I N F O R M A T I K

Rank-Maximal through Maximum Weight Matchings

Dimitrios Michail

MPI–I–2005–1–001 February 2005

FORSCHUNGSBERICHT RESEARCH REPORT

M A X - P L A N C K - I N S T I T U T
F Ü R
I N F O R M A T I K

Stuhlsatzenhausweg 85 66123 Saarbrücken Germany

Authors' Addresses

Dimitrios Michail
Max-Planck-Institut für Informatik
Stuhlsatzenhausweg 85
66123 Saarbrücken
michail@mpi-inf.mpg.de

Document revised: December 2005

Abstract

Given a bipartite graph $G(V, E)$, $V = A \dot{\cup} B$ where $|V| = n, |E| = m$ and a partition of the edge set into $r \leq m$ disjoint subsets $E = E_1 \dot{\cup} E_2 \dot{\cup} \ldots \dot{\cup} E_r$, which are called ranks, the *rank-maximal matching* problem is to find a matching M of G such that $|M \cap E_1|$ is maximized and given that $|M \cap E_1|$ is maximized, $|M \cap E_2|$ is also maximized, and so on. Such a problem arises as an optimization criteria over a possible assignment of a set of applicants to a set of posts. The matching represents the assignment and the ranks on the edges correspond to a ranking of the posts submitted by the applicants.

The rank-maximal matching problem has been previously studied where a $O(min(n + r, r\sqrt{n})m)$ time and linear space algorithm [1] was presented. In this paper we present a new simpler algorithm which matches the running time and space complexity of the above algorithm. The new algorithm is based on a different approach, by exploiting that the rank-maximal matching problem can be reduced to a maximum weight matching problem where the weight of an edge of rank i is $2^{\lceil \log n \rceil (r-i)}$. Using the fact that these edge weights are steeply distributed we design a scaling algorithm which scales by a factor of n in each phase. We also show that in each phase one maximum cardinality computation is sufficient to get a new optimal solution.

The new algorithm answers an open question, raised in the same paper, on whether the reduction to the maximum-weight matching problem can help us derive an efficient algorithm.

Keywords

Rank Maximal, Matching, Bipartite Graphs, Polynomial Algorithms, Maximum Weight Matching, Network Problems

1 Introduction

Consider a bipartite graph $G(V, E)$, $V = A \dot{\cup} B$, where $|V| = n, |E| = m$. To avoid triviality we assume that $n > 3$ and $m = \Omega(n)$. Let also a partition of the edge set into $r \leq m$ disjoint subsets $E = E_1 \dot{\cup} E_2 \dot{\cup} \ldots \dot{\cup} E_r$. Each set of edges E_i is denoted as the edges of rank i. A matching M is called rank-maximal if it maximizes $|M \cap E_1|$, and given that $|M \cap E_1|$ is maximized, it maximizes $|M \cap E_2|$, and so on. Note that such a matching is not necessarily of maximum cardinality. The problem is motivated as an optimization criteria when assigning a set of applicants to a set of posts. The matching M corresponds to the assignment and the ranks on the edges correspond to a ranking of the posts submitted by the applicants.

The rank-maximal matching problem can be solved by using the well-studied maximum weight matching problem. Given an edge e of rank i let its weight be $2^{\lceil \log n \rceil (r-i)}$. Since any matching M has less that n edges, it follows that no collection of edges of rank at least i can replace an edge of rank $i - 1$ and therefore given these weights a maximum weight matching is a rank-maximal matching. The maximum weight matching problem has been previously studied and efficient algorithms exist. The fastest algorithms have running time $O(n(m + n \log n))$ [2] and $O(\sqrt{n} m \log nC)$ [3] where C is the maximum weight in an instance. The first is strongly- while the second weakly-polynomial. Direct application of these algorithms on solving the rank-maximal matching problem does not result in efficient algorithms, neither in time nor in space requirements. This is a consequence of the fact that the edge weights are as large as n^r, which is non-polynomial in the input size. Both algorithms assume that arithmetic operations between numbers which are in $O(C)$ can be performed in constant time. This is not true in this case, where arithmetic on numbers in $O(n^r)$ takes time $\Omega(r)$. Hence, the running times are $O(rn(m + n \log n))$ and $O(r^2 \sqrt{n} m \log n)$ respectively, both using space $O(rn)$. It is known [4] that the scaling algorithm for the weighted matching problem can be implemented such that all arithmetic is performed on numbers with $O(\log n)$ bits independent of the edge weights. However, even such an implementation requires $O(r\sqrt{n} m \log n)$ time.

In what follows we present an algorithm which needs to perform r maximum cardinality matching computations, in order to compute a rank-maximal matching. Using the algorithm of Hopcroft and Karp [5] we get a running time $O(min(n + r, r\sqrt{n})m)$ using linear space. For sparse graphs we can also use the improved version using graph compression [6]. In the special case where there are at most n edges of each rank and at most n ranks, the algorithm requires time $O(nm)$. This special case corresponds to the case where each applicant ranks the posts according to its preference and no ties are allowed in the ranking.

The algorithm is very simple and is based on performing augmenting path calculations. Its correctness is based on the reduction to the maximum weight matching problem and a decomposition theorem for maximum weight matching presented in [7]. The time and space requirement are exactly the same with another combinatorial algorithm [1] which does not use the reduction to the maximum weight matching problem, but instead identifies edges which cannot be a part of a rank-maximal matching and deletes them. The new algorithm is simpler and more intuitive, in the sense that it is directly related to the weighted matching approach which is a natural way to approach the problem.

Given the partition of the edge set $E = E_1 \dot\cup E_2 \dot\cup \ldots \dot\cup E_r$ we can also ask for matchings satisfying different criteria. The *maximum cardinality rank-maximal* matching problem asks for a rank-maximal matching with maximum cardinality. Similarly the *fair* matching problem asks for a matching of maximum cardinality such that the minimum number of edges of rank r are used, and given that the minimum number of edges of rank $r-1$ are used, and so on. These problems can also be reduced to weighted matchings in bipartite graphs. Although the exact technique used in this work does not apply, the fastest method to solve these problems is by their corresponding weighted matching reductions. Note that the combinatorial technique in [1] does not trivially generalize. Such variations are the main motivation for trying to achieve the $O(min(n + r, r\sqrt{n})m)$ time bound using the weighted matching approach.

The rest of this paper is organized as follows. In Section 1.1 we give some necessary background. In Section 2 we present our new algorithm and prove its correctness. We summarize and offer conclusions in Section 3.

1.1 Preliminaries

Let $\pi : V \mapsto \mathbb{Z}_{\geq 0}$ be a *potential* function defined on the vertices of G. For an edge $e = (v, w) \in E$ denote its cost by $c(e)$ and define its *reduced cost* with respect to π as $\bar{c}(e) = \pi(v) + \pi(w) - c(e)$. Moreover call such an edge *tight* if $\bar{c}(e) = 0$. We say that π is a feasible potential function for c if $\bar{c}(e) \geq 0$ for all edges $e \in E$. We say that a feasible π is an optimal potential function for G if there is a matching M of G such that $\bar{c}(e) = 0$ for each $e \in M$ and $\pi(v) = 0$ for each $v \in V$ which is free in M. Define also $\Pi = \sum_{v \in V} \pi(v)$. Note that any matching M has

$$c(M) = \sum_{e=(v,w)\in M} c(e) \leq \sum_{e=(v,w)\in M} \pi(v) + \pi(w) \leq \sum_{v\in V} \pi(v) = \Pi$$

Moreover for an optimal potential function and the corresponding matching M, $c(M) = \Pi$ since any free vertex has zero potential.

In order to solve the rank-maximal matchings problem efficiently we use a decomposition theorem for maximum weight matching problems presented in [7]. For an integer $h \in [1, C]$ divide G into two lighter subgraphs G_h and G'_h as follows:

- G_h is formed by edges $(u, v) \in G$ such that $c(e) \in [C - h + 1, C]$. The weight of an edge $e \in G_h$ is $c(e) - (C - h)$.
- Let π_h be an optimal potential function for G_h. G'_h is formed by edges $e = (u, v) \in G$ such that $\bar{c}(e) = \pi_h(u) + \pi_h(v) - c(e) < 0$. The weight of an edge in G'_h is $-\bar{c}(e)$.

Theorem 1 ([7]) *Consider G, G_h and G'_h as defined above and let $mwm(G)$ denote the cost of a maximum weight matching in G. Then*

$$mwm(G) = mwm(G_h) + mwm(G'_h)$$

If π_h and π'_h are optimal potential functions for G_h and G'_h respectively, then $\pi_h + \pi'_h$ is an optimal potential function for G. This follows directly by Theorem 1 and by the fact that this potential function is feasible for the original cost function. See [7] for more details.

2 The algorithm

Consider an instance of the rank-maximal matching problem. Graph G has edges with weights of the form $1, n, n^2, \ldots, n^{r-1}$. We use Theorem 1 in order to decompose the problem into two easier subproblems and solve it recursively. The base case for the recursion will be a maximum cardinality matching computation.

Choose $h = n^{r-1} - 1$. Then G_h contains all edges of G with costs in the range $[2, n^{r-1}]$. Each edge in G_h has cost $c_h(e) = c(e) - 1$. Therefore, graph G_h contains all edges of G with rank at most $r - 1$ and these edges have costs $n - 1, n^2 - 1, \ldots, n^{r-1} - 1$. Graph G'_h is much simpler. Assuming that π_h is an optimal potential function for G_h, G'_h contains only edges of G with negative reduced costs w.r.t π_h. Such edges fall into two categories:

- Edges $e = (u, w) \in G$ where $c(e) = 1$, if $\pi_h(u) + \pi_h(w) = 0$. Such edges have cost 1 in G'_h.

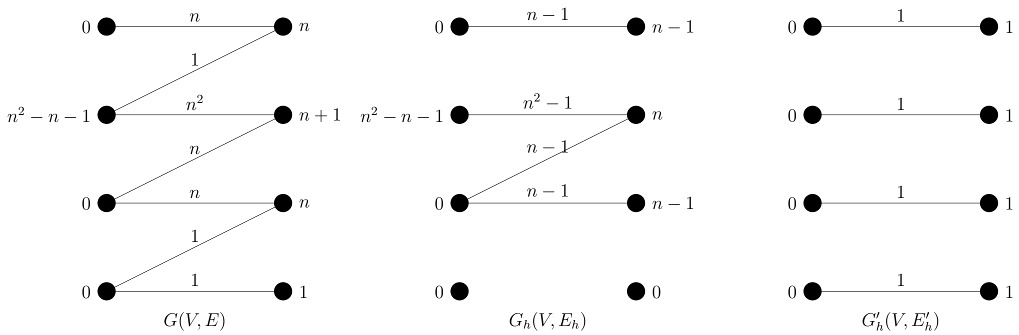

Figure 1: Example of an instance of the rank-maximal matching problem reduced to maximum weight matching and the two graphs resulting from the decomposition. The numbers next to the edges represent the costs. The numbers next to the vertices represent an optimal potential function.

- Edges $e = (u, w) \in G$ where $c(e) > 1$, if $\pi_h(u) + \pi_h(w) - c(e) < 0$. Due to the feasibility of π_h, $\pi_h(u) + \pi_h(w) - c(e) \geq -1$, and therefore all such edges have also cost 1 in G'_h. These edges are exactly the ones in G_h that are tight w.r.t. π_h.

The problem has been split into two subproblems, one of which is a maximum cardinality matching computation (see Figure 1 for an example of the decomposition). Let π'_h be the optimal potential function for G'_h. Such a potential function can be computed by a maximum cardinality computation. The other subproblem, as it will be shortly clear, is the initial problem with one fewer type of weight for the edges. This is a consequence of the steeply increasing weight function, i.e., the large difference between the weight of two edges of different ranks. Some extra care is required in order to watch out for the cost of arithmetic operations during the algorithm since potential function π_h can have values up to $O(n^r)$. It turns out that we do not actually need to explicitly store or maintain π_h, but use it only for correctness.

The edges of G_h have costs of the form $n-1, n^2-1, \ldots, n^{r-1}-1$. We are going to relate the optimal potential function for these costs with the optimal potential function for the costs $1, n, n^2, \ldots, n^{r-2}$. Then instead of computing a maximum weight matching for the costs resulting from our decomposition for G_h, we will compute it for the costs $1, n, n^2, \ldots, n^{r-2}$. These costs are the original problem with one less type of weights and can be solved recursively in time $T(r-1)$ assuming $T(r)$ is the time required to solve the original problem.

For an optimal potential function π, consider the following representation:

- A set of nodes V^0 containing the nodes $v \in V$ with $\pi(v) = 0$.

- A set of edges E^0 containing the edges $e = (u, w) \in E$ which are tight w.r.t π, i.e., $\bar{c}(e) = \pi(u) + \pi(w) - c(e) = 0$.

The above two sets are defined w.r.t an optimal potential function for an instance, and therefore can guide the construction of a matching M in $O(\sqrt{n}m)$ such that any matched edge is tight and every free vertex has zero potential. The above imply that such a matching has the same cost as our optimal potential function and therefore is itself optimal. The choice of such a representation of a potential function, is based on the need to maintain the arithmetic demands of the algorithm to a reasonable level. During the algorithm we will manipulate these two sets and maintain the invariant that all such tuples will correspond to the representation of some optimal potential function.

Assume that we have solved the subproblem with edge costs $1, n, n^2, \ldots, n^{r-2}$ in time $T(r-1)$ and we have an optimal potential function π represented with V^0 and E^0. In order to obtain an optimal potential function for the edge costs $n-1, n^2-1, \ldots, n^{r-1}-1$ the first step is to obtain an optimal potential function for the edge costs $n, n^2, \ldots, n^{r-1}$. The following lemma is part of the folklore.

Lemma 1 *Consider a graph G with edge costs $1, n, n^2, \ldots, n^{r-2}$. Let a maximum weight matching M of G and a potential function $\pi : V \mapsto \mathbb{Z}_{\geq 0}$ proving its optimality. Then the potential function $n\pi$ proves the optimality of M for G with edge costs $n, n^2, n^3, \ldots, n^{r-1}$.*

Corollary 1 *Let π' be the potential function obtained by multiplying π with n in Lemma 1. Then for any node $v \in V$ either $\pi'(v) = 0$ or $\pi'(v) \geq n$. Moreover for any edge $e \in E$ either $\bar{c}(e) = \pi'(v) + \pi'(w) - c(e) = 0$ or $\bar{c}(e) \geq n$.*

Proof: Follows directly from the feasibility of π, the definition of π' and the integrality of the potential functions. ∎

Let π' be the potential function obtained by multiplying π with n in Lemma 1. Note that the same sets V^0 and E^0 are a representation of the new optimal potential function π' for the new costs. The next step is the construction of a potential function π'' which will be optimal for the costs $n-1, n^2-1, \ldots, n^{r-1}-1$. We will need the following definition.

Definition 1 (equality subgraph) *For a graph $G(V, E)$ with edge costs $c : E \mapsto \mathbb{Z}_{>0}$ and a potential function $\pi : V \mapsto \mathbb{Z}_{\geq 0}$, let the equality subgraph $G_=(V, E_=)$ be the graph with edge set $E_= = \{e = (u, v) \in E : \bar{c}(e) = \pi(u) + \pi(v) - c(e) = 0\}$.*

Input: optimal potential function π' for edge costs $n, n^2, \ldots, n^{r-1}$
Output: optimal potential function π'' for edge costs
$n-1, n^2-1, \ldots, n^{r-1}-1$

while $G_{h=}$ has a vertex v with $\pi'(v) = 0$
  let A_v and B_v be the vertices reachable from v in $G_{h=}$ by even and odd paths respectively
  for $w \in A_v$ set $\pi''(w) = \pi'(w)$
  for $w \in B_v$ set $\pi''(w) = \pi'(w) - 1$
  delete A_v and B_v
endwhile
if $G_{h=}$ is not empty
  let $A \dot\cup B$ be the remaining vertex set of $G_{h=}$
  for each $w \in A$, set $\pi''(w) = \pi'(w)$
  for each $w \in B$, set $\pi''(w) = \pi'(w) - 1$
endif

Figure 2: Construct optimal potential function

In the following let $G_{h=}$ be the equality subgraph of G_h w.r.t π'. The algorithm in Figure 2 finds a new optimal potential function. The following lemmata prove its correctness.

Lemma 2 *Let G be a graph with edge costs $n, n^2, \ldots, n^r$, $n > 1$ and $\pi : V \mapsto \mathbb{Z}_{\geq 0}$ be an optimal potential function for G. Let $G_=$ be the equality subgraph of G w.r.t π, meaning the edges which have reduced cost zero, and let $v \in V$ be a vertex with $\pi(v) = 0$. Then, there is no odd length path in $G_=$ starting from v which ends to a vertex w with $\pi(w) = 0$.*

Proof: Assume otherwise and let p be an odd length path in $G_=$ from v to w such that $\pi(v) = \pi(w) = 0$ (see Figure 3). Since every edge on p has reduced cost zero we get the following

$$\sum_{i \in \{\alpha_1, \alpha_2, \ldots, \alpha_k\}} n^i = \sum_{j \in \{\beta_1, \beta_2, \ldots, \beta_{k-1}\}} n^j \tag{1}$$

where $\alpha_i \geq 1, \beta_i \geq 1, n > 1$ and $2k-1 < n$. Assume w.l.o.g that there are no i, j such that $n^{\alpha_i} = n^{\beta_j}$, otherwise remove both of them. At least one term will remain since the number of terms is odd. Both sides have at most $k < n$

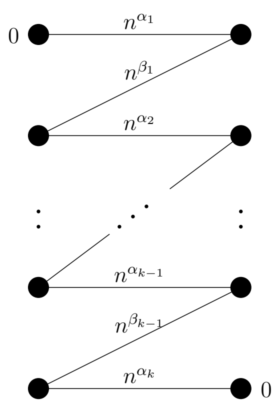

Figure 3: Path p in equality subgraph $G_=$ for proof of Lemma 2.

terms and any term in one side needs at least n terms on the other side in order to cancel out. But there are not so many terms and therefore we get a contradiction. ∎

Lemma 3 *The algorithm in Figure 2 constructs an optimal potential function π'' for the edge costs $n-1, n^2-1, \ldots, n^{r-1}-1$ given an optimal potential function π' for the edge costs $n, n^2, \ldots, n^{r-1}$.*

Proof: By Corollary 1 every edge $e = (v, w)$ which does not belong to $G_{h=}$ has $\bar{c}'(e) = \pi'(v) + \pi'(w) - c(e) \geq n$. Every potential can decrease by at most 1 during the algorithm and therefore $\bar{c}''(e) = \pi''(v) + \pi''(w) - (c(e) - 1) \geq \pi'(v) - 1 + \pi'(w) - 1 - (c(e) - 1) = \pi'(v) + \pi'(w) - c(e) - 1 \geq n - 1 > 0$ since $n > 1$. This means that all these edges are feasible and every such edge remains non-tight.

In the first part of the algorithm, by Lemma 2 all vertices v that get their potential decreased have $\pi'(v) > 0$. The same is true for the second part, since the second part is executed only if $G_{h=}$ is non-empty and there is no vertex with zero potential. By Corollary 1 every such vertex has $\pi'(v) \geq n$ and therefore $\pi''(v) \geq n-1 > 0$. Hence, the set of vertices with zero potential remain the same.

We will now examine the effect on $G_{h=}$. For each edge e that is tight w.r.t π', exactly one of its endpoints get their potential reduced by 1 and therefore e remains tight in π'' w.r.t the cost function $c(e) - 1$.

Finally, let M be a maximum weight matching such that π' proves its optimality. Every edge $e \in M$ belongs to $G_{h=}$ and every free vertex has zero potential. The potential function π'' also proves the optimality of M w.r.t

Input: graph G with edge partition $E_1, E_2, \ldots, E_r$
Output: sets V^0, E^0

if $r = 1$
  compute maximum matching M of G and optimal π
  based on π compute V^0 and E^0 and return them
else
  solve recursively instance for $G(V, E \setminus E_r)$ and $E_1, E_2, \ldots, E_{r-1}$
  let V^0, E^0 be the solution
  let $E^\ddagger = \{e = (v, u) \in E_r : \{v, u\} \in V^0\}$
  form unweighted $G'_h(V, E^\ddagger \cup E^0)$ and find optimal potential
  function π'_h in G'_h
  set $E^0 = \{e = (v, u) \in E^\ddagger \cup E^0 : \pi'_h(v) + \pi'_h(u) = 1\}$
  set $V^0 = \{v \in V^0 : \pi'_h(v) = 0\}$
  return V^0 and E^0
endif

Figure 4: Find a rank-maximal matching

the new cost function, since the sets V^0 and E^0 have not changed. Thus the resulting function is optimal. ∎

Given the solution for edge costs $1, n, n^2, \ldots, n^{r-2}$ as V^0 and E^0, the same sets are a solution for the edge costs $n-1, n^2-1, \ldots, n^{r-1}-1$. More precisely, there is a potential function such that these two sets are its representation. Note that the above lemma is a result of the edge cost distribution, and the fact that reducing by one every edge cost results in an instance which has exactly the same maximum weight matchings.

2.0.1 Combining the solutions

Given a solution of the subproblems, induced by the decomposition of the original graph it remains to combine them to get the solution for the original problem. Combining the two solutions requires to add up the two potential functions, π_h and π'_h. The addition will be performed implicitly by changing V^0 and E^0 based on the potential function π'_h. Updating V^0 requires checking for each $v \in V$ whether $\pi_h(v) = \pi'_h(v) = 0$ which can be done by checking whether $\pi'_h(v) = 0$ and $v \in V^0$. Updating E^0 is slightly more complicated.

- For an edge $e = (v, u)$ with $c(e) = 1$ ($e \in E_r$) in G we have to check whether $\pi_h(v) + \pi'_h(v) + \pi_h(u) + \pi'_h(u) = 1$. Recall that if a vertex $v \in V$ has $\pi_h(v) > 0$ then $\pi_h(v) > n - 2$ and therefore it is enough to check (a) that $\pi'_h(v) + \pi'_h(u) = 1$ an operation which takes constant time since π'_h is polynomially bounded and (b) that $\pi_h(v) = \pi_h(u) = 0$ which can be done by checking whether $\{v, u\} \in V^0$.

- For the rest of the edges, we have to check whether $\pi_h(v) + \pi'_h(v) + \pi_h(u) + \pi'_h(u) = c(e)$, but by feasibility of π_h we know that $\pi_h(v) + \pi_h(u) \geq c(e) - 1$.

  Moreover, for an edge e s.t $\pi_h(v) + \pi_h(u) \neq c(e) - 1$ we know that $\pi_h(v) - 1 + \pi_h(u) - 1 - (c(e) - 1) \geq n - 1$ (in the worst case where both endpoints got their potential decreased by 1, when transforming to a new potential function for edge weights which are reduced by 1), and hence any such edge cannot be tight.

  We conclude that if an edge has $\pi_h(v) + \pi_h(u) = c(e) - 1$ and therefore belongs already to E^0, then it will remain tight if $\pi'_h(v) + \pi'_h(u) = 1$.

The final output of the algorithm is sets V^0 and E^0, which represent an optimal potential function. See Figure 4 for a succinct description.

From these sets a rank-maximal matching can be constructed by performing one maximum cardinality matching computation. This is done in the following way. Let $G_=(V, E^0)$ be the final equality subgraph. Create a new graph, $G^{\alpha\beta}$, containing two copies of $G_=$, $G^\alpha(V, E^\alpha)$ and $G^\beta(V, E^\beta)$. For a vertex $v \in V$, let v^α and v^β be the two copies in the new graph. Then, if $v \in V^0$ include the edge (v^α, v^β). Finally find a maximum cardinality matching M in $G^{\alpha\beta}$. The matching $M \cap E^\alpha$ is then a maximum weight matching in the original graph, as the following lemma proves.

Lemma 4 *The matching $M \cap E^\alpha$ constructed as above, is a maximum weight matching in G.*

Proof: Let π be the optimal potential function for G, represented as the sets V^0 and E^0. We first argue that the graph $G^{\alpha\beta}$ has a perfect matching. Let M' be a maximum weight matching in G corresponding to π. Each edge in M' is tight w.r.t π and therefore belongs to E^0. Thus, M' is also a matching of $G_=$. Take two copies of this matching one for G^α and one for G^β. All remaining unmatched nodes in $G^{\alpha\beta}$ were also unmatched in M' and therefore have zero potential. We added the extra edges $v^\alpha v^\beta$ in case a vertex v had zero potential. Together with these edges a perfect matching for $G^{\alpha\beta}$ can be formed.

Let now M be a maximum cardinality matching of $G^{\alpha\beta}$. By the above M is a perfect matching. Consider any node v s.t $v \notin V^0$. There is no edge between v^α and v^β and therefore v^α must be matched inside G^α. Therefore, any node $v \notin V^0$ must be matched by $M \cap E^\alpha$. Hence,

$$\Pi = \sum_{v \in V} \pi(v) = \sum_{\substack{v \in V \\ v \notin V^0}} \pi(v) = \sum_{e \in M \cap E^\alpha} c(e) = c(M \cap E^\alpha)$$

and the lemma follows. ∎

In the case where $r = 1$, finding an optimal potential function is done in the following way. First calculate a maximum cardinality matching. Such a matching partitions the vertices of the graph into three categories, based on whether a vertex is reachable or *unreachable* from a free vertex by an alternating path and for reachable vertices based on whether it is reachable by an *even* or *odd* length alternating path. For each *even* vertex $v \in V$ set $\pi'_h(v) = 0$, and for each *odd* vertex v set $\pi'_h(v) = 1$. For *unreachable* vertices set $\pi'_h(v)$ either to 0 or 1 depending on whether v is in the right or the left side of the bipartite graph. It is easy to see that this potential function is feasible, every free vertex has zero potential and each edge which belongs to the matching is tight.

2.1 Running Time

Denote the algorithm's running time as $T(r)$ for an instance with r ranks. Moreover, assume that together with sets V^0, E^0 the recursive call returns a maximum cardinality matching M^0 of the graph induced by the edges in E^0. Then $T(r)$ consists of solving a subproblem of size $r-1$ recursively in $T(r-1)$, a maximum matching computation and the time taken to combine the two solutions. Finding a new maximum cardinality matching can be done in $O(min(\sqrt{n}, |M'|-|M^0|+1)m)$ where M' is the new maximum matching. All the administrative work, like updating the various graphs and sets, does not dominate the running time. The recursion solves to $O(min(n+r, r\sqrt{n})m)$, a running time which matches the currently best known running time [1] for the rank-maximal matching problem. The space requirement of the algorithm is linear.

2.1.1 Fewer phases

The number of phases can be reduced from r to r^*, where r^* is the largest rank used in an optimal solution for a particular instance. Assume we are at the start of phase i. We have sets V^0 and E^0 corresponding to edges of ranks $i-1, \ldots, 1$. These edges now have weights $n-1, \ldots, n^i-1$. We can check whether this matching is already optimal by assuming that all remaining edges are of rank i, and therefore solving our initial problem with edge weights $n^{r-i}, \dots, n^{r-i}, n^{r-i+1}, \dots, n^{r-1}$ or equivalently for $1, \dots, 1, n, \dots, n^{i-1}$. This way we simply boost the importance of less important edges in order to check whether they would be used or not. Solving this instance can be done easily, since only graph G'_h has changed by including more edges with weight 1. To summarize, in phase i we first form G'_h by including the appropriate edges from edges $e \in E_i \cup E_{i+1} \cup \cdots \cup E_r$ and try to find a maximum weight matching in G'_h. If the resulting matching after summing up the two potential functions does not use any edge of rank $E_{\geq i}$ then we know that we already found a rank-maximal matching. Otherwise we form G'_h by including only edges from E_i as we would normally do and continue the phases.

3 Conclusions

We presented a new algorithm which solves the rank-maximal matching problem by r maximum cardinality matching computations. Using the Hopcroft and Karp algorithm [5] we get time $O(min(n+r, r\sqrt{n})m)$ using linear space.

The algorithm is based on the idea of reducing the problem to the maximum weight matching problem. Doing this reduction implicitly, allows us to maintain the numbers appearing during the computation up to a reasonable level. The weights of a reduced instance can be as large as $O(n^r)$, but they are steeply distributed. This allows us to design a scaling algorithm which scales in each phase by n, resulting in $O(\log_n n^r)$ phases in total. In each phase we show that one maximum cardinality matching computation is enough in order to find an optimal solution for the new edge weights.

Our algorithm answers an open question of [1], on whether such a running time can be achieved by using the maximum weight matching reduction.

References

[1] R. W. Irving, T. Kavitha, K. Mehlhorn, D. Michail, K. Paluch, Rank-maximal matchings, in: Proceedings of the fifteenth annual ACM-SIAM symposium on Discrete algorithms, Society for Industrial and Applied Mathematics, 2004, pp. 68–75.

[2] R. K. Ahuja, T. L. Magnanti, J. B. Orlin, Network Flows: Theory, Algorithms, and Applications, Prentice Hall, 1993.

[3] H. N. Gabow, R. Tarjan, Faster scaling algorithms for network problems, SIAM Journal of Computing 18 (1989) 1013–1036.

[4] A. V. Goldberg, R. Kennedy, An efficient cost scaling algorithm for the assignment problem, Math. Program. 71 (2) (1995) 153–177.

[5] J. E. Hopcroft, R. M. Karp, An $n^{5/2}$ algorithm for maximum matching in bipartite graphs, SIAM Journal on Computing 2 (1973) 225–231.

[6] T. Feder, R. Motwani, Clique partitions, graph compression and speeding-up algorithms, in: Proceedings of the twenty-third annual ACM symposium on Theory of computing, 1991, pp. 123–133.

[7] M.-Y. Kao, T. W. Lam, W.-K. Sung, H.-F. Ting, A decomposition theorem for maximum weight bipartite matchings, SIAM Journal of Computing 31 (1) (2001) 18–26.

Below you find a list of the most recent technical reports of the Max-Planck-Institut für Informatik. They are available by anonymous ftp from `ftp.mpi-sb.mpg.de` under the directory `pub/papers/reports`. Most of the reports are also accessible via WWW using the URL `http://www.mpi-sb.mpg.de`. If you have any questions concerning ftp or WWW access, please contact `reports@mpi-sb.mpg.de`. Paper copies (which are not necessarily free of charge) can be ordered either by regular mail or by e-mail at the address below.

Max-Planck-Institut für Informatik
Library
attn. Anja Becker
Stuhlsatzenhausweg 85
66123 Saarbrücken
GERMANY
e-mail: `library@mpi-sb.mpg.de`

MPI-I-2005-5-002	S. Siersdorfer, G. Weikum	Automated Retraining Methods for Document Classification and their Parameter Tuning
MPI-I-2005-4-006	C. Fuchs, M. Goesele, T. Chen, H. Seidel	An Emperical Model for Heterogeneous Translucent Objects
MPI-I-2005-4-005	G. Krawczyk, M. Goesele, H. Seidel	Photometric Calibration of High Dynamic Range Cameras
MPI-I-2005-4-004	C. Theobalt, N. Ahmed, E. De Aguiar, G. Ziegler, H. Lensch, M.A.,. Magnor, H. Seidel	Joint Motion and Reflectance Capture for Creating Relightable 3D Videos
MPI-I-2005-4-003	T. Langer, A.G. Belyaev, H. Seidel	Analysis and Design of Discrete Normals and Curvatures
MPI-I-2005-4-002	O. Schall, A. Belyaev, H. Seidel	Sparse Meshing of Uncertain and Noisy Surface Scattered Data
MPI-I-2005-4-001	M. Fuchs, V. Blanz, H. Lensch, H. Seidel	Reflectance from Images: A Model-Based Approach for Human Faces
MPI-I-2005-2-004	Y. Kazakov	A Framework of Refutational Theorem Proving for Saturation-Based Decision Procedures
MPI-I-2005-2-003	H.d. Nivelle	Using Resolution as a Decision Procedure
MPI-I-2005-2-002	P. Maier, W. Charatonik, L. Georgieva	Bounded Model Checking of Pointer Programs
MPI-I-2005-2-001	J. Hoffmann, C. Gomes, B. Selman	Bottleneck Behavior in CNF Formulas
MPI-I-2005-1-007	I. Katriel, M. Kutz	A Faster Algorithm for Computing a Longest Common Increasing Subsequence
MPI-I-2005-1-003	S. Baswana, K. Telikepalli	Improved Algorithms for All-Pairs Approximate Shortest Paths in Weighted Graphs
MPI-I-2005-1-002	I. Katriel, M. Kutz, M. Skutella	Reachability Substitutes for Planar Digraphs
MPI-I-2005-1-001	D. Michail	Rank-Maximal through Maximum Weight Matchings
MPI-I-2004-NWG3-001	M. Magnor	Axisymmetric Reconstruction and 3D Visualization of Bipolar Planetary Nebulae
MPI-I-2004-NWG1-001	B. Blanchet	Automatic Proof of Strong Secrecy for Security Protocols
MPI-I-2004-5-001	S. Siersdorfer, S. Sizov, G. Weikum	Goal-oriented Methods and Meta Methods for Document Classification and their Parameter Tuning
MPI-I-2004-4-006	K. Dmitriev, V. Havran, H. Seidel	Faster Ray Tracing with SIMD Shaft Culling
MPI-I-2004-4-005	I.P. Ivrissimtzis, W.-. Jeong, S. Lee, Y.a. Lee, H.-. Seidel	Neural Meshes: Surface Reconstruction with a Learning Algorithm
MPI-I-2004-4-004	R. Zayer, C. Rssl, H. Seidel	r-Adaptive Parameterization of Surfaces

MPI-I-2004-4-003	Y. Ohtake, A. Belyaev, H. Seidel	3D Scattered Data Interpolation and Approximation with Multilevel Compactly Supported RBFs
MPI-I-2004-4-002	Y. Ohtake, A. Belyaev, H. Seidel	Quadric-Based Mesh Reconstruction from Scattered Data
MPI-I-2004-4-001	J. Haber, C. Schmitt, M. Koster, H. Seidel	Modeling Hair using a Wisp Hair Model
MPI-I-2004-2-007	S. Wagner	Summaries for While Programs with Recursion
MPI-I-2004-2-002	P. Maier	Intuitionistic LTL and a New Characterization of Safety and Liveness
MPI-I-2004-2-001	H. de Nivelle, Y. Kazakov	Resolution Decision Procedures for the Guarded Fragment with Transitive Guards
MPI-I-2004-1-006	L.S. Chandran, N. Sivadasan	On the Hadwiger's Conjecture for Graph Products
MPI-I-2004-1-005	S. Schmitt, L. Fousse	A comparison of polynomial evaluation schemes
MPI-I-2004-1-004	N. Sivadasan, P. Sanders, M. Skutella	Online Scheduling with Bounded Migration
MPI-I-2004-1-003	I. Katriel	On Algorithms for Online Topological Ordering and Sorting
MPI-I-2004-1-002	P. Sanders, S. Pettie	A Simpler Linear Time 2/3 - epsilon Approximation for Maximum Weight Matching
MPI-I-2004-1-001	N. Beldiceanu, I. Katriel, S. Thiel	Filtering algorithms for the Same and UsedBy constraints
MPI-I-2003-NWG2-002	F. Eisenbrand	Fast integer programming in fixed dimension
MPI-I-2003-NWG2-001	L.S. Chandran, C.R. Subramanian	Girth and Treewidth
MPI-I-2003-4-009	N. Zakaria	FaceSketch: An Interface for Sketching and Coloring Cartoon Faces
MPI-I-2003-4-008	C. Roessl, I. Ivrissimtzis, H. Seidel	Tree-based triangle mesh connectivity encoding
MPI-I-2003-4-007	I. Ivrissimtzis, W. Jeong, H. Seidel	Neural Meshes: Statistical Learning Methods in Surface Reconstruction
MPI-I-2003-4-006	C. Roessl, F. Zeilfelder, G. Nrnberger, H. Seidel	Visualization of Volume Data with Quadratic Super Splines
MPI-I-2003-4-005	T. Hangelbroek, G. Nrnberger, C. Roessl, H.S. Seidel, F. Zeilfelder	The Dimension of C^1 Splines of Arbitrary Degree on a Tetrahedral Partition
MPI-I-2003-4-004	P. Bekaert, P. Slusallek, R. Cools, V. Havran, H. Seidel	A custom designed density estimation method for light transport
MPI-I-2003-4-003	R. Zayer, C. Roessl, H. Seidel	Convex Boundary Angle Based Flattening
MPI-I-2003-4-002	C. Theobalt, M. Li, M. Magnor, H. Seidel	A Flexible and Versatile Studio for Synchronized Multi-view Video Recording
MPI-I-2003-4-001	M. Tarini, H.P.A. Lensch, M. Goesele, H. Seidel	3D Acquisition of Mirroring Objects
MPI-I-2003-2-004	A. Podelski, A. Rybalchenko	Software Model Checking of Liveness Properties via Transition Invariants
MPI-I-2003-2-003	Y. Kazakov, H. de Nivelle	Subsumption of concepts in $DL\ \mathcal{FL}_0$ for (cyclic) terminologies with respect to descriptive semantics is PSPACE-complete
MPI-I-2003-2-002	M. Jaeger	A Representation Theorem and Applications to Measure Selection and Noninformative Priors
MPI-I-2003-2-001	P. Maier	Compositional Circular Assume-Guarantee Rules Cannot Be Sound And Complete
MPI-I-2003-1-018	G. Schaefer	A Note on the Smoothed Complexity of the Single-Source Shortest Path Problem
MPI-I-2003-1-017	G. Schfer, S. Leonardi	Cross-Monotonic Cost Sharing Methods for Connected Facility Location Games
MPI-I-2003-1-016	G. Schfer, N. Sivadasan	Topology Matters: Smoothed Competitive Analysis of Metrical Task Systems
MPI-I-2003-1-015	A. Kovcs	Sum-Multicoloring on Paths

MPI-I-2003-1-014	G. Schfer, L. Becchetti, S. Leonardi, A. Marchetti-Spaccamela, T. Vredeveld	Average Case and Smoothed Competitive Analysis of the Multi-Level Feedback Algorithm
MPI-I-2003-1-013	I. Katriel, S. Thiel	Fast Bound Consistency for the Global Cardinality Constraint
MPI-I-2003-1-012		- not published -