

Assigning Papers to Referees

Naveen Garg* Telikepalli Kavitha† Amit Kumar* Kurt Mehlhorn‡
Julián Mestre‡§

Abstract

Refereed conferences require every submission to be reviewed by members of a program committee (PC) in charge of selecting the conference program. There are many software packages available to manage the review process. Typically, in a bidding phase PC members express their personal preferences by ranking the submissions. This information is used by the system to compute an assignment of the papers to referees (PC members).

We study the problem of *assigning papers to referees*. We propose to optimize a number of criteria that aim at achieving fairness among referees/papers. Some of these variants can be solved optimally in polynomial time, while others are NP-hard, in which case we design approximation algorithms. Experimental results strongly suggest that the assignments computed by our algorithms are considerably better than those computed by popular conference management software.

1 Introduction

In Computer Science, the preferred way of disseminating scientific articles is through refereed conferences. A program committee (PC) selects the papers to be presented at the conference and published in the conference proceedings from among the submissions to the conference. The most prestigious conferences have acceptance rates as low as 20% [4].

The main responsibility of the PC chair is to organize the review process, in particular, to decide which papers are assigned to which member of the PC. The PC chair typically bases her decision on input from the PC, her knowledge of submissions and PC members, or scores that are computed automatically from keywords provided by authors and PC members. From now on, we call PC members *referees*.

There are many software systems available that support the PC chair in her task; for example, EasyChair [29], HotCRP [18], Softconf [2], Linklings [1], CMT [8], and Websubrev [12]. Used in more than 1300 conferences in 2008 alone [30], EasyChair is currently the most popular conference management software. The system asks the referees to declare conflicts of interests and to rank the papers (for which the referee has no conflict of interest) into three classes: high interest, medium interest, and low interest. This process is called bidding. Based on this information, the system automatically computes an assignment that the PC chair can later review and modify accordingly. Creating an assignment from scratch by hand is normally not feasible since many conferences get in excess of 500 submissions [4].

We abstract from the scenario described above and assume that the input for the paper assignment problem is an edge-labelled bipartite graph $G = (V, E, v)$ where $V = R \cup P$, R

*Computer Science and Engineering. Indian Institute of Technology. New Delhi, India.

†Department of Computer Science and Automation. Indian Institute of Science. Bangalore, India.

‡Max-Planck-Institut für Informatik. Saarbrücken, 66123, Germany.

§Research supported by an Alexander von Humboldt Fellowship.

is the set of referees and P is the set of papers, and $(r, p) \in E$ if r has no conflict of interest w.r.t. paper p , and $v : E \mapsto \{1, \dots, \Delta\}$. We call $v(r, p)$ the *value* or *rank* of paper p to referee r (or the value of referee r to paper p) or the valuation of p by referee r . We use the convention that rank Δ is the most preferred option.

Let $M \subseteq E$ be an *assignment* of papers to referees. We use $\delta(x)$ to denote the edges incident on a node x of G , and $\delta_M(x)$ to denote the edges in M incident on x .

What are the desired properties for a good assignment?

Coverage: *Every paper should be reviewed a sufficient number of times.* We require that each paper is assigned to k referees, where k is a number defined by the PC chair. We note, however, that our algorithms can be adapted to the setting where each paper p has a range and the number of reviews for p must fall within this range.

Load Balance: *No referee should be overburdened with papers.* We assume that the load is equally shared among the referees, i.e., no referee has to review more than $h := \lceil k|P|/|R| \rceil$ papers and no less than $h - 1$. (Again, we note that this could be changed to any arbitrary range.) In order to ease the discussion, we add a fictitious paper d that requires $|R|h - |P|k$ reviews and is connected to all referees with an edge of rank Δ^1 . Thus, in the modified instance every referee is assigned exactly h papers.

Quality: *How does a referee choose between two assignments?* For any referee r an assignment M gives rise to a *signature* vector $\sigma_r(M) = (\sigma_{r,\Delta}(M), \dots, \sigma_{r,1}(M))$, where $\sigma_{r,i}(M) = |\{e \in \delta_M(r) \mid v(e) = i\}|$ is the number of rank i edges incident to r in M . We call $\sigma_r(M)$ the *signature* of M w.r.t. r or r 's signature under M . Referees will prefer certain assignments over others based on their signatures. Only one general principle can be stated concerning this preference relation: if two assignments M and T are such that $\sigma_r(M) \neq \sigma_r(T)$ and $\sum_{j=i}^{\Delta} \sigma_{r,j}(M) \geq \sum_{j=i}^{\Delta} \sigma_{r,j}(T)$ for all $\Delta \geq i \geq 1$ then r will clearly prefer M over T . Unfortunately, this is only a partial order over signatures. We will restrict ourselves to two kinds of preference relations:

Lexicographic: A referee r prefers M over T if $\sigma_r(M)$ follows $\sigma_r(T)$ in lexicographic order, i.e., there is an i such that $\sigma_{r,j}(M) = \sigma_{r,j}(T)$ for $j > i$ and $\sigma_{r,i}(M) > \sigma_{r,i}(T)$.

Weighted: There is an increasing weight function w that maps ranks to reals. The weight of the signature of referee r is then $w(\sigma_r(M)) = \sum_{1 \leq i \leq \Delta} w(i) \sigma_{r,i}(M)$. Hence, referee r prefers M over T if $w(\sigma_r(M)) > w(\sigma_r(T))$.

The weighted preference relation turns the signature vector into a single number by assigning a weight to each rank. The lexicographic preference relation is the limit case of the weighted relation for $w(i+1) \gg w(i)$ for all i . Both relations define strict weak orders on signatures.

Fairness: *No referee should benefit at the expense of others.* The preference relations over signatures capture the quality of an assignment for a single referee. What is the overall quality of an assignment? An assignment should be *fair*, i.e., treat the different referees (papers) in a fair manner. In order to understand fairness better, let us see an assignment that is *unfair*.

Define the weight of an assignment M as

$$w(M) = \sum_r w(\sigma_r(M)) = \sum_{(r,p) \in M} w(v(r,p)).$$

¹This choice will be justified later in Section 5. We note here, though, that this rank can be set to any other value in $\{1, \dots, \Delta\}$ without invalidating any of our results.

An optimum assignment would then be an assignment of maximum weight. In fact, Easy-chair’s automatic assignment feature computes a maximum weight assignment w.r.t. the weight function $w(i) = i$ [28]. A simple example shows that this need not be fair.

We have four papers and two referees and each paper is to be reviewed once. The valuations of the referees are identical; both referees prefer papers 1 and 2 over papers 3 and 4. More precisely, papers 1 and 2 are ranked 2 and papers 3 and 4 are ranked 1.

	p_1	p_2	p_3	p_4
r_1	2	2	1	1
r_2	2	2	1	1

Consider the following two assignments. Under the first assignment the first referee reviews papers 1 and 2, and the second referee reviews papers 3 and 4; under the second assignment the first referee reviews papers 1 and 3, and the second referee reviews papers 2 and 4. Under the maximum weight objective, the assignments are the same. However, *the second assignment is clearly fairer than the first*. Under the second assignment, both referees review one paper for which they expressed high interest and one paper for which they expressed low interest. Under the first assignment, the first referee reviews two papers for which he expressed high interest and the second referee has to review two papers for which he expressed low interest. The second assignment treats the referees evenly, the first assignment treats them unevenly and, as the second assignment shows, does so without need.

Our toy example is relevant for practical assignment problems. We should expect a certain correlation between the valuations of different referees. If referees completely agree in their valuations, all assignments have the same weight.

How can we model fairness? A PC is a group effort. Fairness means that none of the members profits at the expense of other members. Thus, particular attention must be paid to the referee that has the worst signature. A fair assignment should maximize the worst signature of any referee:

$$\max_M \min_r \sigma_r(M).$$

Here the maximization is over all assignments that guarantee coverage and load balance and for each fixed assignment M , minimization is over the referees. Signatures are compared using either the lexicographic or weighted preference order.

Let us now restrict attention to assignments that maximize the minimum signature of any referee. The referees that cannot be treated any better should be satisfied by any assignment in this set, because there is no way to treat them better. The other referees should be satisfied by restricting attention to this subset of assignments, because they maximize the fate of their worst-off colleague. Which assignment should we choose among this restricted set of assignments? We should try to maximize the minimum fate of those referees that are not bound to the minimum. Continuing in this way, we arrive at the *leximin* objective from Social Choice Theory [27, 22]. More precisely, for any assignment M , consider the sorted vector $sort(\sigma_{r_1}(M), \dots, \sigma_{r_{|R|}}(M))$ of referee signatures; $sort(\cdot)$ re-arranges the entries of the argument vector into non-decreasing order. A *leximin optimal assignment* maximizes the sorted vector of signatures, i.e., it achieves

$$\max_M sort(\sigma_{r_1}(M), \dots, \sigma_{r_{|R|}}(M)),$$

where the maximum compares vectors using the usual lexicographic order performing left-to-right pointwise comparisons using lexicographic or weighted preference order for individual signatures, and M ranges over the set of all valid assignments.

We study the problem of maximizing the leximin objective under lexicographic and weighted preferences. We show that both problems are NP-hard even when $\Delta = 3$. On

the positive side, for $\Delta = 2$ we show that both problems can be solved in polynomial time by establishing a connection to a variant of rank-maximal matchings [14, 20]. For larger values of Δ in the case of the weighted preference order we give approximation algorithms building upon ideas of Bezáková and Dani [5], and Shmoys and Tardos [25].

The rest of the paper is organized as follows. In Section 2, we deal with the case where $\Delta = 2$. In Section 3 we show NP-hardness for both variants when $\Delta \geq 3$. In Section 4 we design approximation algorithms for the weighted preference order and $\Delta \geq 3$. Finally, in Section 5 we report preliminary computational experiments. We ran our algorithms on a real-life instance from the 16th European Symposium on Algorithms (ESA) 2008. The experiments suggest that the assignments computed by our algorithms are considerably better than the maximum weight assignment currently used by the EasyChair system.

1.1 Related work

Different aspects of the paper assignment problem have been studied by researchers in different fields. In Artificial Intelligence, data mining techniques have been applied to the task of inferring goodness of match between a referee and a paper based on keyword analysis. In Theoretical Computer Science and Operations Research, combinatorial optimization tools have been used to produce “good” assignments. (See the survey of Wang *et al.* [31] for references on these two aspects.) In Economics, the topic of incentive-compatible mechanisms for allocating indivisible goods to a set of agents with ranked preferences has been studied extensively [3, 6, 7, 17, 26, 32].

These three aspects are largely orthogonal to each other. Clearly, the data mining aspect is unrelated to incentives and optimization issues; indeed, its output can be used as input for the optimization problem. Also the incentive-compatible aspect is largely independent from optimization considerations: The only mechanisms that are strategy-proof (referees cannot benefit from falsifying their preferences) and satisfy other reasonable assumptions are the so-called serial dictatorships [26] where agents choose objects on a first-come first-served basis.

This paper deals with the optimization aspect of the paper assignment problem. Previous work considered mostly optimizing global properties of the assignment, and the proposed algorithms are based on min-cost matching/flow [11, 13, 23], integer programming [15], or heuristics without provable guarantees [9, 21]. Fairness of the assignment was not considered as an objective. We take a different approach by arguing that fairness is captured by the leximin criterion. For the variants of the problem that are NP-hard, instead of heuristics, we resort to approximation algorithms with worst-case guarantees.

Throughout the paper we focus mainly on one side of the assignment, the referees. We note, however, that we could also look at the assignment from the point of view of the papers. Presumably, referees that value a certain paper highly will do a better job reviewing the paper. Under this assumption, it could make sense to optimize the assignment from the point of view of the papers. Our algorithms can be made to work in this setting simply by interchanging the roles of referees and papers. Finally, we note that one could also try to optimize the assignment simultaneously from both sides. In Section 4.2, we explore this option for weighted preferences and the max-min objective.

2 An exact algorithm for $\Delta = 2$

In this section we deal with the problem of finding a leximin optimal assignment under lexicographic preferences for the interesting case of $\Delta = 2$. We note that this also constitutes

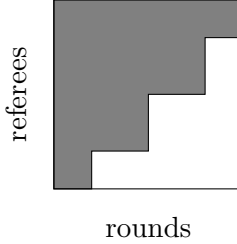


Figure 1: Visualizing an assignment when $\Delta = 2$. A staircase pattern divides the dark and light areas corresponding to papers with rank 2 and 1 respectively.

an optimal algorithm for the weighted preferences: Since w is increasing, the signature order is the same for both types of preferences when $\Delta = 2$.

Let $\widehat{\mathcal{Q}}$ be the set of assignments that obey coverage and load balance requirements; that is, every paper (except the dummy) is assigned to k referees and every referee is assigned h papers. Given an assignment $S \in \widehat{\mathcal{Q}}$, we define its *round decomposition* A_1, \dots, A_h as follows. Each referee $r \in R$ sorts the edges in $\delta_S(r)$ in non-decreasing value. Then A_i is constructed by taking the i th edge, in sorted order, from each referee.

A *rank-maximal assignment* is one that maximizes $\sigma(A_1)$ and subject to this, maximizes $\sigma(A_2)$, and so on. Here $\sigma(X)$ is the signature of the set $X \subseteq E$. Equivalently, we can ask that the concatenation of the signatures $\sigma(A_1), \sigma(A_2), \dots, \sigma(A_h)$ is lexicographically maximum. This objective is closely related the rank-maximal matchings of Irving *et al.* [14] and Mehlhorn and Michail [20], thus its name.

First we establish a connection between leximin optimal and rank-maximal assignments, and then we show how to compute the latter.

Lemma 1. *Let $S \in \widehat{\mathcal{Q}}$ be a rank-maximal assignment. If $\Delta = 2$ then S is leximin optimal under lexicographic preferences.*

Proof. Visualize S as a $\{1, 2\}^{|R| \times h}$ matrix. Each row corresponds to a referee and each column corresponds to a round A_i in the round decomposition of S ; rows are sorted in lexicographic order of their signature. The result is a staircase pattern with rank 2 entries on top of rank 1 entries as shown in Figure 1. This pattern is the same for all rank-maximal assignments.

Let $S' \in \widehat{\mathcal{Q}}$ a leximin optimal assignment and consider a similar matrix visualization for S' . If the staircase patterns of S and S' are the same then both assignments are rank-maximal and leximin optimal. Assume, for the sake of contradiction, that they are different. Let i be the first round in which they differ. If S has more 2's than S' in the i th round then S' is not leximin optimal. Likewise, if S' has more 2's than S in the i th round then S is not rank-maximal. A contradiction, thus the patterns are the same and the lemma follows. \square

A rank-maximal assignment can be found using an appropriate objective function over the round decomposition A_1, \dots, A_h . This is done by first translating ranks into costs: Rank $j \in \{1, \dots, \Delta\}$ translates into cost N^j , where $N = |R| + 1$. This choice guarantees that the gain of improving one edge, say from rank j to rank $j + 1$ offsets the loss of all other edges of rank j or less: The gain is $N^{j+1} - N^j = |R|N^j$ and the loss is at most $(|R| - 1)N^j$, so the gain exceeds the loss and the choice of costs guarantees rank-maximality of a single round. With this cost function, the maximum cost of a single round is less than $C = N^{\Delta+1}$.

Let A_1, A_2, \dots, A_h be a sequence of assignments. We assign cost

$$\sum_{1 \leq i \leq h} c(A_i)C^{h-i}$$

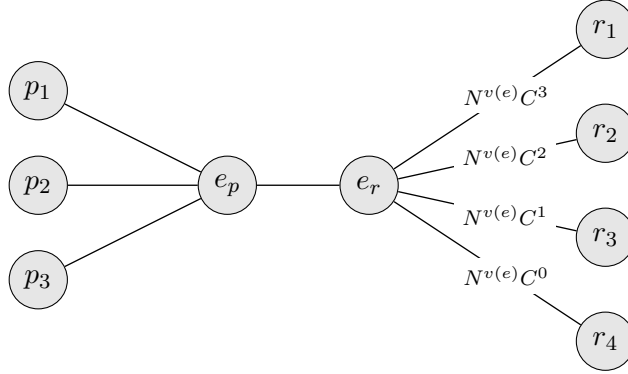


Figure 2: Edge construction for $e = (p, r)$ used in the reduction from rank-maximal assignment to maximum cost matching. In this case $k = 3$ and $h = 4$.

to it, where $c(A_i) = \sum_{(r,p) \in A_i} N^{v(r,p)}$. In other words, assigning a paper of rank j in round i contributes $N^j C^{h-i}$ to the objective value. This choice of costs guarantees that an increase in $c(A_i)$ offsets any decrease in subsequent assignments. Thus maximizing the cost of the round decomposition of some assignment $S \in \hat{\mathcal{Q}}$ guarantees that S is rank-maximal. The next theorem translates this insight into an actual algorithm.

Theorem 1. *A rank-maximal assignment can be computed in $O(\Delta h \sqrt{n'} m' \log n')$ time², where $n' = O(|E|)$ and $m' = O(|E|(k+h))$.*

Proof. We now show how to reduce this problem to finding a maximum cost perfect matching in a bipartite graph H . For every paper $p \in P - d$ we create k nodes p_1, \dots, p_k in H and as many copies as necessary for the dummy paper d ; for every referee $r \in R$ we create h nodes r_1, \dots, r_h in H ; finally, for every edge $e \in E$ we create two nodes e_p and e_r . This completes the vertex set of H . For every $e = (p, r) \in E$, we include edges in H connecting every p_i to e_p for all $i \in \{1, \dots, k\}$, every r_j to e_r for all $j \in \{1, \dots, h\}$, and e_p with e_r . The cost of the edge (r_j, e_r) is $N^{v(r,p)} C^{h-j}$ and all other costs are zero. Figure 2 depicts the edge gadget just described. We note that this construction is very similar to the standard gadget [24] used to reduce subgraph degree constrained problems to regular matching.

There is a clear correspondence between perfect matchings of H and round decompositions A_1, \dots, A_h of an assignment in $\hat{\mathcal{Q}}$: If (p_i, e_p) and (r_j, e_r) belong to the matching then e belongs to A_j . Furthermore, the cost of M equals the cost of A_1, \dots, A_h discussed above. Thus, finding a rank-maximal assignment reduces to finding a maximum cost perfect matching in H .

Let us conclude by discussing the implementation of the algorithm. To deal with the large costs used in H we use the scaling algorithm of Mehlhorn and Michail [20] for maximum cost bipartite matching with huge costs. The algorithm runs in $O(\sqrt{n'} m' \log(n' C^{h+1}))$ time, where $n' = |V[H]|$ and $m' = |E[H]|$. Recall that $C \leq n'^{\Delta+1}$, $n' = O(|E[G]|)$, and $m' = O(|E[G]|(h+k))$, which yields the desired time bound. \square

Putting Lemma 1 and Theorem 1 together we get an algorithm for finding an optimal leximin assignment under either preference order.

Corollary 1. *If $\Delta = 2$ then an optimal leximin assignment under each of the weighted and lexicographic preference orders can be found in polynomial time.*

²In practice, G is almost complete and $h > k$, so the running time simplifies to $O(\Delta h^2 |E|^{1.5} \log |E|)$.

We finish this section noting that similar techniques to those used in Theorem 1 can be used to obtain an assignment where the reverse signature for each round is minimized. Namely, we can compute an assignment minimizing the number of rank 1 edges in A_1 , and subject to this, minimize the number of rank 2 edges in A_1 , until reaching rank Δ ; subject to this, we minimize the number of rank 1 edges in A_2 , and so on. Notice that even though the assignment under this objective is rank-maximal for $\Delta = 2$, this is not the case for larger values of Δ .

3 Hardness

In this section we show that maximizing the signature of the worst-off referee is NP-hard for $\Delta \geq 3$ under both lexicographic and weighted preferences. Our proof uses a reduction from 3-dimensional matching (3DM) very similar to that used by Lenstra *et al.* [19] to show NP-hardness of scheduling jobs on unrelated machines.

An instance of 3DM is defined by three disjoint sets A , B , and C of n elements each and a set of triplets $T \subseteq A \times B \times C$. The problem is to decide whether there is a subset M of T with cardinality n such that $\cup_{(a,b,c) \in M} \{a, b, c\} = A \cup B \cup C$. 3DM is one of Karp's famous 21 NP-complete problems [16].

Given an instance (A, B, C, T) of 3DM, we construct an instance (G, v) of the paper assignment problem where each paper must be assigned once ($k = 1$) and each referee is assigned two papers ($h = 2$). For each triplet $t \in T$ we create a referee r_t . For each $a \in A$, let ℓ_a be the number of triplets in T containing a , i.e., the degree of a in T . For each $a \in A$ we create papers $l_a^1, \dots, l_a^{\ell_a-1}$ with rank 3, and papers $s_a^1, \dots, s_a^{\ell_a-1}$ with rank 1. For each $b \in B$ and $c \in C$ we create papers m_b and m_c with rank 2. For each triplet $t = (a, b, c) \in T$ we have edges (l_a^i, r_t) and (s_a^i, r_t) for each $i \in \{1, \dots, \ell_a - 1\}$, and edges (m_b, r_t) and (m_c, r_t) . Note that $2|R| = |P|$, so it is not necessary to introduce a dummy paper to achieve load balance.

Lemma 2. *Let (A, B, C, T) be an instance of 3DM and (G, v) be the instance of the paper assignment problem induced by the reduction described above. Then (A, B, C, T) has a perfect matching if and only if (G, v) admits an assignment where every referee gets either one rank 3 and one rank 1 paper, or two rank 2 papers.*

Proof. ([19]). Suppose that the 3DM instance has a perfect matching $N \subseteq T$. We show how to construct an assignment as described in the lemma statement. For each $t = (a, b, c) \in N$, we assign m_b and m_c to r_t so that the referee gets two rank 2 papers. For $a \in A$ there are $\ell_a - 1$ triplets $t \in T \setminus N$ containing a ; each of their corresponding referees can be assigned one of the $\ell_a - 1$ rank 3 papers and one of the $\ell_a - 1$ rank 1 papers.

Conversely, suppose that there is an assignment as described in the lemma statement. Let N be the set of triplets t such that r_t is not assigned a paper l_a^i . Note that for each $a \in A$ there is at least one such triplet. For each $t \in N$ the referee r_t is assigned papers m_b and m_c . This can only happen if N is a perfect matching. \square

This reduction can be used to prove hardness for both types of preferences.

Theorem 2. *Maximizing the signature of the worst-off referee is NP-hard for both lexicographic and weighted preferences for $\Delta \geq 3$.*

Proof. By Lemma 2 the instance of 3DM has a matching if and only if signature of the worst-off referee is at least $(0, 2, 0)$ under lexicographic preferences and at least 4 under weighted preferences when using $w(i) = i$. \square

4 An approximation algorithm for weighted preferences

In this section we shift our attention to weighted preferences. Recall that we are given a weight function $w : \{1, \dots, \Delta\} \rightarrow \mathbb{R}^+$ to map ranks to real numbers. For any edge e in the instance, we use the shorthand notation $w(e)$ to denote $w(v(e))$.

We will make frequent use of the polytope defined by the set of fractional assignments that obey coverage and load balance requirements:

$$\mathcal{Q} = \left\{ x \in [0, 1]^{|E|} \mid \begin{array}{l} x(\delta(p)) = k \quad \forall p \in P - d \\ x(\delta(r)) = h \quad \forall r \in R \end{array} \right\}$$

Here $\delta(u)$ denotes the set of edges incident on vertex u , and $x(S)$ is a short hand for $\sum_{e \in S} x(e)$. The variable $x(e)$ indicates whether e is chosen in the assignment: $x(e) = 1$ if e is chosen, and $x(e) = 0$ otherwise. The constraints enforce that every real paper is reviewed k times and that each referee is assigned h papers.

The constraint matrix defining \mathcal{Q} is totally unimodular. Therefore, the polytope is integral. We denote with $\widehat{\mathcal{Q}}$ the set of extreme points of \mathcal{Q} . Sometimes we abuse notation slightly and write $S \in \widehat{\mathcal{Q}}$ for $S \subseteq E$ meaning that the characteristic vector $x^S \in \{0, 1\}^{|E|}$ associated with S belongs to $\widehat{\mathcal{Q}}$.

Suppose we are given values I_r for each referee $r \in R$ and I_p for each paper $p \in P$. Our goal is to find an assignment where the total weight of papers assigned to each referee r is at least I_r and the weight of referees assigned to a paper p is at least I_p . The first objective aims at making the referees happy, while the second aims at improving the review process since, presumably, a referee that values a paper highly will do a better job than one who is not interested in the paper. The linear program for \mathcal{Q} can be extended imposing additional constraints on the total weight each referee and paper sees:

$$\mathcal{T} = \left\{ x \in \mathcal{Q} \mid \begin{array}{l} \sum_{e \in \delta(p)} w(e) x(e) \geq I_p \quad \forall p \in P - d \\ \sum_{e \in \delta(r)} w(e) x(e) \geq I_r \quad \forall r \in R \end{array} \right\}$$

We now present an algorithm that, given a fractional solution $x \in \mathcal{T}$, produces an integral assignment in $\widehat{\mathcal{Q}}$ with a small additive loss in the weights seen by each referee and each paper. Later we use this rounding procedure to approximate two different objectives.

4.1 Rounding a fractional assignment

Consider a particular referee r and let e_1^r, e_2^r, \dots be the edges incident on r sorted by decreasing weight. We break these edges into groups according to their fractional value in x . Some edges belong to two groups. An edge e_j^r belongs to the i th group of r if $x(\{e_1^r, \dots, e_{j-1}^r\}) < i$ and $i - 1 \leq x(\{e_1^r, \dots, e_{j-1}^r\})$. We use $A(r, i)$ to denote the edges in groups 1 through i of r . A grouping with respect to each paper is defined analogously. Based on these sets, we create a new assignment problem

$$\mathcal{U} = \left\{ y \in \mathcal{Q} \mid \begin{array}{l} y(A(p, i)) \geq i \quad \forall 1 \leq i \leq k \text{ and } p \in P - d \\ y(A(r, i)) \geq i \quad \forall 1 \leq i \leq h \text{ and } r \in R \end{array} \right\}$$

The constraints associated with the referees in \mathcal{U} and \mathcal{Q} form a laminar family: If we consider any two of these constraints and compare the set of variables they use, these sets are either disjoint or one is a subset of the other. Similarly, the constraints associated with the papers also form a laminar family. It is well known that a 0-1 matrix whose rows can be decomposed into two laminar families is totally unimodular. (For completeness we include a proof of this fact in Appendix A.) It follows that \mathcal{U} is integral.

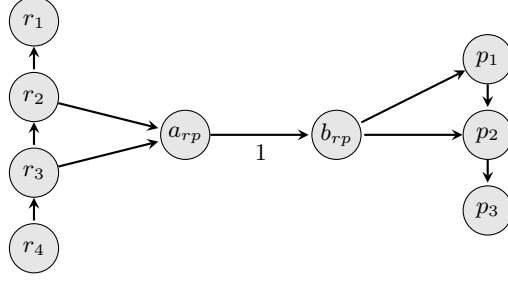


Figure 3: Edge (r, p) belongs to the second and third group with respect to r and the first and the second group with respect to p . The edge (a_{rp}, b_{rp}) has capacity one. All other edges have unbounded capacity.

By definition, we have that $x \in \mathcal{U}$. Hence, \mathcal{U} is not empty and, because it is integral, there must exist $\hat{y} \in \mathcal{U} \cap \{0, 1\}^{|E|}$. In fact, we can show that the problem of finding such a solution reduces to computing a max flow in a suitable graph.

Lemma 3. *An integral solution of \mathcal{U} can be computed using one max-flow computation.*

Proof. The edge gadget shown in Figure 3 is the basis of our max flow network. For each reviewer r we have nodes r_1, r_2, \dots, r_h and for each real paper p , we have nodes p_1, p_2, \dots, p_k (similarly, the dummy paper induces as many copies as necessary.) Each r_i has a supply of one unit of flow and each p_j has a demand of one unit of flow. Also we have edges of unbounded capacity from r_i to r_{i-1} for all i and from p_j to p_{j+1} for each j .

For each edge (r, p) , we have two auxiliary nodes a_{rp} and b_{rp} . There is an edge of capacity 1 from a_{rp} to b_{rp} . If (r, p) belongs to the i th group of r , we have an edge from r_i to a_{rp} . If (r, p) belongs to the j th group of p , we have an edge from b_{rp} to p_j . So a_{rp} has indegree one or two and b_{rp} has outdegree one or two. Figure 3 illustrates the construction.

For this network we construct a fractional feasible flow as follows: If $(r, p) = e_j^r$ and (r, p) belongs to the i th group of r , we send

$$\left| [i-1, i] \cap \left[\sum_{\ell < j} x(e_\ell^r), \sum_{\ell \leq j} x(e_\ell^r) \right] \right|$$

units of flow from r_i to a_{rp} . Similarly, if $(r, p) = e_{j'}^r$ and (r, p) belongs to the i' th group of p , we send

$$\left| [i'-1, i'] \cap \left[\sum_{\ell < j'} x(e_\ell^r), \sum_{\ell \leq j'} x(e_\ell^r) \right] \right|$$

units of flow from b_{rp} to $p_{j'}$. The flow from a_{rp} to b_{rp} thus equals x_{rp} .

Since the network is feasible, the Ford Fulkerson algorithm finds an integral flow. Note that an integral flow readily induces an integral assignment. Observe that in this integral flow, for all i , the flow out of $\{r_1, \dots, r_i\}$ is at least i , which means that the integral assignment selects at least i edges from $A(r, i)$; similarly, for all i , the flow into $\{p_1, \dots, p_j\}$ is at least j , which means the integral assignment selects at least j edges from $A(p, j)$. In other words, the integral assignment is a feasible solution of \mathcal{U} . \square

Now that we have established how to find an integral solution of \mathcal{U} , we shall prove that this is in fact a good approximation of x .

Lemma 4. *Let x be a fractional assignment in \mathcal{T} and \hat{y} be an integral assignment in \mathcal{U} , the polytope induced by x as defined above. Then for any node $u \in R \cup P - d$ we have*

- (i) $\sum_{e \in \delta(u)} w(e) \hat{y}(e) \geq I_u$ if $w(e_1^u) = w(e_{s_u}^u)$, and
(ii) $\sum_{e \in \delta(u)} w(e) \hat{y}(e) > I_u - (w(e_1^u) - w(e_{s_u}^u))$ otherwise.

Proof. For the sake of simplicity, we prove the lemma for some referee r , but the same argument works for papers. Referee r is assigned h edges under \hat{y} . Let f_1, \dots, f_h be these edges, sorted in non-increasing order of weight. Let $B(r, i)$ be the set of edges in the i th group of r . Notice that $v(f_i) \geq \min_{e \in A(r, i)} v(e) = \min_{e \in B(r, i)} v(e)$. It follows that

$$\begin{aligned} \sum_{e \in \delta(r)} w(e) \hat{y}(e) &= \sum_{i=1}^h w(f_i), \\ &\geq \sum_{i=1}^h \min_{e \in B(r, i)} w(e), \\ &\geq \sum_{i=1}^h \left[\sum_{e \in B(r, i)} w(e) x(e) - (1 - \epsilon) \left(\max_{e \in B(r, i)} w(e) - \min_{e \in B(r, i)} w(e) \right) \right], \end{aligned}$$

for some $\epsilon > 0$, and therefore

$$\begin{aligned} \sum_{e \in \delta(r)} w(e) \hat{y}(e) &\geq I_r - (1 - \epsilon) \sum_{i=1}^h \left(\max_{e \in B(r, i)} w(e) - \min_{e \in B(r, i)} w(e) \right), \\ &\geq I_r - (1 - \epsilon) (w(e_1^r) - w(e_{s_r}^r)). \end{aligned}$$

□

With Lemma 4 in hand, we show how to approximate two different objectives: Maximizing the minimum weight the papers and referees get in an assignment, and maximizing the leximin objective.

4.2 Max min assignments

We say a pair $(\lambda_P, \lambda_R) \in Z_0^+ \times Z_0^+$ is *feasible* if $\mathcal{T} \cap \{0, 1\}^{|E|} \neq \emptyset$ when setting $I_p = \lambda_P$ for all $p \in P$ and $I_r = \lambda_R$ for all $r \in R$; in addition, the pair is said to be Pareto optimal if it is not dominated by another feasible pair. The Pareto set Π of a given instance is simply the set of all Pareto optimal pairs. Note that maximizing the minimum weight is equivalent to finding the pair $(0, \lambda_R) \in \Pi$ maximizing λ_R . Thus, by Theorem 2, it is NP-hard to compute Π . We show how to approximate the Pareto set Π .

Theorem 3. *If $k \in O(1)$ and $\Delta \in O(1)$, an approximation $\tilde{\Pi}$ of Π can be computed in polynomial time such that for all $(\lambda_P, \lambda_R) \in \Pi$ there exists $(\lambda'_P, \lambda'_R) \in \tilde{\Pi}$ such that $\lambda'_P > \lambda_P - (w(\Delta) - w(1))$ and $\lambda'_R > \lambda_R - (w(\Delta) - w(1))$.*

Proof. For a fixed value λ_P , we write an LP based on \mathcal{T} so that $I_p = \lambda_P$ and $I_r = t$; the objective is to maximize t . Clearly, if there exists $(\lambda_P, \lambda_R) \in \Pi$ then we are guaranteed $t \geq \lambda_R$. We can round a solution from \mathcal{T} using Lemma 4, which yields a solution where each paper gets total weight strictly larger than $\lambda_P - (w(\Delta) - w(1))$ and each referee gets total weight strictly larger than $t - (w(\Delta) - w(1))$.

If $k \in O(1)$ and $\Delta \in O(1)$, we can try all possible values of λ_P . Letting $\tilde{\Pi}$ be the Pareto set of the assignments found in this manner finishes the proof. □

4.3 Leximin assignments

We consider the problem of finding a leximin optimal assignment under weighted preferences. Here we are only interested in optimizing the weight of one side of the assignment; for simplicity we focus on the referees, but the same applies to the papers. Our main result in this subsection is an algorithm for computing a leximin optimal *fractional* assignment.

Let $x \in \mathcal{Q}$ be a fractional assignment. We define $\text{sort}(x)$ to be the value vector $(t_1, t_2, \dots, t_{|R|})$ sorted in non-decreasing order of its value, where $t_r = \sum_{e \in \delta(r)} w(e)x(e)$ for each $r \in R$. We now show how to find a fractional solution $x^* \in \mathcal{Q}$ maximizing this quantity. This solution can be rounded using Lemma 4. If the weight vector defined by x^* is $(t_1^*, \dots, t_{|R|}^*)$ then we get a solution $\hat{y} \in \hat{\mathcal{Q}}$ with value at least $(t_1^* - (w(\Delta) - w(1)), \dots, t_{|R|}^* - (w(\Delta) - w(1)))$.

The optimal vector x^* can be computed through a sequence of LP computations. We maintain a set of *floating referees* F and call the remaining $R \setminus F$ *grounded referees*. Initially $F = R$. Each grounded referee r has associated a minimum value level I_r and we maintain the invariant that $I_r = t_r^*$ for all grounded referees. First, we solve the linear program

$$\begin{aligned}
 & \text{maximize } q && \text{(LP1)} \\
 \text{subject to} & && \\
 & \sum_{e \in \delta(r)} w(e)x(e) \geq I_r && \forall r \in R \setminus F \\
 & \sum_{e \in \delta(r)} w(e)x(e) \geq q && \forall r \in F \\
 & x \in \mathcal{Q} \\
 & t \geq 0
 \end{aligned}$$

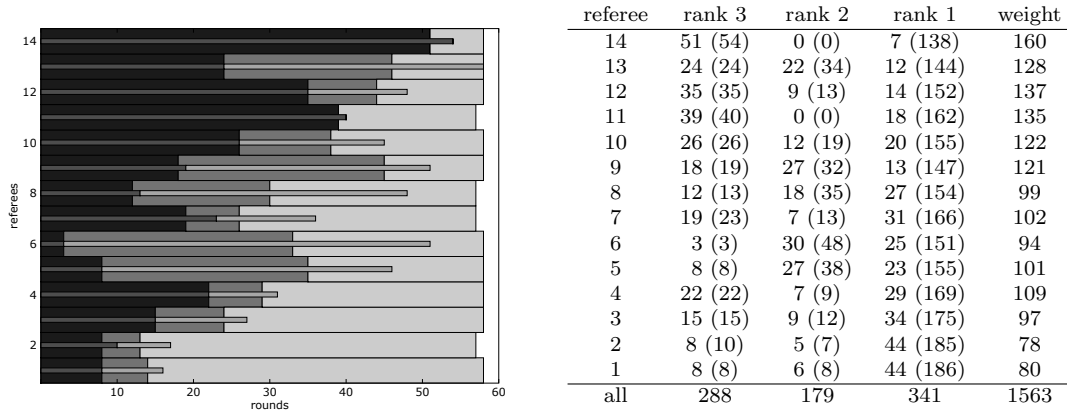
Let q^* be the optimal value. For each floating referee $r' \in F$, we solve another linear program

$$\begin{aligned}
 & \text{maximize } \sum_{e \in \delta(r')} w(e)x(e) && \text{(LP2)} \\
 \text{subject to} & && \\
 & \sum_{e \in \delta(r)} v(e)w(e) = I_r && \forall r \in R \setminus F \\
 & \sum_{e \in \delta(r)} v(e)w(e) \geq q^* && \forall r \in F - r' \\
 & x \in \mathcal{Q}
 \end{aligned}$$

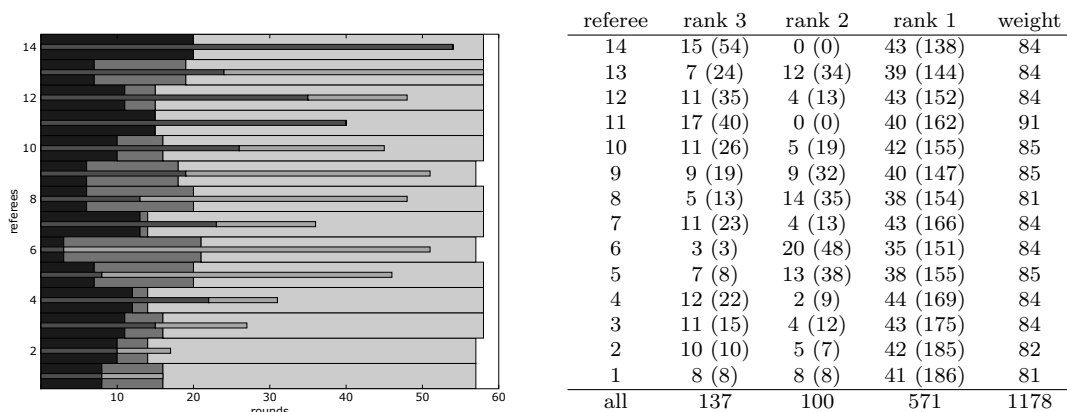
If the value of (LP2) is still q^* then we ground r' and set $I_{r'} = q^*$. Clearly, the new set of grounded papers maintains the invariant. Also, note at least one paper must be grounded on each iteration. Otherwise, taking the average of all the solutions found for (LP2) gives us a solution for (LP1) whose value is strictly larger than q^* , thus contradicting the optimality of q^* . Eventually the set of grounded papers equals R and by the invariant the value vector $(I_1, \dots, I_{|R|})$ is leximin optimal.

Theorem 4. *A leximin optimal fractional assignment can be computed in polynomial time.*

Once this fractional solution is computed, it can be rounded using the procedure from Section 4.1 to get an “approximate” leximin assignment.

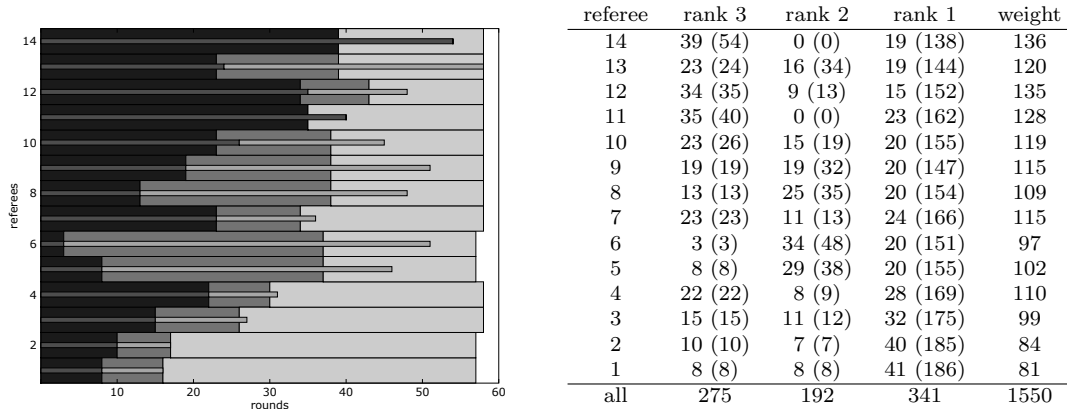


(a) Max-weight assignment

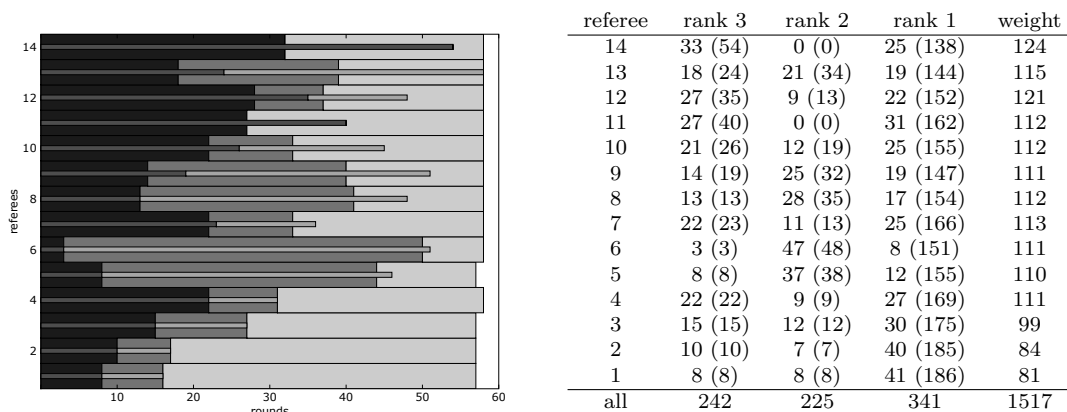


(b) Max-min weighted preferences

Figure 4: A comparison of the max-weight and the max-min assignments for the ESA 2008 instance ($|P| = 202$, $|R| = 14$, $k = 4$, $h = 58$, and $\Delta = 3$). Each subfigure shows a bar plot of the round decomposition of the assignment and a table with the signature of each referee under this assignment (and her valuation in parenthesis). Rows correspond to referees. Each row has one tall horizontal bar encoding the referee’s assignment and one short (height-wise) bar encoding the referee’s valuation for rank 3 and 2 papers (the remaining papers had rank 1 and are not shown in the graphs because of their large number.) Colors encode preferences levels: The darker the color, the higher the rank.



(a) Rank-maximal assignment



(b) Leximin weighted preferences

Figure 5: A comparison of the rank-maximal and the leximin assignments for the ESA 2008 instance ($|P| = 202$, $|R| = 14$, $k = 4$, $h = 58$, and $\Delta = 3$). Each subfigure shows a bar plot of the round decomposition of the assignment and a table with the signature of each referee under this assignment (and her valuation in parenthesis). Rows correspond to referees. Each row has one tall horizontal bar encoding the referee’s assignment and one short (height-wise) bar encoding the referee’s valuation for rank 3 and 2 papers (the remaining papers had rank 1 and are not shown in the graphs because of their large number.) Colors encode preferences levels: The darker the color, the higher the rank.

5 Experimental evaluation

In this section we investigate the performance of four algorithms: rank-maximal (Section 2), max-min (Section 4.2), leximin³ (Section 4.3), and max-weight (a maximum weight assignment obeying coverage and load-balance constraints).

EasyChair was used for the 16th European Symposium on Algorithms 2008. We ran the four algorithms on this instance and compared them to the maximum weight assignment currently used by the EasyChair system. For this instance $|P| = 202$, $|R| = 14$, $k = 4$, $h = 58$, and $\Delta = 3$. Therefore, 808 reviews are required, which means 10 referees will be assigned 58 papers and 4 referees will be assigned 57 papers. We use the weight function normally used by EasyChair: $w(i) = i$. Our results are shown in Figures 4 and 5. The referees are ordered by the total weight of their bid.

As one would expect, the max-weight assignment is not very fair⁴. Referee 14 expressed high interest for 55 papers, out of which more than 50 are assigned to him. On the other hand, referee 2 expressed a high interest for 10 papers and a medium interest for another 7 and has only 8 high interest and 5 medium interest papers assigned to him. The max-min assignment has low overall weight. This is because some referees rank very few papers and create a bottleneck beyond which the LP does not care to optimize. We do not consider it further.

The leximin assignment and the rank-maximal assignment are both fair and have good overall weight. This is not surprising for the leximin objective, as it is designed to guarantee fairness. The rank-maximal assignment guarantees fairness only for $\Delta = 2$; no guarantee is given for larger Δ . The instance uses $\Delta = 3$.

It is interesting to compare the assignments from a referee perspective. The leximin and the rank-maximal assignment completely satisfy the bids of referees 1 and 2. The bids of referees 3 and 4 are completely satisfied by the leximin assignment and by none of the other assignments. Referees 5 and 6 are also best treated by the leximin assignment. Referee 7 is treated best by the rank-maximal assignment with the leximin assignment coming close. Referee 8 would prefer the leximin assignment with the rank-maximal assignment coming close. Referee 9 would either prefer the max-weight or the rank-maximal assignment, it is not clear which. Referees 10 through 14 would opt for the max-weight assignment. In summary, referees who assigned ranks 2 and 3 to few papers prefer either the leximin or the rank-maximal assignment and referees who assigned these ranks to many papers prefer the max-weight assignment.

All assignments, except for the max-min assignment, use the same total number of rank 2 and rank 3 edges. However, ensuring fairness requires the use of fewer rank 3 edges. The leximin assignment uses the weight functions $w(i) = i$; making the weight difference more pronounced would shift the emphasis towards rank 3 edges.

The rank-maximal assignment is able to satisfy bids in the first 23 rounds. Starting in round 24, the coverage and load-balance constraints make it impossible to satisfy the bids.

Finally, we note that in our instance the load is not perfectly balanced: There are 4 referees that get one less paper than the rest. In the leximin assignment these 4 referees are the ones with the *worst assignment*. In an earlier implementation of our algorithms we gave the dummy paper a rank of 1, which had the opposite effect: The leximin assignment gave one less paper to the 4 referees with the *best assignment*. We feel that setting the rank of the dummy paper to Δ results in a more fair assignment.

³This is not a leximin optimal assignment, but the output of the rounding algorithm applied to the optimal leximin fractional assignment. For brevity's sake though we use the term leximin assignment. The same holds for the max-min assignment.

⁴Fair in the sense discussed in the introduction: No referee should benefit at the expense of others.

6 Concluding remarks

In this paper we have studied the problem of assigning papers to referees. We identified several desirable objectives for these assignments and designed efficient algorithms for them. Some variants can be solved optimally in polynomial time. In other cases, the problem is NP-hard and so we gave approximation algorithms.

Our next goal is to perform a thorough experimental evaluation of our algorithms and eventually incorporate them into conference management software such as EasyChair.

References

- [1] Linklings. <http://www.linklings.com/>.
- [2] Softconf. <http://www.softconf.com/>.
- [3] A. Alkan, G. Demange, and D. Gale. Fair allocation of indivisible goods and criteria of justice. *Econometrica*, 59(4):1023–1039, 1991.
- [4] P. Apers. Acceptance rates major database conferences. <http://wwwhome.cs.utwente.nl/~apers/rates.html>.
- [5] I. Bezáková and V. Dani. Allocating indivisible goods. *SIGecom Exchanges*, 5(3):11–18, 2005.
- [6] A. Bogomolnaia and H. Moulin. A new solution to the random assignment problem. *Journal of Economic Theory*, 100(2):295–328, 2001.
- [7] A. Bogomolnaia and H. Moulin. A simple random assignment problem with a unique solution. *Economic Theory*, 75(3):257–279, 2004.
- [8] S. Chaudhuri. Microsoft’s academic conference management service. <http://cmt.research.microsoft.com/cmt/>.
- [9] W. Cook, B. Golany, M. Kress, M. Penn, and T. Raviv. Optimal allocation of proposals to reviewers to facilitate effective ranking. *Management Science*, 51(4):655–661, 2005.
- [10] A. Ghouilla-Houri. Caractérisations des matrices totalment unimodulaires. *C. R. Acad. Sc. Paris*, 254:1192–1193, 1962.
- [11] J. Goldsmith and R. H. Sloan. The AI conference assignment problem. In *Proceedings of the Workshop on Preference Handling for Artificial Intelligence*, 2007.
- [12] S. Halevi. Websubrev. <http://people.csail.mit.edu/shaih/websubrev/>.
- [13] D. Hartvigsen, J. C. Wei, and R. Czuchlewski. The conference paper-reviewer assignment problem. *Decision Sciences*, 30(3):865–876, 1999.
- [14] R. W. Irving, T. Kavitha, K. Mehlhorn, D. Michail, and K. Paluch. Rank-maximal matchings. *ACM Transactions on Algorithms*, 2(4):602–610, 2006.
- [15] S. L. Janak, M. S. Taylor, C. A. Floudas, M. Burka, and T. J. Mountzizris. Novel and effective integer optimization approach for the NSF panel-assignment problem. *Industrial and Engineering Chemical Research*, 45(1):258–265, 2005.
- [16] R. M. Karp. Reducibility among combinatorial problems. In *Complexity of Computer Computations*, pages 85–103. Plenum Press, 1972.

- [17] A.-K. Katta and J. Sethuraman. A solution to the random assignment problem on the full preference domain. *Journal of Economic Theory*, 131(1):231–250, 2005.
- [18] E. Kohler. HotCRP. <http://www.cs.ucla.edu/~kohler/hotcrp/>.
- [19] J. K. Lenstra, D. B. Shmoys, and É. Tardos. Approximation algorithms for scheduling unrelated parallel machines. *Math. Program.*, 46:259–271, 1990.
- [20] K. Mehlhorn and D. Michail. Network problems with non-polynomial weights and applications. Manuscript.
- [21] J. J. Merelo-Gurvós and P. Castillo-Valdivieso. Conference paper assignment using a combined greedy/evolutionary algorithm. In *Proceedings of Parallel Problem Solving from Nature*, pages 602–611, 2004.
- [22] H. Moulin. *Fair division and collective welfare*. MIT Press, 2003.
- [23] R. O’Dell, M. Wattenhofer, and R. Wattenhofer. The paper assignment problem. Technical Report 491, ETH Zürich, 2005.
- [24] Y. Shiloach. Another look at the degree constrained subgraph problem. *Inf. Process. Lett.*, 12(2):89–92, 1981.
- [25] D. B. Shmoys and É. Tardos. An approximation algorithm for the generalized assignment problem. *Math. Program.*, 62:461–474, 1993.
- [26] L.-G. Svensson. Strategy-proof allocation of indivisible goods. *Social Welfare Choice*, 16(4):557–567, 1999.
- [27] B. Tungodden. Egalitarianism: Is leximin the only option? *Economics and Philosophy*, 16:229–245, 2000.
- [28] A. Voronkov. Personal communication.
- [29] A. Voronkov. Easychair. <http://www.easychair.org>.
- [30] A. Voronkov. Easychair - users. <http://www.easychair.org/users.cgi>.
- [31] F. Wang, B. Chen, and Z. Miao. A survey on reviewer assignment problem. In *Proceedings of the 21st international conference on Industrial, Engineering and Other Applications of Applied Intelligent Systems*, pages 718–727, 2008.
- [32] L. Zhou. On a conjecture by Gale about one-sided matching problems. *Journal of Economic Theory*, 52(1):123–135, 1990.

A Proof of the integrality of \mathcal{U}

Let A be the constraint matrix defining \mathcal{U} . A classical theorem of Ghouilla-Houri [10] states that A is totally unimodular if and only if every submatrix A' of A has an equitable coloring. An *equitable coloring* of a 0-1 matrix A' is a partition of its rows into red and blue rows such that in every column of A' , the number of blue 1’s and red 1’s differs by at most one. Recall that A' can be written as a block matrix

$$A' = \begin{bmatrix} B \\ C \end{bmatrix},$$

where the rows of B and C form two laminar families; that is, the set of 1's in any two rows is either disjoint or one is included in the other. The Hasse diagram of the "is included in" relation for the rows of B is a forest of rooted trees. We color the rows of B by alternating colors between adjacent levels of these trees starting with red for the roots. Using the Hasse diagram of C we color its rows in a similar way starting with blue for the root. Every column of A' gets the same number of red and blue 1's from rows in B , or one extra red. Likewise, every column gets the same number of red and blue 1's from rows in C , or one extra blue. In either case, the total number of red 1's and blue 1's differs by one. Thus, A' has an equitable coloring and A is totally unimodular.