Ranking the vertices of a complete multipartite paired comparison digraph

Gregory Gutin *
Anders Yeo
Department of Mathematics and Computer Science
Odense University, Denmark

September 1, 2003

Abstract

A paired comparison digraph (abbreviated to PCD) $D = (V, A)$ is a weighted digraph in which the sum of the weights of arcs, if any, joining two distinct vertices equals one. A one-to-one mapping α from V onto $\{1, 2, ..., |V|\}$ is called a ranking of D. For every ranking α, an arc $vu \in A$ is said to be forward if $\alpha(v) < \alpha(u)$, and backward, otherwise. The length of an arc vu is $\ell(vu) = \epsilon(vw)|\alpha(v) - \alpha(u)|$, where $\epsilon(vw)$ is the weight of vu. The forward (backward) length $f_D(\alpha)$ ($b_D(\alpha)$) of α is the sum of the lengths of all forward (backward) arcs of D. A ranking α is forward (backward) optimal if $f(\alpha)$ is maximum ($b(\alpha)$ is minimum). M. Kano (Disc. Appl. Math., 17 (1987) 245-253) characterized all backward optimal rankings of a complete multipartite PCD D and raised the problem to characterize all forward optimal rankings of a complete multipartite PCD L. We show how to transform the last problem into the single machine job sequencing problem of minimizing total weighted completion time subject to precedence "parallel chains" constraints. This provides an algorithm for generating all forward optimal rankings of L as well as a polynomial algorithm for finding

*Corresponding author. This work was supported by the Danish Research Council under grant no. 11-0534-1. The support is gratefully acknowledged.

the average rank of every vertex in L over all forward optimal rankings of L.

1 Introduction

Kano and Sakamoto [7, 8, 9] introduced a few new methods (forward, backward and mutual) of ranking the vertices of a paired comparison digraph (abbreviated to PCD). Advantages and applications of these methods were described in [4, 7, 8, 9]. We only note that, for tournaments, all these methods, unlike some others, coincide with the most popular approach consisting of computing the scores, the number of games won by each player, and comparing them (see [7]).

It is not difficult to find all optimal mutual rankings of any PCD using Theorem 1 in [7]. At the same time the problems of finding a forward or backward optimal ranking are NP-hard (see Theorem 6.5 in [9] and Theorem 3.1 here). Moreover, Brightwell and Winkler [1] showed that the problem of counting the number of backward optimal rankings of an acyclic digraph is $\#P$-complete. In contrast, Kano characterized all optimal backward rankings of a complete multipartite PCD (Theorem 1 in [6]).

Kano and Sakamoto derived a characterization of all forward optimal rankings of a complete multipartite PCD containing not more than two vertices in each colour class (Theorem 4 in [7]) and Kano [6] raised the problem to characterize all forward optimal rankings of any complete multipartite PCD L. We show how to transform the last problem into the single machine job sequencing problem of minimizing total weighted completion time subject to precedence "parallel chains" constraints. The single machine job sequencing problem of minimizing total weighted completion time subject to various precedence constraints has been considered in a number of papers (see [3]).

The transformation above provides an algorithm for generating all forward optimal rankings of L as well as a polynomial algorithm for computing the average rank (called the *proper forward rank*) of every vertex in L over all forward optimal rankings of L. A polynomial algorithm for finding the proper backward rank of every vertex in L was described in [4]. In contrast, the problem of calculating the proper backward rank of a vertex of an acyclic digraph is proved [1] to be polynomially equivalent to a $\#P$-complete problem.

Kano [6] proved his main theorem using an approach based on Lemma 2.3 [9], i.e. he considered differences $f_D(\alpha_{xy}) - f_D(\alpha)$ (see Section 2.) Our approach is based on considering the explicit form of the function $f_D(\alpha)$. Using our approach the main result of [6] can be proved in a somewhat easier manner.

2 Terminology and notation

Let $D = (V, A)$ be a weighted digraph in which every arc xy has a positive real weight $\epsilon(xy)$. A digraph D is called a *paired comparison digraph* if D satisfies the following conditions:

1. $\epsilon(xy) + \epsilon(yx) = 1$ if both xy and yx are arcs;
2. $\epsilon(xy) = 1$ if $xy \in A$ but $yx \notin A$.

A PCD D is said to be *multipartite* if the vertex set V of D can be partitioned into *colour classes* $V_1, ..., V_r$ such that there is no arc between any pair of vertices from the same colour class. A multipartite PCD D is *complete* if every two vertices from different colour classes are joined by at least one arc. A complete multipartite PCD D is a *complete* PCD if every colour class of D consists of a single vertex.

The *positive (negative*, resp.) *score* of a vertex $x \in V$ is

$$\sigma^+(x) = \sum_{xy \in A} \epsilon(xy), \quad (\sigma^-(x) = \sum_{yx \in A} \epsilon(yx), \text{resp.})$$

A one-to-one mapping α from V onto $\{1, 2, ..., |V|\}$ is called a *ranking* of D. For $\alpha(x) = i$, $\alpha^{-1}(i) = x$. Sometimes, we shall determine a ranking α by the string $(\alpha^{-1}(1), ..., \alpha^{-1}(|V|))$. For a subset X of V, a ranking α of D induces the following permutation $\alpha|_X$: for a vertex $x \in X$, $\alpha|_X(x) = |\{y \in X : \alpha(y) \leq \alpha(x)\}|$. Let x and y be two vertices in D and let α be a ranking of D. Then α_{xy} denotes a ranking of D as follows: $\alpha_{xy}(z) = \alpha(z)$ for every $z \notin \{x, y\}$, and $\alpha_{xy}(x) = \alpha(y)$, $\alpha_{xy}(y) = \alpha(x)$.

For a ranking α, an arc $vu \in A$ is called *forward* if $\alpha(v) < \alpha(u)$. The *length* of an arc vu is $\epsilon(vu)|\alpha(v) - \alpha(u)|$. The *forward length* $f_D(\alpha)$ of α is the sum of the lengths of all forward arcs. A ranking α is *forward optimal* if

$f_D(\alpha)$ is maximum. The set of all forward optimal rankings of D is denoted by $FOR(D)$. The main objective is to calculate the *proper forward rank* of every vertex x of D, i.e.

$$\pi(x) = \frac{1}{|FOR(D)|} \sum_{\alpha \in FOR(D)} \alpha(x).$$

Let $D = (V, A)$ be a multipartite PCD and let α be a ranking of D. Then, for a vertex $x \in V$, we define $\psi^-(\alpha, x) = \sigma^-(x) + |\{y \in U : \ \alpha(y) > \alpha(x)\}|$ and $\psi^+(\alpha, x) = \sigma^+(x) + |\{y \in U : \ \alpha(y) < \alpha(x)\}|$, where U is the colour class of D containing x.

3 NP-hardness

Theorem 3.1 *The problem of finding a forward optimal ranking of a PCD is NP-hard.*

Proof: This proof is similar to that of Theorem 4 in [9] but we shall use the following problem instead of the optimal linear arrangement problem (OLAP), see [2], p.200.

Maximum linear arrangement problem (MLAP).
Instance: Graph $G = (V, E(G))$ and positive integer k.
Question: Is there a ranking α so that

$$\sum_{\{x,y\} \in E(G)} |\alpha(x) - \alpha(y)| \geq k.$$

(If we replace $\geq$ by $\leq$ we get the OLAP.)

The MLAP is NP-complete since the OLAP is NP-complete and since

$$\sum_{\{x,y\} \in E(G)} |\alpha(x) - \alpha(y)| + \sum_{\{x,y\} \in E(\overline{G})} |\alpha(x) - \alpha(y)|$$

is a constant depending only on the number of vertices in G ($\overline{G}$ is the complement of G.)

Let $G = (V, E)$ be a graph and let $D = (V, A)$ be the symmetric digraph corresponding to G, i.e. $A = \{xy, yx : \{x, y\} \in E\}$. Let also $\epsilon(xy) = 0.5$ for every $xy \in A$. Then

$$\sum_{\{x,y\}\in E} |\alpha(x) - \alpha(y)| = 2f_D(\alpha).$$

Hence, the MLAP is polynomial reducible to the problem of finding an optimal forward ranking of a PCD. $\square$.

4 Lemmas

Lemma 4.1 *[7] Let $K = (V, A)$ be a complete PCD with n vertices, and let α be a ranking of K. Then*

$$f_K(\alpha) = \sum_{x\in V} \sigma^-(x)\alpha(x) - \frac{1}{6}n(n^2 - 1) = \frac{1}{3}n(n^2 - 1) - \sum_{x\in V} \sigma^+(x)\alpha(x). \quad (1)$$

In the sequel, let $D = (V, A)$ be a complete multipartite PCD with n vertices contained in r colour classes $V_1, ..., V_r$.

The result of Lemma 4.2 (involving $\psi^-(\alpha, x)$ only) was proved in [6]. Note that our proof is shorter.

Lemma 4.2 *Let α be a ranking of D. Then*

$$f_D(\alpha) = \sum_{x\in V} \psi^-(\alpha, x)\alpha(x) - \frac{1}{6}n(n^2-1) = \frac{1}{3}n(n^2-1) - \sum_{x\in V} \psi^+(\alpha, x)\alpha(x). \quad (2)$$

Proof: For every colour class U of D, add the set of arcs $\{vw : v, w \in U, \alpha(w) < \alpha(v)\}$ (all of weight one) to A. The new PCD H is complete. Note that the negative (positive, resp.) score of a vertex x in H equals $\psi^-(\alpha, x)$ ($\psi^+(\alpha, x)$, resp.). Now (2) follows from (1) and an obvious fact that $f_D(\alpha) = f_H(\alpha)$. $\square$.

Lemma 4.3 *Let α be a forward optimal ranking and β be a ranking of D, and let x and y be vertices in the same colour class of D. Then*

(i) $f_D(\beta_{xy}) - f_D(\beta) = m(\sigma^+(y) - \sigma^+(x))$, where $m = \alpha(y) - \alpha(x)$;

(ii) if $\sigma^+(x) = \sigma^+(y)$, then $\alpha_{xy} \in FOR(D)$ as well, and in particular, $\pi(x) = \pi(y)$;

(iii) $\alpha(x) < \alpha(y)$ implies $\sigma^+(x) \geq \sigma^+(y)$.

Proof: (i) can be proved by (2), and (ii) and (iii) are easy consequences of (i). □.

5 Forward optimal rankings

Lemma 4.3 (iii) provides a unique 'optimal' ranking of the vertices in any colour class of D up to the vertices having the same positive score. Now we wish to fix completely the order of vertices having the same positive score and contained in the same colour class of D. To do that we consider a collection $\Lambda = \{\lambda_1, ..., \lambda_r\}$ of rankings of the colour classes of D, where λ_i is a ranking of V_i, such that

$$\sigma^+(\lambda_i^{-1}(1)) \geq \sigma^+(\lambda_i^{-1}(2)) \geq ... \geq \sigma^+(\lambda_i^{-1}(|V_i|)), \tag{3}$$

for every $i = 1, ..., r$. We wish to find all forward optimal rankings α so that $\alpha|_{V_i} = \lambda_i$ for each $i = 1, ..., r$. We call such rankings Λ-*optimal.* By Lemma 4.3, Λ-optimal rankings exist for every Λ satisfying (3).

Fix a collection Λ satisfying (3). Then, for every Λ-optimal ranking α of D, $\psi^+(\alpha, x)$ does not depend on α. Hence, we can define $\omega(x) = \psi^+(\alpha, x)$, for a Λ-optimal ranking α and a vertex x of D. By (2), the problem to find all Λ-optimal rankings of D is equivalent to the following problem. Find all rankings α of D which provide minimum to the function

$$F(D, \alpha) = \sum_{x \in V} \omega(x)\alpha(x)$$

subject to the constraints

$$\alpha|_{V_i} = \lambda_i, \text{ for every } i = 1, ..., r. \tag{4}$$

Consider the following single machine job sequencing problem (for basic terminology on the theory of scheduling see, e.g., [3].) Let us be given n jobs which, for simplicity, are the vertices of D and a collection Λ satisfying (3). Let $\omega(x)$ be the weight and $p(x)$ be the processing time of a job (vertex) x. Then, the total weighted completion time $C(D,\alpha)$, when the jobs are sequenced according to a permutation α, is

$$C(D,\alpha) = \sum_{x\in V} \omega(x) \sum \{p(y) : \ \alpha(y) \leq \alpha(x)\}.$$

We wish to minimize $C(D,\alpha)$ subject to the constraints (4). But these constraints are just the "parallel chains" constraints [10]. Horn [5] was the first to propose an algorithm for finding an optimal permutation, i.e. a permutation providing the minimum total completion time subject to the "parallel chains" constraints (he has really considered more general forest-like constraints.) Obviously, the job sequencing problem above, when all $p(x) = 1$, is equivalent to the problem of finding Λ-optimal rankings of D.

We shall deal with a modification of Horn's algorithm due to Sidney (see Algorithm 2 in [10].) Sidney proved (Theorem 9 and Lemma 14 in [10]) that a permutation is optimal if and only if it can be generated by his algorithm. Below we describe Sidney's algorithm adopted to our problem.

Let $\lambda_i = (v_1^{(i)}, v_2^{(i)}, ..., v_{n_i}^{(i)})$ ($n_i = |V_i|$) for every $i = 1, ..., r$. For an index $i \in \{1, ..., r\}$ and a pair j, k so that $1 \leq j \leq k \leq n_i$, define the set $S_{jk}^{(i)} = \{v_j^{(i)}, v_{j+1}^{(i)}, ..., v_k^{(i)}\}$ and its average of ω: $\rho(S_{jk}^{(i)}) = (k-j+1)^{-1} \sum_{m=j}^{k} \omega(v_m^{(i)})$. For $j \in \{1, ..., n_i\}$, a set $S_{jk}^{(i)}$ is called ρ^*-*maximal* if, for every $\ell \in \{j, j+1, ..., n_i\}$, $\rho(S_{jk}^{(i)}) \geq \rho(S_{j\ell}^{(i)})$ and, for every $t \in \{j, j+1, ..., k-1\}$, $\rho(S_{jk}^{(i)}) > \rho(S_{jt}^{(i)})$. We shall denote such a ρ^*-maximal set by $S_j^{(i)}$ and its average of ω by $\rho_j^{(i)}$.

Algorithm 1.

1. Choose a collection $\Lambda = \{\lambda_1, ..., \lambda_r\}$ satisfying (3). Call $\lambda_i = (v_1^{(i)}, v_2^{(i)}, ..., v_{n_i}^{(i)})$ the i'th *chain*.
2. The initial current permutation α is the empty permutation. For every $i = 1, ..., r$, the current first vertex on the i'th chain has index $m_i = 1$ and set $S_{n_i+1}^{(i)} = \emptyset$ and $\rho_{n_i+1}^{(i)} = -\infty$.
3. For every $i = 1, ..., r$ and $j = 1, ..., n_i$, compute $S_j^{(i)}$ and $\rho_j^{(i)}$.

4. Choose an index i so that $\rho^{(i)}_{m_i} = \max_{1\le t\le r} \rho^{(t)}_{m_t}$. Let $S^{(i)}_{m_i} = S^{(i)}_{m_i,k}$. Append the elements of $S^{(i)}_{m_i}$, in their natural order, to α. Set $m_i = k+1$.

5. If all $S^{(i)}_j$ are empty, stop; α is optimal. Otherwise, return to Step 4.

By the discussion above we obtain the following characterization of forward optimal rankings.

Theorem 5.1 *A ranking α is forward optimal if and only if it can be generated by Algorithm 1.*

To illustrate the last theorem consider the complete multipartite PCD H treated in [7], Fig.4. The PCD H has colour classes $V_1 = \{a, b\}$, $V_2 = \{c\}$ and $V_3 = \{d, e\}$. The positive scores are $\sigma^+(a) = 2.6,\ \sigma^+(b) = 1.3,\ \sigma^+(c) = 1.7,\ \sigma^+(d) = 1,\ \sigma^+(e) = 1.4$. The only collection Λ satisfying (3) is $\Lambda = \{(a, b), (c), (e, d)\}$. Hence, we obtain $\omega(a) = 2.6, \omega(b) = 2.3, \omega(c) = 1.7, \omega(e) = 1.4, \omega(d) = 2$. The ρ^*-maximal sets and their averages of ω are $S'_1 = \{a\}$, $\rho'_1 = 2.6$, $S'_2 = \{b\}$, $\rho'_2 = 2.3$, $S''_1 = \{c\}$, $\rho''_1 = 1.7$, $S'''_1 = \{e, d\}$, $\rho'''_1 = 1.7$, $S'''_2 = \{d\}$, $\rho'''_2 = 2$. Therefore, by Algorithm 1, $FOR(H) = \{(a, b, e, d, c),\ (a, b, c, e, d)\}$. This set coincides with that constructed in [7] using Theorem 4 of [7].

A fast implementation of Algorithm 1 is based on the following procedure (Procedure 2) for finding $S^{(i)}_j$ and $\rho^{(i)}_j$ (in Step 3) due to Horn [5].

For $j = n_i, n_i - 1, \ldots, 1$ (in that order) do the following 3 steps:

1. $F(j) = j + 1$ and $\rho^{(i)}_j = \omega(v^{(i)}_j)$

2. While $F(j) \le n_i$ and $\rho^{(i)}_j < \rho^{(i)}_{F(j)}$ do the following:

$$\begin{aligned}
\rho^{(i)}_j &= \frac{\rho^{(i)}_j (F(j) - j) + \rho^{(i)}_{F(j)} (F(F(j)) - F(j))}{F(F(j)) - j} \\
F(j) &= F(F(j))
\end{aligned}$$

3. $S^{(i)}_j = \{v^{(i)}_j, v^{(i)}_{j+1}, \ldots, v^{(i)}_{F(j)-1}\}$

Lemma 5.2 *Procedure 2 computes all* $S_j^{(i)}$ *and* $\rho_j^{(i)}$ *(*$i = 1, 2, \ldots, r \quad j = 1, 2, \ldots, n_i$*) in time* $O(n)$.

Proof: Define a potential function $\Phi(j)$ as follows. $\Phi(j)$ is the minimum integer so that $F^{\Phi(j)+1}(j) = n_i + 1$ (e.g. $\Phi(n_i) = 0$ since $F(n_i) = n_i + 1$). Clearly all $\Phi(j) \geq 0$, when $j = 1, 2, \ldots, n_i$. Let $T(j)$ be the total time used in the procedure, up to and including the point when we have computed $S_j^{(i)}$, where a performance of Step 1 takes a total of one unit of time, and each iteration of the while-loop (i.e. computing $\rho_j^{(i)}$ and $F(j)$) takes one unit of time. Step 1 sets $T(n_i) = 1$, $T(j) = T(j+1) + 1$, $\Phi(n_i) = 0$ and $\Phi(j) = \Phi(j+1) + 1$ when $j = 1, 2, \ldots, n_i - 1$. Each time one goes down into the while-loop it decreases $\Phi(j)$ by one, but increases $T(j)$ by one. This means that $T(j) + \Phi(j) = 2(n_i - j) + 1$. Since $\Phi(1) \geq 0$, we get that $T(1) < 2n_i$, which provides the desired complexity. $\Box$.

By Lemma 5.2 Steps 1, 2 and 3 can be completed in $O(n)$ time. Now by using an appropriate priority queue, such as a heap (see [11]), we can perform Steps 4 and 5 in $O(log\ r)$ time per iteration. Since there are at most n iterations of Steps 4 and 5, we obtain the following:

Theorem 5.3 *Algorithm 1 runs in time* $O(n\ log\, n)$.

Algorithm 1 can be easily modified to an algorithm for generating all optimal forward rankings. It is less trivial, but still not difficult to obtain an algorithm for computing proper forward ranks of the vertices of D. By Lemma 4.3 (ii), we may fix a collection Λ satisfying (3) in the beginning of the algorithm and, then, in the end recalculate the average rank of any vertex as the mean of the ranks of all vertices from the same colour class having the same positive score. The only non-trivial problem which arises here is how to find the proper forward ranks of the vertices in ρ^*-maximal sets with the same average of ω.

It is easy to see that, in order to solve the problem above, it is sufficient to solve the following auxiliary problem. Let $P = \{P_1, P_2, \ldots, P_r\}$ be a collection of sequences $P_i = (R_1^{(i)}, R_2^{(i)}, \ldots, R_{q_i}^{(i)})$, where $R_k^{(i)}$ are disjoint sets of V_i and $i = 1, 2, \ldots, r$; $q_1, q_2, \ldots, q_r \geq 1$. A ranking, β, of all the sets $R_k^{(i)}$ is *feasible* if, for every $i = 1, 2, ..., r$, $1 \leq s < j \leq q_i$ implies that $\beta(R_s^{(i)}) < \beta(R_j^{(i)})$. Let F_P denote the set of all feasible rankings of the sets in P. We wish to find $E_P(k,i)$, the average number of all vertices in front of $R_k^{(i)}$ taken over all feasible rankings of P, i.e.

$$E_P(k,i) = \frac{1}{|F_P|} \sum_{\beta \in F_P} \sum_{\beta(X) < \beta(R_k^{(i)})} |X|$$

For each $i = 1, 2, ..., r$ and $k = 1, 2, \ldots, q_i$, $E_P(k,i)$ can be computed as follows.

$$E_P(k,i) = \sum_{k'=1}^{k-1} |R_{k'}^{(i)}| + \sum_{\substack{i'=1 \\ i' \neq i}}^{r} \sum_{k'=0}^{q_{i'}} Q_P(k,k',q_i,q_{i'}) \sum_{k''=1}^{k'} |R_{k''}^{(i')}|,$$

$$\text{where} \qquad Q_P(k,k',q_i,q_{i'}) = \frac{\binom{k+k'-1}{k'}\binom{q_i+q_{i'}-k-k'}{q_{i'}-k'}}{\binom{q_i+q_{i'}}{q_{i'}}}$$

To show that the formula above is correct we just notice that $Q_P(k,k',q_i,q_{i'})$ is the probability that a randomly chosen $\beta \in F_P$ will have $\beta(R_{k'}^{(i')}) < \beta(R_k^{(i)}) < \beta(R_{k'+1}^{(i')})$ where $\beta(R_0^{(i')}) = -\infty$ and $\beta(R_{q_{i'}+1}^{(i')}) = \infty$. Indeed, there are $\binom{k+k'-1}{k'}$ ways of permuting the first $k-1$ sets of P_i together with the first k' sets of $P_{i'}$, there are also $\binom{q_i+q_{i'}-k-k'}{q_{i'}-k'}$ ways of permuting the last $q_i - k$ sets of P_i together with the last $q_{i'} - k'$ sets of $P_{i'}$. The total number of permutations of the sets in P_i together with the sets in $P_{i'}$ is $\binom{q_i+q_{i'}}{q_{i'}}$. Obviously, the formula above leads to a polynomial algorithm for finding all $E_P(k,i)$.

In order to keep the paper in appropriate length we omit a detailed consideration of an algorithm for finding the proper forward ranks of the vertices in D.

6 Acknowledgments

The authors would like to thank N. Alon, G. Brightwell and M. Kano for some helpful correspondence, and both referees for some useful suggestions.

References

[1] G. Brightwell and P. Winkler, Counting linear extensions. *Order* **8** (1991) 225-242.

[2] M.R. Garey and D.S. Johnson, **Computers and Intractability: a Guide to the Theory of NP-Completeness**, Freeman, San Fransisco, 1979.

[3] R.L. Graham, E.L. Lawler, J.K. Lenstra and A.H.G. Rinnooy Kan, Optimization and approximation in deterministic sequencing and scheduling: a survey. *Annals Disc. Math.* **5** (1979) 287-326.

[4] G. Gutin, Determining the ranks of vertices in a complete multipartite graph of paired comparisons. *Automation and Remote Control no. 10* (1989) 139-147.

[5] W.A. Horn, Single machine job sequencing with treelike precedence ordering and linear delay penalties. *SIAM J. Appl. Math.* **23** (1972) 189-202.

[6] M. Kano, Ranking the vertices of an r-partite paired comparison digraph. *Disc. Appl. Math.* **17** (1987) 245-253.

[7] M. Kano and A. Sakamoto, Ranking the vertices of a weighted digraph using the length of forward arcs. *Networks* **13** (1983) 143-151.

[8] M. Kano and A. Sakamoto, Ranking the vertices of a paired comparison digraph with normal completeness theorems. *Bull. Faculty Engineering, Tokushima Univ.* **20** (1983) 119-128.

[9] M. Kano and A. Sakamoto, Ranking the vertices of a paired comparison digraph. *SIAM J. Alg. Disc. Meth.* **6** (1985) 79-92.

[10] J.B. Sidney, Decomposition algorithms for single-machine sequencing with precedence relations and deferral costs. *Operations Res.* **23** (1975) 283-298.

[11] J.W.J. Williams, Algorithm 232: Heapsort. *Comm. ACM* **7** (1964) 347-348.