, \tau_K$ and their weights $\omega_1, \omega_2, \cdots, \omega_K$. Then π^* satisfies the weighted ECC.*

4 Polynomial Time Approximation Schemes

Arora, et al., [2] presented a unified framework for developing into polynomial time approximation schemes (PTASs) for "dense" instances of many NP-hard optimization problems, such as, maximum cut, graph bisection and maximum 3-satisfiability. Their unified framework begins with the idea of exhaustive sampling: picking a small random set of elements, guessing where they go on the optimum solution, and then using their placement to determine the placement of other elements. Arora, et al.,[1] applied this technique to assignment problems by shrinking the space of possible placements of the random sample. They designed PTASs for some 'smooth' dense subcases of many well known NP-hard arrangement problems, including minimum linear arrangement, d-dimensional arrangement, betweenness, maximum acyclic subgraph, etc. In this section, we show that the same techniques in [1] can also derive a PTAS for the Coherence aggregation problem, though the coefficients do not satisfy the 'smoothness' condition.

In this section, we consider the Coherence aggregation problem for K partial rankings $\tau_1, \tau_2, \cdots, \tau_K$, where K is an integer independent of $n = |N| = |N_{\tau_1} \cup \cdots \cup N_{\tau_K}|$, $|\tau_s| = n_s \geq 3$ $(s = 1, 2, \cdots, K)$. The weight of each partial ranking ω_s and the preference value r_{ij} are defined as in Section 3. According to Proposition 3.2, for any complete ranking π and its reversal π^r, $\Phi(\pi) + \Phi(\pi^r) = \sum_{s=1}^{K} n_s \geq n$, the optimal value of this problem is no less than $n/2$. Therefore, to obtain an optimal ranking with at least the value $(1 - \gamma)$ times the optimum, where $\gamma > 0$ is arbitrary, it suffices to find a ranking whose value is within an additional factor of ϵn from the optimal value of the optimal ranking for a suitable $\epsilon > 0$. Our main result is presented in the following theorem.

Theorem 4.1 *Suppose the ranking π^* is the optimal solution of the Coherence aggregation problem. Then for any fixed $\epsilon > 0$, in time $n^{O(1/\epsilon^2)}$ we can find a ranking π of N such that*

$$\Phi(\pi) \geq \Phi(\pi^*) - \epsilon n.$$

Several Chernoff-style tail bounds are important in the analysis of randomized procedure. The following result is needed repeatedly in this paper, which we present as a lemma for completeness.

Lemma 4.2 [1] *Let $X_1, X_2, \cdots, X_n$ be n independent random variables such that $0 \leq X_i \leq 1$. Then for $X = \sum_{i=1}^{n} X_i$, $\mu = E[X]$ and $\lambda \geq 0$,*

$$\Pr[|X - \mu| > \lambda] \leq 2e^{-2\lambda^2/n}.$$

Let ϵ be a given small positive, and $t = c/\epsilon$ for some suitable large constant $c > 0$. Here we assume for simplicity that n is a multiple of t. Partition the positions $\{1, 2, \cdots, n\}$ in the final ranking into consecutive equal-sized intervals $I_1, I_2, \cdots, I_t$, each of size n/t. A *placement* is a mapping $g : N \to \{1, 2, \cdots, t\}$ from the set N to the set of intervals $I_1, I_2, \cdots, I_t$. A placement is called *proper* if it maps n/t elements of N to each interval, that is, for every $1 \leq j \leq t$, $|\{i \in N | g(i) = j\}| = n/t$. Every complete ranking corresponds a proper placement which is called the induced placement. Two different rankings may induce the same placement in which case they only differ "locally". The value of a placement g, denoted by $\phi(g)$, is defined as

$$\phi(g) = \sum_{(i,j):g(i)<g(j)} r_{ij} = \sum_{s=1}^{K} \omega_s |\{(i,j) : \tau_s(i) < \tau_s(j) \text{ and } g(i) < g(j)\}|.$$

An optimal placement is a proper placement which maximizes the value $\phi(g)$ over all proper placement g.

Lemma 4.3 *If π is a ranking and g is its induced proper placement, then*

$$\phi(g) \leq \Phi(\pi) \leq \phi(g) + \frac{2Kn}{t}.$$

Proof. The lower bound follows from the fact that $\Phi(\pi) - \phi(g) = \sum_{\pi(i)<\pi(j), g(i)=g(j)} r_{ij} \geq 0$. For each partial ranking τ_s $(s = 1, 2, \cdots, K)$, the elements in τ_s give additional coherence value to the ranking π at most

$$\binom{n/t}{2} \times (\frac{n_s}{n/t}) \times \omega_s = (\frac{n}{t} - 1) \times \frac{n_s}{n_s - 1} \leq \frac{2n}{t}.$$

Therefore, the total difference between $\Phi(\pi)$ and $\phi(g)$ is at most $K \times (2n/t)$. □

Let π^* be an optimal ranking and g^* be its induced placement, and let $\epsilon' = (1 - \frac{2K}{c})\epsilon$. Assume that g is a proper placement such that

$$\phi(g) \geq \phi(g^*) - \epsilon' n,$$

and π is an arbitrary ranking such that g is the placement induced by π. By Lemma 4.3, we have that

$$\Phi(\pi) \geq \phi(g) \geq \phi(g^*) - \epsilon' n \geq \Phi(\pi^*) - \frac{2Kn}{t} - \epsilon' n = \Phi(\pi^*) - \epsilon n.$$

Therefore, finding an optimal ranking to our problem can be reduced to the problem of finding a proper placement within an additive factor of $\epsilon' n$ from the optimal placement.

The optimal placement problem can be formulated as a quadratic arrangement problem:

$$\begin{array}{ll} \text{Max} & \sum_{s=1}^{K} [\sum_{ijkl} c_{ijkl}^{s} x_{ik} x_{jl}] \\ \text{s.t.} & \begin{cases} \sum_{i=1}^{n} x_{ik} = n/t & k = 1, 2, \cdots, t \\ \sum_{k=1}^{t} x_{ik} = 1 & i = 1, 2, \cdots, n \\ x_{ik} = 0, 1 & i = 1, 2, \cdots, n;\ k = 1, 2, \cdots, t \end{cases} \end{array}$$

Here, $c_{ijkl}^{s} = \begin{cases} \omega_s & \text{if } \tau_s(i) < \tau_s(j) \text{ and } 0 < k < l \\ 0 & \text{otherwise} \end{cases}$.

Let g be a proper placement, and let $g_{ik} = 1$ if the element i is assigned to interval I_k by g, and $g_{ik} = 0$ otherwise. For each $i \in N_{\tau_s}$ and $k = 1, 2, \cdots, t$, we define

$$\hat{e}_{ik}^{s} = \sum_{jl} c_{ijkl}^{s} g_{jl} = \omega_s \mid \{j \in N_{\tau_s} : \tau_s(i) < \tau_s(j) \text{ and } g(j) > k\} \mid .$$

We also define $\hat{e}_{ik}^{s} = 0$ for $i \notin N_{\tau_s}$ $(k = 1, 2, \cdots, t)$. Then the proper placement g corresponds to an integral solution to the following linear program:

$$\begin{array}{ll} \text{Max} & \sum_{s=1}^{K} [\sum_{i=1}^{n} \sum_{k=1}^{t} \hat{e}_{ik}^{s} x_{ik}] \\ \text{s.t.} & \begin{cases} \sum_{i=1}^{n} x_{ik} = n/t & k = 1, 2, \cdots, t \\ \sum_{k=1}^{t} x_{ik} = 1 & i = 1, 2, \cdots, n \\ \sum_{jl} c_{ijkl}^{s} x_{jl} = \hat{e}_{ik}^{s} & s = 1, 2, \cdots, K;\ i \in N_{\tau_s};\ k = 1, 2, \cdots, t \\ 0 \leq x_{ik} \leq 1 & i = 1, 2, \cdots, n;\ k = 1, 2, \cdots, t \end{cases} \end{array}$$

We will use the method of exhaustively sampling [1,2] to estimate $\hat{e}_{ik}^{s}$'s. However, since the lengths of K given partial rankings may be quite different from each other, the coefficients of above quadratic arrangement problem do not satisfy the "smooth" condition. Thus, to make a more accurate estimate of

different coefficients, we extend Arora's framework [1,2] by making sampling and estimation for each given ranking separately.

The main idea is: first we make independent experiments for each given rankings to get different sampling sets $T_1, T_2, \cdots, T_K$; then we put all the sampling sets together and enumerate all possible placement h that assign the elements in $T = \cup_{i=1}^{K} T_i$ to intervals $I_1, I_2, \cdots, I_t$; finally we make use of the restriction placement of h on T_s's to estimate the coefficients $\hat{e}_{ik}^s$'s of different rankings.

Our procedure of exhaustively sampling is as follows. Randomly picking with replacement a multi-set T_s of $O(\log n_s/\delta^2)$ elements (where δ is a sufficiently small fraction of ϵ' which we will determine later) from the set N_{τ_s} ($s = 1, 2, \cdots, K$) respectively, we estimate $\hat{e}_{ik}^s$ by the sum $(n_s/|T_s|)\omega_s|\{j \in T_s : \tau_s(i) < \tau_s(j) \text{ and } g(j) > k\}|$. Thus, we chose randomly a multi-set $T = T_1 \cup \cdots \cup T_K$ with size $|T| = O(\log n)$. Since the optimal placement is not known in advance, we enumerate all possible function $h : T \to \{1, 2, \cdots, t\}$ that assign elements in T to intervals $I_1, I_2, \cdots, I_t$. For each such function, we solve a linear program $\mathcal{M}_h$ described below and round the (fractional) optimal solution to construct a proper placement. Among all these placements, we pick up one with maximum value. When the function h we considered is the same as h^* which is the restriction of an optimal placement g^* to T, the placement g we get from the linear program $\mathcal{M}_h$ will satisfy $\phi(g) \geq \phi(g^*) - \epsilon' n$ with high probability, over the random choice of T.

Let h be a given function $h : T \to \{1, 2, \cdots, t\}$. For simplicity, we will identify h with its restrictions on T_s's ($s = 1, 2, \cdots, K$) in the rest of this section. For the partial ranking τ_s, we compute an estimate e_{ik}^s of the value $\hat{e}_{ik}^s$ when assigning the element $i \in N_{\tau_s}$ to interval I_k ($k = 1, 2, \cdots, t$) in any placement g whose restriction to T_s is h:

$$e_{ik}^s = \frac{n_s}{|T_s|}\omega_s \mid \{j \in T_s : \tau_s(i) < \tau_s(j) \text{ and } h(j) > k\} \mid,$$

and set $e_{ik}^s = 0$ for the element $i \notin N_{\tau_s}$ ($k = 1, 2, \cdots, t$).

Lemma 4.4 *Pick uniformly at random with replacement a multi-set T_s of $O(\log n_s/\delta^2)$ elements from N_{τ_s}. Let g be a placement and h be the restrictions of g on T_s. Then with high probability (over the choice of sample T_s),*

$$\mid e_{ik}^s - \hat{e}_{ik}^s \mid \leq 3\delta. \tag{4.1}$$

Proof. Let X_l be a random variable that equals $\omega_s = 2/(n_s - 1)$ if the lth element sampled is j and $\tau_s(i) < \tau_s(j)$, $g(j) > k$; otherwise, $X_l = 0$. Note that

$$\sum_l X_l = \frac{|T_s|}{n_s} e_{ik}^s, \quad E\left[\sum_l X_l\right] = \frac{|T_s|}{n_s}\hat{e}_{ik}^s.$$

Divide each X_l by ω_s to scale it to $\{0,1\}$-variable. Applying Lemma 4.2 to the sum of $X_1, X_2, \cdots, X_{|T_s|}$ after scaling, with $\lambda = \delta|T_s|$ and $|T_s| = O(\log n_s/\delta^2)$, we have with high probability that

$$\frac{|T_s|}{n_s\omega_s} \mid e^s_{ik} - \hat{e}^s_{ik} \mid \leq \lambda = \delta|T_s|, \quad \text{i.e.} \quad \mid e^s_{ik} - \hat{e}^s_{ik} \mid \leq \delta n_s \omega_s \leq 3\delta.$$

□

Consider the following linear program $\mathcal{M}_h$:

$$\mathcal{M}_h: \quad \text{Max} \quad Z(x) = \sum_{s=1}^{K}(\sum_{i=1}^{n}\sum_{k=1}^{t} e^s_{ik}x_{ik})$$

$$\text{s.t.} \quad \begin{cases} \sum_{i=1}^{n} x_{ik} = n/t & k = 1,2,\cdots,t \\ \sum_{k=1}^{t} x_{ik} = 1 & i = 1,2,\cdots,n \\ \Big| \sum_{j:\tau_s(i)<\tau_s(j)} \sum_{l:l>k} \omega_s x_{jl} - e^s_{ik} \Big| \leq 3\delta & s = 1,2,\cdots,K;\ i \in N_{\tau_s};\ k = 1,2,\cdots,t \\ 0 \leq x_{ik} \leq 1 & i = 1,2,\cdots,n;\ k = 1,2,\cdots,t \end{cases}$$

Let x^h be the optimal solution for $\mathcal{M}_h$. We round x^h_{ik} using randomized rounding techniques of Raghavan and Thompson [16] to obtain a placement $\tilde{r}$ and corresponding proper placement r^h as follows: (1) for each element i, independently take $\tilde{r}(i) = k$ with probability x^h_{ik}; (2) construct a proper placement r^h from $\tilde{r}$ by moving elements from intervals with more than n/t elements assigned to them to intervals with less than n/t elements assigned to them arbitrarily. We will discuss the relation between the optimal value $Z(x^h)$ of $\mathcal{M}_h$ and the value of corresponding placement r^h, $\phi(r^h)$. Let

$$Z_s(x^h) = \sum_{i=1}^{n}\sum_{k=1}^{t} e^s_{ik}x^h_{ik}, \qquad Z(x^h) = \sum_{s=1}^{K} Z_s(x^h);$$

$$\phi_s(\tilde{r}) = \omega_s|\{(i,j) : \tau_s(i) < \tau_s(j) \text{ and } \tilde{r}(i) < \tilde{r}(j)\}|, \quad \phi(\tilde{r}) = \sum_{s=1}^{K} \phi_s(\tilde{r}).$$

Lemma 4.5 *Let h be a function that assigns elements of T to intervals $I_1, I_2, \cdots, I_t$, and r^h be the proper placement constructed from the optimal fractional solution x^h of $\mathcal{M}_h$. Then*

$$\phi(r^h) \geq Z(x^h) - 4K\delta n. \tag{4.2}$$

Proof. First think of the placement $\tilde{r}$ obtained after randomized rounding of x^h as a vector $\tilde{x}$ such that $\tilde{x}_{ik} = 1$ if and only if $\tilde{r}(i) = k$. From the randomized rounding procedure, we obtain that for the partial ranking τ_s $(s = 1, \cdots, K)$,

$$E\left[\sum_{i=1}^{n}\sum_{k=1}^{t} e^s_{ik}\tilde{x}_{ik}\right] = Z_s(x^h).$$

Let X_{ik} be the random variable taking the value e^s_{ik} if $\tilde{x}_{ik} = 1$ and 0 otherwise. Scaling X_{ik}'s to the interval $[0, 1]$ ($e^s_{ik} = O(1)$), and applying Lemma 4.2 to the sum of the scaled variables X_{ik}, we have with high probability,

$$\Big| \sum_{i=1}^{n} \sum_{k=1}^{t} e^s_{ik} \tilde{x}_{ik} - Z_s(x^h) \Big| \leq O\left(\sqrt{n_s \log n_s}\right). \tag{4.3}$$

Next we consider the difference between $\sum_{i=1}^{n} \sum_{k=1}^{t} e^s_{ik} \tilde{x}_{ik}$ and the partial placement value $\phi_s(\tilde{r})$. Let

$$\tilde{f}^s_{ik} = \sum_{j:\tau_s(i)<\tau_s(j)} \sum_{l:l>k} \omega_s \tilde{x}_{jl} \quad \text{and} \quad f^s_{ik} = \sum_{j:\tau_s(i)<\tau_s(j)} \sum_{l:l>k} \omega_s x^h_{jl}.$$

By the definition of $\tilde{r}$, we have $E[\tilde{f}^s_{ik}] = f^s_{ik}$. Let Y_{jl} be the random variable taking the value ω_s if $\tilde{x}_{jl} = 1$ and 0 otherwise. Applying Lemma 4.2 to the sum of random variables Y_{jl}/ω_s, we obtain with high probability that $\tilde{f}^s_{ik} \geq f^s_{ik} - O(\sqrt{\log n_s / n_s})$. Also since x^h is a feasible solution to $\mathcal{M}_h$, $|f^s_{ik} - e^s_{ik}| \leq 3\delta$, we have

$$\tilde{f}^s_{ik} \geq f^s_{ik} - O(\sqrt{\log n_s / n_s}) \geq e^s_{ik} - 3\delta - O(\sqrt{\log n_s / n_s}). \tag{4.4}$$

Combining the formulas (4.3), (4.4) and $\sum_{i=1}^{n} \sum_{k=1}^{t} \tilde{x}_{ik} = n_s$,

$$\begin{aligned} \phi_s(\tilde{r}) &= \textstyle\sum_{i=1}^{n} \sum_{k=1}^{t} \tilde{f}^s_{ik} \tilde{x}_{ik} \geq \sum_{i=1}^{n} \sum_{k=1}^{t} [e^s_{ik} - 3\delta - O(\sqrt{\log n_s / n_s})] \tilde{x}_{ik} \\ &= \textstyle\sum_{i=1}^{n} \sum_{k=1}^{t} e^s_{ik} \tilde{x}_{ik} - \sum_{i=1}^{n} \sum_{k=1}^{t} [3\delta + O(\sqrt{\log n_s / n_s})] \tilde{x}_{ik} \\ &\geq Z_s(x^h) - 3\delta n_s - O(\sqrt{n_s \log n_s}). \end{aligned} \tag{4.5}$$

Therefore,

$$\begin{aligned} \phi(\tilde{r}) &= \textstyle\sum_{s=1}^{K} \phi_s(\tilde{r}) \geq \sum_{s=1}^{K} [Z_s(x^h) - 3\delta n_s - O(\sqrt{n_s \log n_s})] \\ &\geq Z(x^h) - 3K\delta n - O(\sqrt{n \log n}). \end{aligned} \tag{4.6}$$

From the construction of $\tilde{r}$, we have $||\{i : \tilde{r}(i) \in I_k\}| - n/t| \leq O(\sqrt{n \log n})$, with high probability. Thus, we move at most $O(\sqrt{n \log n})$ elements to obtain the proper placement r^h from $\tilde{r}$. This changes the value of the placement at most $O(\sqrt{n \log n})$. It follows (4.6) that

$$\phi(r^h) \geq Z(x^h) - 3K\delta n - O(\sqrt{n \log n}) \geq Z(x^h) - 4K\delta n.$$

The last inequality holds because K and δ are both constant, and $O(\sqrt{n \log n}) < K\delta n$ for large n. $\square$

Lemma 4.6 *Let g^* be the optimal placement, h^* be the restriction of g^* to the sample T and r^* be the proper placement constructed from the optimal solution x^* of $\mathcal{M}_{h^*}$. Then*

$$\phi(r^*) \geq \phi(g^*) - \epsilon' n.$$

Proof. It follows from Lemma 4.4 that with high probability, g^* is a feasible solution to $\mathcal{M}_{h^*}$, hence,

$$Z(x^*) \geq Z(g^*) \geq \phi(g^*) - 3K\delta n,$$

where the second inequality is obtained by substituting the lower bound on the estimates e^s_{ik} in (4.1). Also from Lemma 4.5, $\phi(r^*) \geq Z(x^*) - 4K\delta n$, we have that

$$\phi(r^*) \geq \phi(g^*) - 7K\delta n.$$

By choosing $\delta = \epsilon'/7K$, the result follows. □

In this procedure, we enumerate all possible function $h : T \to \{1, 2, \cdots, t\}$ and choose a placement with maximum value among all placement r^h constructed. Since r^* is a candidate for our chosen placement r^h, and we choose the placement with maximum value which is no less than the value of r^*, therefore, we obtain the desired result of Theorem 4.1.

The PTAS described above uses randomization in picking the sample set of elements T and in rounding the optimal solution to linear program $\mathcal{M}_h$. For the procedure of rounding the optimal solution x^h of linear program $\mathcal{M}_h$, we can derandomize it in a standard way using the method of conditional probabilities [15]. As discussed in [1] (also in [11]), the procedure of sampling the set of elements T_s can be substituted by an alternative way of picking random walks of length $|T_s|$ on a constant degree expander graph. Since there are only polynomial many random walks of length $|T_s| = O(\log n_s/\delta^2)$ on this expander, the procedure of sampling the total set T can be substituted by picking polynomial many random walks of length $O(\log n/\delta^2)$. Thus, we can derandomize the algorithm by exhaustively going through all possibilities, i.e., $t^{|T|} = t^{O(\log n/\delta^2)} = n^{O(1/\epsilon^2)}$ placements of the elements in the sample. The running time of our algorithm is $n^{O(1/\epsilon^2)}$.

5 Dynamic Ranking Problems

In meta search applications, the rank aggregation problem is often dynamic. Partial rank lists from the voters arrive and become available to the aggregation function a block at a time. To simplify our discussion, we consider a model where the rank list of each voter arrives one candidate by one candidate in the decreasing order of their ranks. The arrival times of rank lists of different

voters are not coordinated and are in arbitrary orders. That is, the rank list of Voter 1 may all arrive before any information is known of Voter 2's rank list; or they may also arrive alternatively, one from Voter 1's list and one from Voter 2's list. Such uncertainty is a matter of fact in today asynchronous communication networks on which the Internet is based. Note that information from some voters may not arrive at all. Still the user request of an aggregated rank list is to be met.

Naturally, a data structure problem pops up itself here: update the aggregated rank list when a piece (or a block) of new data (in the form of partial rank lists) arrive. The problem would require different data structures for different aggregation functions. For concreteness and illustration purpose, we discuss the solution for the Borda's rule. Recall that the Borda's rule is a positional method where each candidate in a rank list is assigned a score equal to the number of candidates that ranked below it, and its total score is the sum of its scores in all the rank lists. The final rank is in the decreasing order of candidate's total scores. For simplicity, we consider candidates of the same total score tied for the position. Another issue for our application is how to assign scores to candidates whose scores are not yet known. There are various methods dealing with it, we should adopt one that assign the same score to all candidates not yet appeared in a voter's partial rank list, at the same total value as they have. For a given collection of partial lists, the above definition specifies an aggregation value for the candidates and results in a rank list for them. However, each time a partial list gets updated with its next candidate becomes known, the aggregated value of all the candidate's that has not appeared in the partial list will change that may result in a change of the aggregated rank list.

In addition to the data structure problem, there are other related problems. The aggregated rank of some candidates may become fixed, no matter what the not-yet-known partial lists of some voters. To determine this subset of candidates is interesting for some applications. Sometimes we may not be interested in the ranks of all candidates but the top few (and not even their orders but the fact that they are identified to be among the top few). The computational problem can be easy for some aggregation functions and difficult for others. Note that, we may be interested in making a minimum number of the total size of the partial lists of the voters to determine the top few since that queries over the Internet are costly. However, in the worst case, we may have to go through all the lists even to determine the top element in a aggregate ranking list for many aggregation method, e.g., for the Borda rule.

Proposition 5.1 *To determine the top element of the aggregated ranking list for m lists of n candidates according to the Borda's rule, we may have to go through all the elements (but the last one) for all the lists.*

Obviously, such extreme cases are rare and may represent issues deserve further studies in information retrieval. For example, if the candidate lists are obtained by keyword searches on different search engines, such worst case outcome may represent cases of a high level of ambiguity in the language. Therefore, we introduce the following definitions:

Definition 5.2 *Consider m lists of n candidates, if the k-th ranked element in the aggregated list, according to a aggregation rule, can be determined by examination of a total of $O(mk)$ elements in the m candidate lists, the collection of m lists is called a coherent collection for the aggregation rule.*

A study of coherent collections for social choice rules would be an interesting research topic. A more relaxed defition could allow the number of items examined to be up to $O(mf(k))$ for a moderately growing function $f(k)$. In comparison, we may define a non-coherent collection of candidate lists to be one such that it is necessary to examine $\Omega(mg(n))$ elements, for a non-trivial increasing function $g(n)$, to determine the top k ranked element in the aggregated list.

6 Remarks and Discussion

The application of rank aggregation methods to meta search has attracted research attention recently [10,19]. Considering the distinct features in the context of meta-search on the web, we have developed a new rank aggregation method based on the criterion of Coherence. We have proposed not only a practical heuristic algorithm with the solution satisfying the weighted extended Condorcet criterion, but also a theoretical polynomial time approximation scheme (PTAS) for the Coherence aggregation problems. Our algorithm extends and exploits the general framework of Arora, et al., [1,2], for design and analysis of polynomial time approximation schemes.

Our work combines the normalized Kendall-τ distance and the size of overlap between ranking lists in rank aggregation context. Other metrics in social choice theory are also worth of further exploration with the algorithmic approach.

Note that, the work of Dwork, et al., [10], first seriously applies algorithmic method to the study of rank aggregation and their approximation algorithm of ratio two for full ranking Kendall-τ distance minimization relies on its deep relationship with Spearman's footrule distance, developed in Statistics [8]. Our work extends their proposed normalized Kendall-τ distance for partial rankings and obtains a polynomial time approximation scheme.

There are other models where ranking lists are weighted differently. Compared with traditional voting problem where each voter is treated equally in the aggregation procedure, each voter in weighted case will make different contribution to the final aggregation result. The weight for each voter (search engine) could be computed according to its quality (performance). Some approaches have been proposed to evaluate the quality of the web search engine, such as statistical approach in [18] where some statistical information such as query term frequency is kept to predict the quality of the search engine, and learning based approach in [9,12] where users past retrieval experiences on these search engines are utilized to predict the quality of them.

Dynamic models for the rank aggregation problem is important for the applications to information retrieval over the Internet. A practical issue related to meta search is that the delays between submitting a query and obtaining candidate lists from different search engines may not be even [6]. It would be interesting to include this factor into consideration.

References

[1] Sanjeev Arora, Alan M. Frieze, Haim Kaplan: A New Rounding Procedure for the Assignment Problem with Applications to Dense Graph Arrangement Problems. FOCS 1996: 21-30

[2] Sanjeev Arora, David R. Karger, Marek Karpinski: Polynomial time approximation schemes for dense instances of NP-hard problems. STOC 1995: 284-293 (1999): 193-210.

[3] J.P. Barthelemy, A. Guenoche and O. Hudry, Median linear orders: Heuristics and a branch and bound algorithm, *European Journal of Operational Research*, Vol. 42 (1989): 313-325.

[4] J.P. Barthelemy and B. Monjardet, The median Procedure in cluster analysis and social choice theory, *Mathematical Social Sciences*, Vol. 1 (1981): 235-267.

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

[6] Cai MC, Deng XT, Wang LS Approximate sequencing for variable length tasks THEOR COMPUT SCI 290 (3): 2037-2044 JAN 3 2003.

[7] I. Charon, A. Guenoche, O. Hudry and F. Woirgard, New results on the computation of median orders, *Discrete Mathematics* Vol. 165/166 (1997): 139-153.

[8] P. Diaconis and R. Graham, Spearman's footrule as a measure of disarray, *Journal of the Royal Statistical Society*, Series B, Vol. 39 (1977): 262-268.

[9] D. Dreilinger and A. Howe, Experiences with Selecting Search Engines Using Metasearch. *ACM Transactions on Information System*, Vol. 15 (1997): 195-222.

[10] C. Dwork, R. Kumar, M. Naor and D. Sivakumar, Rank aggregation methods for the web, *WWW10* (2001), 613-622.

[11] D. Gillman, A Chernoff bound for random walks on expanders, *SIAM J. Computing*. Vol. 27 (1998): 1203-1220.

[12] S. Gauch, G. Wang and M. Gomez, ProFusion: Intelligent fusion from multiple distributed search engines. *Journal of Universal Computer Science*, Vol. 2 (1996): 637-649.

[13] R.M. Karp, Reducibility among combinatorial problems, in: R.E. Miller and J.W. Thatcher, eds., *Complexity of Computer Computations* (Plenum Press, New York, 1972) 85-103.

[14] J.G. Kemeny, Mathematics without numbers, Daedalus 88(1959): 577-591.

[15] P. Raghavan, Probabilistic construction of deterministic algorithms: Approximating packing integer programs, *Journal of Computer and System Sciences* Vol. 37 (1988): 130-143.

[16] P. Raghavan and C. Thompson, Randomized rounding: a technique for provably good algorithms and algorithmic proofs, *Combinatorica* Vol. 7 (1987): 365-374.

[17] D. G. Saari. The mathematics of voting: Democratic symmetry. Economist, pp. 83, March 4, 2000.

[18] Z. Wu, W. Meng, C.T. Yu and Z. Li, Towards a highly-scalable and effective metasearch engine. *WWW 2001*, 386-395.

[19] Zhu SF, Fang QZ, Deng XT, Zheng WM, Metasearch via voting. Lecture Notes in Computer Science Vol. 2690: pp.734-741, 2003.