

How to Assign Papers to Referees

Miki Hermann
hermann@lix.polytechnique.fr

Vincent Rudelli
rudelli@lix.polytechnique.fr

LIX, École Polytechnique, France

ABSTRACT

We present an algorithm to distribute a possibly large number of papers among a smaller number of referees, each paper requiring k reports. The optimality criterion for the assignment is not based on a local view of each referee, but on a global performance of the whole k -assignment satisfying a fairness criterion. The algorithm is based on an iterative application of maximum weighted matching. The iterative method is only a constant factor approximation algorithm.

1. INTRODUCTION

Consider the well-known task a chairperson of a conference is confronted with, to distribute the submitted papers among the members of the program committee. This task is not very difficult if the number of referees as well as of the submitted papers is not very large, provided that the chair knows well the competences of the program committee members. However, imagine a quite large conference with 60 program committee members and 700 submitted papers to referee, with the additional requirement to ask for 4 different reports per paper. Such a situation cannot be solved easily and the chair must take advantage of a conference management system. Although the existing conference management systems allow a certain degree of automatization, they do not solve the problem of paper assignment to the referees with an optimal algorithm. ConfMan does not include such a module, CyberChair offers only a rudimentary system working on keyword agreement, and EasyChair uses an assignment algorithm based on local search till it reaches a local maximum.

A k -assignment problem consists of distributing a possibly large number of papers among a smaller number of referees, each paper requiring k reports. Each referee expresses his or her preference towards each paper on a finite positive integer scale, going from 0, expressing a conflict of interest precluding any possibility to referee the particular paper, up to a certain predefined maximal value, expressing the highest pledge for refereeing it. Contrary to popular matchings [1], the optimality criterion is not based on a local view of each referee, but rather on a global *performance* of the whole k -assignment. However, the optimal k -assignment must also sat-

isfy a *fairness* criterion. Informally speaking, by fairness we mean that two referees making almost the same preference choice will get assigned papers with approximately the same preference values. Considered only statically, this constitutes a classical combinatorial optimization problem. It can be reformulated as the Hitchcock problem [6, section 7.4] and solved by a variant of the primal-dual method called the alphabeta algorithm. Nonetheless, this algorithm does not guarantee fairness of the assignment.

In this paper, we present a new algorithm assigning papers to referees, trying to maximize an objective function based on a global performance. It is implemented in the conference management system MyReview. It is based on an iterative application of a bipartite weighted matching algorithm. We will study the performance of the algorithm with respect to optimality.

Our algorithm and the considered problem have to be understood in a more general setting. This algorithm represent a solution to a multi-commodity multi-assignment problem with preferences under a fairness criterion. Such problems often arise in planning or in electronic commerce especially in connection with auctions. We decided to treat this problem on the most popular application.

2. PRELIMINARIES

Let $G = (R, S, E, w)$ be a weighted bipartite unoriented graph, where $R = \{r_1, \dots, r_m\}$ are the left-hand side vertices representing the *referees*, $S = \{s_1, \dots, s_n\}$ are the right-hand side vertices representing the submitted *papers*, $E \subseteq R \times S$ is the set of edges between referees and papers, and $w: E \rightarrow \mathbb{N}$ is the weight function representing the preferences of referees towards papers. Assume that there exists a total order $<$ on R , which is fulfilled in practice when the referees are lexicographically sorted by their names. Also assume that there exists a total order $<$ on S , which is fulfilled in practice when the papers are identified by numbers. For a given vertex $v \in R \cup S$ and a subset of edges $F \subseteq E$ we denote by $F(v)$ the set of edges in F incident to v and by $\deg_F(v)$ the *degree* of v in F . An edge $rs \in E$ between a referee $r \in R$ and a paper $s \in S$ with a null weight, i.e., $w(rs) = 0$ meaning a null preference, represents a conflict of interest for the referee r towards the paper s . We assume that all such edges have been removed from the graph G and the weight function $w: E \rightarrow \mathbb{N}^*$ is always positive for all edges in E . We extend the weight function to sets of edges by the overloading identity $w(F) = \sum_{e \in F} w(e)$ for every subset $F \subseteq E$.

Let also d be an integer parameter greater than the upper bound of w , i.e., $d > \max\{w(e) \mid e \in E\}$. It seems first that d should be variable and depend on a particular graph. In practice the differ-

ent possible values of w will be determined once and for all, since we cannot allow referees to choose whatever preferences they like. Thus, $\sup w$ will always be a value bounded by a constant integer, allowing us to determine a constant parameter d .

Instead of using the weight function w , each referee $r \in R$ could express the preferences for the papers S by means of an order \sqsubset_r . This would allow us to recompute the weight function W from the order \sqsubset_r of each referee $r \in R$ as the relative rank of each paper $s \in S$ in this order. However, the integer parameter d would not be a constant any more, but a function of the number of papers n . Of course, we can perform the opposite operation: given the weight function w we can recover the order \sqsubset_r of preferences for each referee $r \in R$, as we will see in the sequel.

A subset $M \subseteq E$ is a k -assignment for a bipartite weighted graph $G = (R, S, E, w)$ if for all papers $s \in S$ we have the identity $\deg_M(s) = k$. For each referee $r \in R$ we define a total order \sqsubset_r on the edges $E(r)$ asserting for all papers $s, s' \in S$ that $rs \sqsubset_r rs'$ holds if $w(rs) < w(rs')$ or $w(rs) = w(rs')$ and $s \preceq s'$ is satisfied. Let M be a k -assignment for G and let $r \in R$ be a referee. The set of papers $M(r)$ can be sorted by \sqsubset_r in a strictly decreasing order to form a tuple of length $l = \deg_M(r)$ denoted by $M[r] = (M_1(r), M_2(r), \dots, M_l(r))$ and satisfying the relation $M_1(r) \sqsupset_r M_2(r) \sqsupset_r \dots \sqsupset_r M_l(r)$. The set $M(r)$ and the vector $M[r]$ denote the same object, but $M[r]$ also bears the information about the relative preferences for the papers assigned to the referee r . We call $M[r]$ the *assignment order* for the referee r in M . Then we can write the k -assignment as the union $M = \bigcup_{r \in R} \{M_i(r) \mid 1 \leq i \leq \deg_M(r)\}$.

Let $M \subseteq E$ be a k -assignment for G . For any edge $e = rs \in M$ between $r \in R$ and $s \in S$ we define the *position* of e in M with respect to the paper s by the function $\kappa_M(e) = |\{r' \preceq r \mid r's \in M\}|$. For $e \notin M$ we set $\kappa_M(e) = 0$. Since $\deg_M(s) = k$ holds, we clearly have $1 \leq \kappa_M(e) \leq k$ for all edges $e \in M$. Moreover, for all papers $s \in S$ and all indices $j \in \{1, \dots, k\}$ there exists a referee $r \in R$ matching the paper s , i.e., such that $rs \in M$, and also $\kappa_M(rs) = j$. Since the set of referees R is totally ordered by \prec , the referee r is unique.

Let $M \subseteq E$ be a k -assignment for G . For any edge $e = rs \in M$ between $r \in R$ and $s \in S$ there exists a unique integer $i \in \{1, \dots, n\}$, satisfying the identity $e = M_i(r)$, called the *rank* of e within M . Given an edge e and a k -assignment M , we denote the rank of e within M by $\lambda_M(e)$. If $e \notin M$, we set $\lambda_M(e) = 0$. We know from the construction of $M[r]$ that $s' \in S$, $s' \neq s$, and $rs' \in M$ imply $\lambda_M(rs) \neq \lambda_M(rs')$. This means that two edges of M incident to the same referee r cannot have the same rank.

Let M be a k -assignment for G . The *performance* of a referee $r \in R$ within the assignment M is defined as

$$p(r, M) = \sum_{i=1}^{\deg_M(r)} w(M_i(r)) d^{n-i}$$

and the *global performance* of the assignment M is defined by $p^*(M) = \sum_{r \in R} p(r, M)$. A k -assignment M for a graph G is *maximal* if its performance $p^*(M)$ is maximal among all k -assignments. The *papers-to-referees k -assignment problem* is the task to find a maximal k -assignment for a bipartite weighted graph $G = (R, S, E, w)$. We say that the k -assignment problem for a graph G is *feasible* if there exists a k -assignment M for G . The latter is true

if and only if each paper from S has at least k incident edges in E . We always assume that the k -assignment problem for a graph G is feasible, i.e., that each paper $s \in S$ has at least k incident edges in E , and that G is not void, i.e., $R \neq \emptyset$, $S \neq \emptyset$, $E \neq \emptyset$, and $d > 1$.

3. FAIRNESS CRITERION

We want to maximize the global performance rather than simply the total weight because the performance also ensures a kind of a *fairness* in addition to taking weights in account. As we will show, the global performance indeed represents the fact that we had distributed papers almost uniformly among referees who expressed similar preferences, instead of assigning almost all of them to one referee just because he expressed higher preferences.

The assignment order $M[r]$ does not necessarily represent the real order how the papers were assigned to the referee r in the k -assignment M , but rather his assigned papers sorted in decreasing order with respect to the relative preferences of the referee r . It will help us to establish the fairness of the global performance.

Fairness Conditions: The applied fairness criterion must satisfy the following conditions:

- (A) The papers, for which different referees declared high preferences, must be almost equally distributed among these referees.
- (B) The papers must be distributed almost uniformly among referees. The maximal difference between the cardinalities of the sets of assigned papers to two different referees must be minimized, i.e., the value $\max_{r, r' \in R} ||M(r)| - |M(r')||$ must be minimal.
- (C) The criterion must be independent of arbitrary numbering of papers and referees.

Let $G = (R, S, E, w)$ be a weighted bipartite graph. Let M and M' be two k -assignments for G differing only in two consecutive positions j and $j+1$ for two referees $r_1, r_2 \in R$.

$$\begin{aligned} M_j(r_1) &= r_1 s_1, & M_{j+1}(r_1) &= r_1 s_2, \\ M_j(r_2) &= r_2 s_3, & M_{j+1}(r_2) &= r_2 s_4, \end{aligned}$$

and

$$\begin{aligned} M'_j(r_1) &= r_1 s_1, & M'_{j+1}(r_1) &= r_1 s_3, \\ M'_j(r_2) &= r_2 s_2, & M'_{j+1}(r_2) &= r_2 s_4 \end{aligned}$$

Since $M_j(r_i) \sqsupset_{r_i} M_{j+1}(r_i)$ must hold for each $i = 1, 2$, the weights must satisfy the relations $w(M_j(r_i)) \geq w(M_{j+1}(r_i))$ and $w(M'_j(r_i)) \geq w(M'_{j+1}(r_i))$, which is better expressed by means of the following inequalities:

$$\begin{aligned} w(r_1 s_1) &\geq w(r_1 s_2), & w(r_1 s_1) &\geq w(r_1 s_3), \\ w(r_2 s_2) &\geq w(r_2 s_4), & w(r_2 s_3) &\geq w(r_2 s_4). \end{aligned}$$

Assume that $w(r_1 s_2) > w(r_1 s_3)$ and $w(r_2 s_2) > w(r_2 s_3)$ hold. Then it is clear that M' is preferable to M , since it better satisfies the fairness condition (A). Indeed, we have

$$(w(M'_j(r_2)) - w(M_j(r_2)))d > w(M_{j+1}(r_1)) - w(M'_{j+1}(r_1))$$

since $d > \max\{w(e) \mid e \in E\}$, $w(r_1 s_2) - w(r_1 s_3) > 0$, and $w(r_2 s_2) - w(r_2 s_3) > 0$ hold. This implies the relation

$$\begin{aligned} w(M'_j(r_2))d^{n-j} + w(M'_{j+1}(r_1))d^{n-j-1} \\ > w(M_j(r_2))d^{n-j} + w(M_{j+1}(r_1))d^{n-j-1} \end{aligned}$$

leading to $p(r_1, M') + p(r_2, M') > p(r_1, M) + p(r_2, M)$, since $M_j(r_1) = M'_j(r_1)$ and $M_{j+1}(r_2) = M'_{j+1}(r_2)$ hold. Since $p(r, M) = p(r, M')$ holds for all referees $r \in R \setminus \{r_1, r_2\}$, we conclude that $p^*(M') > p^*(M)$ is satisfied.

Moreover, note that there is no assumption on the relation between weights allocated to the same paper by different referees. Indeed, we did not assume anything between the weights $w(M_j(r_1))$ and $w(M_j(r_2))$, or between $w(M'_j(r_1))$ and $w(M'_j(r_2))$, respectively for each $j = 1, \dots, \deg_M(r_i)$. Hence referee r_1 could have allocated high weights to the papers s_1, \dots, s_4 , compared to small values allocated to the same papers by referee r_2 . Nevertheless, an algorithm maximising the global performance will satisfy the fairness condition (A).

Now let N and N' be two k -assignments in the same weighted bipartite graph G , differing only in the number of papers assigned to two referees $r_1, r_2 \in R$.

$$\begin{aligned} \text{Let } |N[r_1]| &= j+2, & |N[r_2]| &= j, \\ |N'[r_1]| &= j+1, & |N'[r_2]| &= j+1. \end{aligned}$$

It is clear that N' is preferable to N , since it better satisfies the fairness condition (B). Indeed, we have

$$w(N'_{j+1}(r_2))d \geq d > w(N_{j+2}(r_1))$$

from $d > \max\{w(e) \mid e \in E\}$ and from the positive integer values of the weights in every assignment order for each referee. This implies

$$w(N'_{j+1}(r_2))d^{n-(j+1)} > w(N_{j+2}(r_1))d^{n-(j+2)}$$

leading to $p(r_1, N') + p(r_2, N') > p(r_1, N) + p(r_2, N)$ since $N_\ell(r_i) = N'_\ell(r_i)$ for $\ell = 1, \dots, j$ and $i = 1, 2$, as well as $N_{j+1}(r_1) = N'_{j+1}(r_1)$ hold. Since $p(r, N) = p(r, N')$ holds for all referees $r \in R \setminus \{r_1, r_2\}$, we conclude that $p^*(N') > p^*(N)$ is satisfied.

The global performance is independent of the numbering of papers and referees, since the assignment order depends on the order \prec on the papers S only if for a certain referee r the weights $w(rs)$ and $w(rs')$ are equal. This is defined in this manner only to break ties, but any other criterion ensuring a total order (e.g. a lexicographic order on papers considered as strings) would be sufficient. Hence the global performance also satisfies the fairness condition (C).

The fairness conditions are strongly relativised by the preferences $w(rs) = 0$ declared by a referee r for a paper s , which precludes any assignment of s to r . If there are too many declared zero preferences, the optimal result to the k -assignment problem can be biased and the fairness cannot be ensured any more, as it is shown in Example 1.. This is the reason why the word “almost” appears in the context of the fairness criterion.

Example 1 Consider a 1-assignment for the bipartite graph $G = (R, S, K_{2,2}, w)$ with the referees $R = \{r_1, r_2\}$, the papers $S = \{s_1, s_2\}$, and the weights w

$$\begin{pmatrix} 1 & 0 \\ 2 & 1 \end{pmatrix}$$

Algorithm 1 Iterative algorithm for k -assignment

Algorithm: ITERATIVE k -ASSIGNMENT

Input: A weighted bipartite graph $G = (R, S, E, w)$ and a positive integer bound k .

Output: A k -assignment for G .

Method:

```

1:  $M \leftarrow \emptyset$ 
2: while  $S \neq \emptyset$  do
3:   Find a maximum weighted matching  $M'$  in  $G$  among non-
     marked edges in  $E$ 
4:   Mark all edges contained in  $M'$ 
5:    $M \leftarrow M \cup M'$ 
6:    $S' \leftarrow \{s \in S \mid s \text{ has } k \text{ incident marked edges}\}$ 
7:    $E \leftarrow E \setminus \{rs \in E \mid s \in S'\}$ 
8:    $S \leftarrow S \setminus S'$ 
9: end while
10: return  $M$ 

```

with the parameter $d = 3$. The maximum 1-assignment, being the set $\{r_2 s_1, r_2 s_2\}$ with performance 7, is not fair. Indeed, it violates the fairness condition (B). The 1-assignment $\{r_1 s_1, r_2 s_2\}$ is fair with respect to the condition (B) but its performance is equal to 6. If the weight $w(r_1 s_2)$ were different from 0, a 1-assignment of the paper s_1 to referee r_2 and of the paper s_2 to referee r_1 would have a performance of at least 9.

4. ITERATIVE ALGORITHM

The iterative method for the k -assignment problem operates on the original bipartite graph $G = (R, S, E, w)$ repeatedly applying an algorithm for weighted bipartite matching. See Algorithm 1 for the implementation. The basic idea of this iterative method is somewhat similar to the method of successive approximations for minimum-cost flow problems presented in [4]. Notice that a k -assignment found through Algorithm 1 is not unique, for it may depend upon the choice of maximum weighted matchings. If the k -assignment problem is feasible, Algorithm 1 will eventually terminate and find a valid k -assignment, although not necessarily maximal. However, we will prove that the performance of a k -assignment found by the iterative method approximates the performance of the maximal k -assignment by a factor depending on both k and d .

Let us first show an example showing that the iterative method does not compute maximal k -assignments.

Example 2 We are looking for a 2-assignment for the graph $G = (R, S, K_{3,3}, w)$, with the referees $R = \{r_1, r_2, r_3\}$, the papers $S = \{s_1, s_2, s_3\}$ and with the weights w given by the following $|R| \times |S|$ matrix

$$\begin{pmatrix} 5 & 1 & 1 \\ 4 & 1 & 3 \\ 1 & 1 & 4 \end{pmatrix}$$

Choose $d = \max\{w(e) \mid e \in E\} + 1 = 6$. The maximum weighted matching in graph G is $N' = \{r_1 s_1, r_2 s_2, r_3 s_3\}$ with a total weight of 10. The maximum weighted matching in graph G with the previous matching N' removed is then the set $N'' = \{r_1 s_3, r_2 s_1, r_3 s_2\}$ with a total weight of 6. Algorithm 1 computes the 2-assignment $N = \{r_1 s_1, r_1 s_3, r_2 s_1, r_2 s_2, r_3 s_2, r_3 s_3\}$. The assignment orders are $N[r_1] = (s_1, s_3)$, $N[r_2] = (s_1, s_2)$, and $N[r_3] = (s_3, s_2)$. The global performance is $p^*(N) = 486$. However, we can easily see that the maximal 2-assignment is the set

$M = \{r_1 s_1, r_1 s_2, r_2 s_1, r_2 s_3, r_3 s_2, r_3 s_3\}$ with the assignment orders $M[r_1] = (s_1, s_2)$, $M[r_2] = (s_1, s_3)$, $M[r_3] = (s_3, s_2)$, and the global performance $p^*(M) = 498$.

The previous example shows that Algorithm 1 does not compute a maximal k -assignment. We will analyze how far the produced k -assignment can be from the optimum. To evaluate the approximation we need the following lemma.

Theorem 3 *Algorithm 1 is a constant factor approximation algorithm for the k -assignment problem with the factor $kd/(d-1)$.*

In the case of a variable parameter d instead of a constant one, we obtain a constant factor by replacing d with a lower bound of d . Recall that $d > 1$ and notice that $d/(d-1)$ is decreasing with respect to d .

When the parameter d becomes arbitrarily large, the approximating factor in Theorem 3 tends towards k . There exists bipartite graphs G with arbitrarily heavy weights on edges for which k is the right approximation factor, i.e., $\lim_{d \rightarrow \infty} p^*(N)/p^*(M) = 1/k$.

5. COMPLEXITY ANALYSIS

The algorithm is based on finding maximum weighted matchings in bipartite graphs. Let $\tau(i, j, \omega)$ denote the time complexity of finding a maximum weighted matching in a bipartite graph with i vertices, j edges and an upper bound on weights of ω . Recall that we consider $n = |S|$ papers in the weighted bipartite graph $G = (R, S, E, w)$. We also note $W = \sup w$. Since we have fewer referees than papers, the graph G has at most $2n$ vertices and at most n^2 edges.

In the iterative method, the most time consuming operation in the while-loop is Step 3 which requires to find a maximum weighted matching in a (sub)graph G . We also need the following proposition to evaluate the complexity of Algorithm 1.

Proposition 4 *The while-loop in Algorithm 1 will be repeated at most $n + k - 1$ times.*

The number of iterations of the while-loop in Algorithm 1 is $O(n)$ since we have $k < n$. Moreover, for all methods considered below, the complexity $\tau(i, j, \omega)$ is monotonically increasing with respect to i , j , and ω . Therefore Algorithm 1 can be performed in $O(n\tau(2n, n^2, W))$ time. Since all aforementioned complexities consist of a product of positive powers of the arguments or their logarithm, we can omit the constants. Hence, the iterative algorithm can be performed in $O(n\tau(n, n^2, W))$ time. The effective complexity of the algorithm depends on the method used to find maximum weighted matchings. We will compare the results using several different complexities, taken from a survey written by Schrijver [7].

Using the Hungarian method (see [6]), the maximum weighted matching can be computed in time $\tau(i, j, \omega) = O(i^4)$. Using this method, Algorithm 1 has a time complexity of $O(n^5)$.

The method of Dinits and Kronrod [2] computes the maximum weighted matching in time $\tau(i, j, \omega) = O(i^3)$. This more efficient method allows the iterative algorithm to be performed in $O(n^4)$ time.

The method of Gabow and Tarjan [3] computes the matching in time $\tau(i, j, \omega) = O(\sqrt{i} j \log(i\omega))$. With this newer method, the iterative algorithm has a time complexity of $O(n^{5/2} \log(nW))$. In this case when either d is a constant or it is dependent on n , we always have that $d = O(W)$. We can reasonably assume that $W = O(n)$. Finally, the algorithm presented in [5] computes the matching in time $\tau(i, j, \omega) = O(\sqrt{i} j \omega)$. This recent method is totally unadapted for the b -matching algorithm, since the resultant complexity of $O(n^4 W d^n)$ is exponential.

6. CONCLUDING REMARKS

We designed an algorithm computing a k -assignment for a weighted bipartite graph. It is based on an iterative application of the maximum weighted matching. The iterative method is only a constant factor approximation algorithm with factor $kd/(d-1)$. The iterative algorithm has already been integrated into the MyReview system.

7. REFERENCES

- [1] D. J. Abraham, R. W. Irving, T. Kavitha, and K. Mehlhorn. Popular matchings. In *Proceedings 16th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA 2005), Vancouver (British Columbia, Canada)*, pages 424–432, January 2005.
- [2] E. A. Dinits and M. A. Kronrod. An algorithm for solving the assignment problem. *Doklady Akademii Nauk SSSR*, 189(1):23–25, 1969.
- [3] H. N. Gabow and R. E. Tarjan. Faster scaling algorithms for network problems. *SIAM Journal on Computing*, 18(5):1013–1036, 1989.
- [4] A. V. Goldberg and R. E. Tarjan. Solving minimum-cost flow problems by successive approximation. In *Proceedings 19th Annual ACM Symposium on Theory of Computing (STOC'87), New York (New York, USA)*, pages 7–18. Association for Computing Machinery, 1987.
- [5] M.-Y. Kao, T. W. Lam, W.-K. Sung, and H.-F. Ting. A decomposition theorem for maximum weight bipartite matchings with applications to evolutionary trees. In J. Nešetřil, editor, *Proceedings 7th Annual European Symposium (ESA'99), Prague (Czech Republic)*, volume 1643 of *Lecture Notes in Computer Science*, pages 438–449. Springer, 1999.
- [6] Ch. H. Papadimitriou and K. Steiglitz. *Combinatorial optimization: Algorithms and complexity*. Dover Publications, 1998.
- [7] A. Schrijver. *Combinatorial Optimization — Polyhedra and Efficiency*, volume A: Paths, Flows, Matchings, chapter 17: Weighted bipartite matching and the assignment problem, pages 285–300. Springer, 2003.