I N F O R M A T I K

A Simpler Linear Time $2/3 - \epsilon$ Approximation for Maximum Weight Matching

Seth Pettie and Peter Sanders

MPI–I–2004–1–002 February 2004

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

Seth Pettie
Max-Planck-Institut für Informatik
Stuhlsatzenhausweg 85
66123 Saarbrücken, Germany
email: `pettie@mpi-sb.mpg.de`

Peter Sanders
Max-Planck-Institut für Informatik
Stuhlsatzenhausweg 85
66123 Saarbrücken, Germany
email: `sanders@mpi-sb.mpg.de`

Abstract

We present two $\frac{2}{3} - \epsilon$ approximation algorithms for the maximum weight matching problem that run in time $\mathcal{O}\left(m \log \frac{1}{\epsilon}\right)$. We give a simple and practical randomized algorithm and a somewhat more complicated deterministic algorithm. Both algorithms are exponentially faster in terms of ϵ than a recent algorithm by Drake and Hougardy. We also show that our algorithms can be generalized to find a $1 - \epsilon$ approximatation to the maximum weight matching, for any $\epsilon > 0$.

Keywords

maximum weight matching, approximation algorithm

1 Introduction

Consider an undirected weighted graph $G = (V, E, w)$, where m and n are the number of edges and vertices, respectively, and $w(e)$ denotes the weight of edge $e \in E$. A *matching* is a set of edges $M \subseteq E$ that are endpoint disjoint from one another. The *maximum weight matching* problem (or MWM) is to find a matching M^* of maximum weight, where $w(M^*) \stackrel{\textbf{def}}{=} \sum_{e \in M^*} w(e)$. The fastest algorithms for solving this problem run in polynomial time: $\mathcal{O}(mn + n^2 \log n)$ for real-weighted graphs [4] and $\mathcal{O}(m\sqrt{n} \cdot \mathrm{polylog}(nC))$ time [6] when the weights are integers less than C. Despite these nice polynomial-time solutions there is considerable interest in simpler and faster algorithms — ideally linear time — that return a solution of some guaranteed quality. For example, weighted matchings are a crucial subroutine for partioning large networks like finite element meshes and VLSI circuits (see, e.g. [8]). We define the δ-MWM problem to be that of finding any matching whose weight is at least $\delta \cdot w(M^*)$.

There is a well known $\frac{1}{2}$-MWM algorithm based on a simple greedy strategy: scan the edges in increasing order of weight, selecting the current edge if it is vertex-disjoint from previously selected edges. This algorithm requires $\mathcal{O}(m \log n)$ time[1]. Preis [8], and later Drake and Hougardy [3], presented linear-time $\frac{1}{2}$-MWM algorithms.

While the $\frac{1}{2}$-MWM algorithms above only compare adjacent edges, it is possible to achieve better approximations by examining short augmenting paths. Drake and Hougardy [2] observed that if a matching M is such that any weight-augmenting path or cycle has more than 2 unmatched edges, then $w(M) \geq \frac{2}{3} w(M^*)$. In a subsequent paper [1] Drake and Hougardy developed a $(\frac{2}{3} - \epsilon)$-MWM algorithm running in time $O(m \cdot \mathrm{poly}(\epsilon^{-1}))$.

The Drake-Hougardy algorithm is somewhat complicated and requires a very detailed analysis (important parts are only sketched in [1].) Moreover, it converges towards a $\frac{2}{3} - \epsilon$ solution very slowly. For example, if $\epsilon = .1$ (that is, we want a .57 approximation to the maximum weight) the algorithm requires over twenty iterations, each involving a traversal of the whole graph. In this paper we give two simple $(\frac{2}{3} - \epsilon)$-MWM algorithms, each running in $\mathcal{O}\big(m \log \frac{1}{\epsilon}\big)$ time. Our first algorithm is randomized and admits a simple analysis. It rivals all previous matching algorithms in terms of simplicity and promises to be a good choice in practice. Our deterministic algorithm is slightly more complicated and requires a more sophisticated analysis. Both algorithms converge on a $\frac{2}{3} - \epsilon$ solution in exponentially fewer iterations than the Drake-Hougardy approach.

Although we can only obtain a linear running time for the $(\frac{2}{3} - \epsilon)$-MWM problem, both our algorithms can be extended in purely mechanical ways to the δ-MWM problem, for any δ. For graphs with sufficiently low degree, our δ-MWM algorithms are faster than the $\mathcal{O}(m\sqrt{n} \cdot \log(n(1-\delta)^{-1}))$ algorithm of [6], for constant δ.[2] It is not clear to us whether the Drake-Hougardy approach can be easily extended to the δ-MWM problem, for $\delta > \frac{2}{3}$.

[1]One can get a $(\frac{1}{2} - m^{-k})$-MWM algorithm in time $\mathcal{O}(km)$ using base m radix sorting of weights rounded to multiples of $\max_{e \in E} w(e)/m^{k+1}$.

[2]As Gabow and Tarjan [5, 6] note, their weighted matching algorithm can be viewed either as an exact algorithm for integer-weighted graphs or as an approximation algorithm for arbitrary graphs.

2 Terminology and Notation

Most of our definitions are implicitly with respect to some matching called M, which in our algorithms is the matching currently under consideration. The maximum weight matching is M^*. A path or cycle is *alternating* if it consists of edges drawn alternately from M and $E\backslash M$. An alternating path or cycle P is an *augmentation* if $M \oplus P$ is also a matching, where $A \oplus B = (A\backslash B) \cup (B\backslash A)$. The *gain* of an alternating path/cycle P is $g(P) = w(P\backslash M) - w(P \cap M)$. The gain of a set of (not necessarily disjoint) paths/cycles is the sum of their individual gains. A k-*augmentation* is one containing at most k non-M edges.

It is well known that if a matching admits no positive-gain k-augmentations then it must have weight at least $k/(k+1)$ of the maximum. See [7] for the unweighted version of this theorem and [2] for the weighted version. Theorem 2.1 shows that any matching can be brought geometrically closer to a $k/(k+1)$-optimal one using *disjoint* k-augmentations.

Theorem 2.1 *For any matching M, there exists a collection A of vertex-disjoint k-augmentations such that $w(M \oplus A) \geq w(M) + \frac{k+1}{2k+1}(\frac{k}{k+1}w(M^*) - w(M))$.*

Proof: The graph $C = M \oplus M^*$ consists of alternating paths and cycles w.r.t. M or M^*. We may assume w.l.o.g. that C is a single path/cycle; our argument is applied to each separately. If C is a (2ℓ)-cycle, list its edges in cyclic order: $e_0, e_0^*, e_1, e_1^*, \ldots, e_{\ell-1}, e_{\ell-1}^*$, where $e_i \in M$, $e_i^* \in M^*$. To conserve ink, let $k^+ = k+1$. Let A_i be the set of disjoint k-augmentations

$$\{\{e_i, \ldots, e_{i+k^+-1}\}, \{e_{i+k^+}, \ldots, e_{i+2k^+-1}\}, \ldots, \{e_{i+k^+\lfloor(\ell-k^+)/k^+\rfloor}, \ldots, e_{i+k^+\lfloor(\ell-k^+)/k^+\rfloor+k^+-1}\}\}$$

That is, the augmentations in A_i are disjoint and the only M-edges not in A_i are the $(\ell \bmod k^+)$ ones at the end of the list, when starting at e_i. We wish to lower bound the gain of the best set of augmentations. Clearly $\max_i g(A_i) \geq \sum_i g(A_i)/\ell$. One can easily see that in $\sum_i g(A_i)$, each M^*-edge is counted $(k^+-1)\lfloor \ell/k^+ \rfloor$ times, and each M-edge $k^+\lfloor \ell/k^+ \rfloor$ times. Therefore,

$$\begin{aligned}
\sum_{i=0}^{\ell-1} g(A_i)/\ell &= \left[(k^+-1)\left\lfloor \frac{\ell}{k^+} \right\rfloor w(M^*) - k^+ \left\lfloor \frac{\ell}{k^+} \right\rfloor w(M)\right]/\ell \\
&= \frac{k^+}{\ell}\left\lfloor \frac{\ell}{k^+} \right\rfloor \left(\frac{k^+-1}{k^+} w(M^*) - w(M)\right) \\
&\geq \frac{k^+}{2k^+-1}\left(\frac{k^+-1}{k^+} w(M^*) - w(M)\right) = \frac{k+1}{2k+1}\left(\frac{k}{k+1} w(M^*) - w(M)\right)
\end{aligned}$$

If C is a path, list the edges as before. Let $M_i^* = \{e_j^* : j = i \pmod{k+1}\}$ and $A_i = C\backslash M_i^*$. A_i consists of disjoint k-augmentations and $\sum_{i=0}^{k} g(A_i) = k\, w(M^*) - (k+1)\, w(M)$. Thus $w(M \oplus A_i) \geq \frac{k}{k+1}\, w(M^*)$ for at least one i. ∎

Before moving on we give a little more notation used in both our matching algorithms. If v is matched in M let $M(v) = u$ where $(v,u) \in M$; otherwise $M(v) = v$. A 2-augmentation is *centered* at vertex v if all its non-M edges are incident to v or $M(v)$. Note that every 2-augmentation has at least two centers. Let $A_{\frac{2}{3}}$ be a set of vertex-disjoint 2-augmentations such that $g(A_{\frac{2}{3}}) \geq \frac{3}{5}(\frac{2}{3}w(M^*) - w(M))$; Theorem 2.1 implied the existence of $A_{\frac{2}{3}}$. Let $\mathrm{aug}^*(v)$ be the 2-augmentation in $A_{\frac{2}{3}}$ centered at v (if any) and let $\mathrm{aug}(v)$ be the maximum-gain 2-augmentation centered at v.

3 A Randomized Matching Algorithm

Our randomized matching algorithm can be described very succinctly. Choose a random vertex v and augment the current matching with the highest-gain augmentation centered at v. Repeat as many times as you wish. See Figure 1 for a more formal description.

Figure 1: Algorithm Random-Match(G, k) : G is a graph, k is an integer

0. $M := \emptyset$ (or initialize M to any matching)
1. **repeat** k times:
2. Let $X \in V(G)$ be selected uniformly at random
3. $M := M \oplus \mathrm{aug}(X)$
4. **return** M

We first examine the expected time of Steps 2–3. Let $\deg(v)$ denote the degree of v in G; w.l.o.g. assume deg is strictly positive.

Lemma 3.1 *The time required to find* $\mathrm{aug}(v)$ *is* $\mathcal{O}(\deg(v) + \deg(M(v)))$.

Proof: If v is an isolated vertex in M, i.e., if $M(v) = v$, then finding $\mathrm{aug}(v)$ is trivially accomplished in $\deg(v)$ time. To find the alternating 4-cycles centered at v we first mark all vertices u s.t. $(v, u) \in E \backslash M$. For each edge $(M(v), x)$, if $M(x)$ is marked then $\langle v, M(v), x, M(x), v\rangle$ is an alternating 4-cycle. This procedure runs in $\mathcal{O}(\deg(v) + \deg(M(v)))$ time and clearly finds all alternating 4-cycles.

The procedure for alternating paths is slightly more complicated. An *arm* of v consists of an edge $(v, u) \notin M$ plus $(u, M(u)) \in M$, if it exists. We find the two highest-gain arms of v, P and P', where $g(P) \geq g(P')$. For each arm Q of $M(v)$ we determine the highest-gain 2-augmenting path centered at v that uses Q. This will be $P \cup \{(v, M(v))\} \cup Q$ if $Q \cap P = \emptyset$ and $P' \cup \{(v, M(v))\} \cup Q$ otherwise. Again, this procedure clearly takes $\mathcal{O}(\deg(v) + \deg(M(v)))$ time and detects the best 2-augmenting path centered at v. ∎

Lemma 3.1 is essentially the same as Theorem 2 in [2]. We now examine the expected performance of Random-Match.

Lemma 3.2 *If* $v \in V$ *is chosen uniformly at random then*

$$\mathbb{E}[g(\mathrm{aug}(v))] \;\geq\; \frac{6}{5n}\left(\frac{2}{3}w(M^*) - w(M)\right)$$

Proof: Let $V_{\frac{2}{3}}$ be the set of center vertices for the augmenting paths/cycles in $A_{\frac{2}{3}}$. Note that $|V_{\frac{2}{3}}| \geq 2 \cdot |A_{\frac{2}{3}}|$ since every 2-augmentation has at least two centers.

$$\begin{aligned}\mathbb{E}[g(\text{aug}(v))] &\geq \Pr[v \in V_{\frac{2}{3}}] \cdot \mathbb{E}[g(\text{aug}(v)) \mid v \in V_{\frac{2}{3}}] \\ &\geq \sum_{v \in V_{\frac{2}{3}}} \frac{g(\text{aug}^*(v))}{n} \geq \frac{6}{5n}\left(\frac{2}{3}w(M^*) - w(M)\right)\end{aligned}$$

∎

Lemma 3.3, given below, shows that by repeating the randomized augmentation step n times we obtain an expected geometric decrease in the gap between $w(M)$ and $\frac{2}{3}w(M^*)$.

Lemma 3.3 *The expected weight of M after k iterations of Steps 2–3 is at least $\frac{2}{3}w(M^*)(1-e^{-6k/5n})$.*

Proof: Let $\tilde{w} = \frac{2}{3}w(M^*)$ and let Y_i be the weight of M after i iterations. Clearly $Y_0 = 0$ and by Lemma 3.2, $\mathbb{E}[Y_{i+1}] \geq Y_i + \frac{6}{5}(\tilde{w} - Y_i)/n$. By linearity of expectation we have the more usable inequality $\mathbb{E}[Y_{i+1}] \geq \mathbb{E}[Y_i] + \frac{6}{5}(\tilde{w} - \mathbb{E}[Y_i])/n$. Assuming inductively that $\mathbb{E}[Y_i] \geq \tilde{w} \cdot (1 - e^{-6i/5n})$ (it holds for $i = 0$), we have:

$$\begin{aligned}\mathbb{E}[Y_{i+1}] &\geq \tilde{w} \cdot (1 - e^{-6i/5n}) + 6\tilde{w} \cdot e^{-6i/5n}/5n \\ &= \tilde{w} \cdot (1 - (1 - 6/5n)e^{-6i/5n}) \geq \tilde{w} \cdot (1 - e^{-6(i+1)/5n})\end{aligned}$$

∎

Theorem 3.4 *In expected time $O(m \log \frac{1}{\epsilon})$ Random-Match returns a matching whose expected weight is at least $\frac{2}{3} - \epsilon$ that of the maximum weight matching.*

Proof: The Theorem follows by setting $k = \frac{5}{6}n \ln \frac{1}{\epsilon}$. By Lemma 3.3 the expected weight of the returned matching is $\frac{2}{3}(1 - e^{-6k/5n}) > \frac{2}{3} - \epsilon$ times the optimum. By Lemma 3.1 the time for Steps 2 and 3 is proportional to $\deg(X) + \deg(M(X))$. Averaged over all $X \in V(G)$ the expected time per iteration is $4m/n$. ∎

Remark. One can also show that $\mathbb{E}[\text{aug}(v)] \geq 3(\frac{2}{3}w(M^*) - w(M))/n$, which gives a faster rate of convergence.

4 A Deterministic Algorithm

The deterministic algorithm operates in phases, each taking linear time. If M and M' are the matchings before and after some phase we guarantee that $w(M') \geq w(M) + c \cdot (\frac{2}{3}w(M^*) - w(M))$, for a constant c. Therefore, executing $O(\log \frac{1}{\epsilon})$ phases yields a $\frac{2}{3} - \epsilon$ matching.

In each phase we generate a set of at most n candidate augmentations (one centered at each vertex) then choose from this set, in a greedy manner, a subset of non-overlapping augmentations. In the analysis we show that the total gain of the augmentations selected is $\Omega(g(A_{\frac{2}{3}})) = \Omega(\frac{2}{3}w(M^*) - w(M))$. The obvious ways to choose the candidate set can perform very poorly in the worst case. For instance, if we choose $\{\text{aug}(v) : v \in V(G)\}$, or in general the best k augmentations centered at each vertex, it is impossible to guarantee a gain of $\Omega(g(A_{\frac{2}{3}}))$.

Recall that nearly all definitions are with respect to the matching M. An *element* is either a matched edge or an unmatched vertex. We will think of augmentations as either being sets of elements or sets of edges, whichever is more convenient. If e is an element and S a set of augmentations then $S(e)$ is the maximum-gain augmentation in S that contains e, if any. If $a = \{e_1, e_2, \dots\}$ is a set of elements (e.g., an augmentation) then $S(a) = \{S(e_1), S(e_2), \dots\}$. An arm[3] r of v is *eligible* if $g(r) \geq \gamma \cdot g(S(r))$, for a constant $\gamma > 1$ to be specified later. For an augmentation a to be eligible it must satisfy $g(a) \geq \gamma \cdot g(S(a))$. In addition, if a is a 3-element augmentation centered at $(u, M(u))$ then the arms of u and $M(u)$ in a must also be eligible.

We denote by greedy(S) a subset of the augmentations S selected by the greedy algorithm. Specifically the algorithm repeatedly selects the maximum-gain augmentation in S that is vertex/element disjoint with previously chosen augmentations.

Figure 2: Algorithm Deterministic-Match(G, k) : G is a graph, k an integer.

0. $M := \emptyset$
1. **repeat** k times: *(Lines 2–6 = 1 Phase)*
2. $S := \emptyset$
3.   **foreach** element e *(Either $e \in M$ or $e \in V \setminus \bigcup_{c \in M} c$)*
4.   Find an eligible augm. a centered at e maximizing $g(a)$
5. $S := S \cup \{a\}$
6. $M := M \oplus \text{greedy}(S)$
7. **return** M

Theorem 4.1 *Deterministic-Match (Figure 2) runs in $O(km)$ time and returns a matching weighing at least $\frac{2}{3}(1 - (\frac{19}{20})^k)$ of the maximum weight matching.*

Proof: Let $a(e)$ be the augmentation centered at e that is selected in Line 4 (if any), and let $S(a(e))$ and $S(\text{aug}^*(e))$ be w.r.t. the set S when e is considered in Line 4. In isolation S refers to the set S after the phase, at Line 6. We will prove that after every phase of the algorithm the following two inequalities hold.

$$g(S) \geq g(A_{\frac{2}{3}})/3\gamma \tag{1}$$

$$g(\text{greedy}(S)) \geq (\gamma - 1)g(S)/\gamma \tag{2}$$

We obtain the sharpest bound by setting $\gamma = 2$, giving: $g(\text{greedy}(S)) \geq (\gamma - 1)g(A_{\frac{2}{3}})/3\gamma^2 = \frac{1}{20}(\frac{2}{3}w(M^*) - w(M))$.

[3]Recall that an arm consists of an unmatched edge (v, u) plus the matched edge $(u, M(u))$ if it exists.

We now consider (1). If $g(\text{aug}^*(e)) \geq \gamma \cdot g(S(\text{aug}^*(e)))$ then either $\text{aug}^*(e)$ was eligible when e was considered at Line 4, implying $g(a(e)) \geq g(\text{aug}^*(e))$, or it was ineligible. If $\text{aug}^*(e)$ were an ineligible two-element augmentation then $g(\text{aug}^*(e)) \leq \gamma \cdot g(S(\text{aug}^*(e)))$. For the last case let $\text{aug}^*(e) = \{r_1, e, r_2\}$ be an ineligible three-element augmentation, where r_1, r_2 are arms. One can see that $\{r_1, e\}$ must be an eligible augmentation and r_2 an ineligible arm (or the reverse), implying that $g(\{r_1, e\}) \leq g(a(e))$ and $g(r_2) \leq \gamma \cdot g(S(r_2))$, and therefore, that $g(\{r_1, e, r_2\}) = g(\text{aug}^*(e)) \leq g(a(e)) + \gamma \cdot g(S(r_2))$. Combining all cases we have:

$$g(\text{aug}^*(e)) \;\leq\; \gamma \cdot g(S(\text{aug}^*(e))) + g(a(e)) \tag{3}$$

Notice that $a(e)$ and each element of $S(\text{aug}^*(e))$ share at least one element with $\text{aug}^*(e)$. Let C be a set of centers representing the augmentations in $A_{\frac{2}{3}}$, with $|C| = |A_{\frac{2}{3}}|$. Each augmentation in S can appear on the right side of (3) for at most three distinct $e \in C$ since all augmentations in $A_{\frac{2}{3}}$ are element-disjoint. Summing (3) over $e \in C$ we have:

$$g(A_{\frac{2}{3}}) = \sum_{e \in C} g(\text{aug}^*(e)) \leq \sum_{e \in C} \gamma \cdot g(S(\text{aug}^*(e))) + g(a(e)) \leq 3\gamma \cdot g(S)$$

Before turning to (2) we make a few observations. If an augmentation $a(e)$ is added to S in Line 5 we will say $a(e)$ *supercedes* each element of $S(a(e))$. We claim that if a_1 and a_2 both supercede a_3, where $a_1, a_2, a_3 \in S$, then $a_1 \cap a_2 \cap a_3 = \emptyset$. Suppose that a_1 was added to S before a_2. Because $a_3 \in S(a_2)$, a_3 must be the maximum-gain augmentation already in S that intersects a_2. Let $e \in s_2 \cap s_3$. Since $g(a_1) > g(a_3)$, it must be that $e \notin a_1$. This observation implies that in the acyclic graph $\mathcal{S} = (S, \{(a, a') : a \text{ supercedes } a'\})$, the subset $\{a : e \in a \in S\}$ forms a directed path, for any element e. Moreover, the in and out degree of vertices in $\mathcal{S}$ are both bounded by 3.

Suppose that $a \in S$ was selected by $\text{greedy}(S)$. This removes from consideration any other element b for which $a \cap b \neq \emptyset$. Let $A_0 = \{a\}$ and let A_i be those augmentations in S superceded by some augmentation in A_{i-1}. Finally let $A = \bigcup_i A_i$. It follows from the observations above that A are exactly those elements removed from consideration by the selection of a. By the definition of eligibility $g(A_i) \leq g(A_{i-1})/\gamma$. Therefore:

$$g(A) \;\leq\; \sum_{i=0}^{\infty} g(A_i) \;\leq\; g(a) \cdot \sum_{i=0}^{\infty} \gamma^{-i} \;=\; \frac{\gamma}{\gamma - 1} \cdot g(a) \tag{4}$$

Summing over all $a \in \text{greedy}(S)$ we have $g(\text{greedy}(S)) \geq (\gamma - 1)\gamma^{-1} \cdot g(S)$, which proves (2).

One can readily see that $O(m)$ time suffices to compute the set S in each phase; see Lemma 3.1 for the details. The set $\text{greedy}(S)$ can be computed in $O(n)$ time by performing a topological sort of the acyclic graph $\mathcal{S}$. ∎

5 Arbitrarily Good Approximations

Both our randomized and deterministic algorithms can be generalized in striaghtforward ways to yield δ-MWM algorithms, for any δ. The penalty for setting $\delta \geq 2/3 - o(1)$ is super-linear running time. Nonetheless, our algorithms are faster than the previous best [6] on sparse to moderately dense graphs.

Theorem 5.1 *There is a $(1 - 1/k - \epsilon)$-MWM algorithm running in time $O(n(\Delta - 1)^{k-2} \log \epsilon^{-1})$, where $\epsilon > 0$, $k \geq 3$, $\Delta > 1$ is the maximum degree, and n the number of vertices.*

The time bound follows from a generalized version of Lemma 3.1. One can easily show that the best $(k-1)$-augmentation centered[4] at any vertex can be found in $O((\Delta - 1)^{k-2})$ time. For regular graphs the time bound of Theorem 5.1 improves on [5, 6] for all k, whenever $\Delta = O(n^{1/2(k-3)})$.

Acknowledgment. We would like to thank Stefan Hougardy for pointing out an error in an earlier version of Theorem 2.1.

References

[1] D. Drake and S. Hougardy. Improved linear time approximation algorithms for weighted matchings. In *7th International Workshop on Randomization and Approximation Techniques in Computer Science (APPROX), LNCS 2764*, pages 14–23, 2003.

[2] D. Drake and S. Hougardy. Linear time local improvements for weighted matchings in graphs. In *International Workshop on Experimental and Efficient Algorithms (WEA), LNCS 2647*, pages 107–119, 2003.

[3] D. Drake and S. Hougardy. A simple approximation algorithm for the weighted matching problem. *Info. Proc. Lett.*, 85:211–213, 2003.

[4] H. N. Gabow. Data structures for weighted matching and nearest common ancestors with linking. In *First Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 434–443, 1990.

[5] H. N. Gabow and R. E. Tarjan. Faster scaling algorithms for network problems. *SIAM J. Comput.*, 18(5):1013–1036, 1989.

[6] H. N. Gabow and R. E. Tarjan. Faster scaling algorithms for general graph-matching problems. *J. ACM*, 38(4):815–853, 1991.

[7] John E. Hopcroft and Richard M. Karp. An $n^{5/2}$ algorithm for maximum matchings in bipartite graphs. *SIAM J. Comput.*, 2:225–231, 1973.

[8] Preis. Linear time 1/2-approximation algorithm for maximum weighted matching in general graphs. In *Proc. 16th Ann. Symp. on Theoretical Aspects of Computer Science (STACS), LNCS 1563*, pages 259–269, 1999.

[4] An augmenting cycle is centered at any of its elements; an augmenting path at the second element in the path.

I N F O R M A T I K

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`

8

MPI-I-2004-1-002	P. Sanders, S. Pettie	?
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. Nürnberger, H. Seidel	Visualization of Volume Data with Quadratic Super Splines
MPI-I-2003-4-005	T. Hangelbroek, G. Nürnberger, 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. 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. Schäfer, S. Leonardi	Cross-Monotonic Cost Sharing Methods for Connected Facility Location Games
MPI-I-2003-1-016	G. Schäfer, N. Sivadasan	Topology Matters: Smoothed Competitive Analysis of Metrical Task Systems
MPI-I-2003-1-015	A. Kovács	Sum-Multicoloring on Paths
MPI-I-2003-1-014	G. Schäfer, 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 -
MPI-I-2003-1-011	P. Krysta, A. Czumaj, B. Voecking	Selfish Traffic Allocation for Server Farms
MPI-I-2003-1-010	H. Tamaki	A linear time heuristic for the branch-decomposition of planar graphs
MPI-I-2003-1-009	B. Csaba	On the Bollobás – Eldridge conjecture for bipartite graphs
MPI-I-2003-1-008	P. Sanders	Polynomial Time Algorithms for Network Information Flow
MPI-I-2003-1-007	H. Tamaki	Alternating cycles contribution: a strategy of tour-merging for the traveling salesman problem
MPI-I-2003-1-006	M. Dietzfelbinger, H. Tamaki	On the probability of rendezvous in graphs
MPI-I-2003-1-005	M. Dietzfelbinger, P. Woelfel	Almost Random Graphs with Simple Hash Functions
MPI-I-2003-1-004	E. Althaus, T. Polzin, S.V. Daneshmand	Improving Linear Programming Approaches for the Steiner Tree Problem
MPI-I-2003-1-003	R. Beier, B. Vöcking	Random Knapsack in Expected Polynomial Time
MPI-I-2003-1-002	P. Krysta, P. Sanders, B. Vöcking	Scheduling and Traffic Allocation for Tasks with Bounded Splittability
MPI-I-2003-1-001	P. Sanders, R. Dementiev	Asynchronous Parallel Disk Sorting
MPI-I-2002-4-002	F. Drago, W. Martens, K. Myszkowski, H. Seidel	Perceptual Evaluation of Tone Mapping Operators with Regard to Similarity and Preference
MPI-I-2002-4-001	M. Goesele, J. Kautz, J. Lang, H.P.A. Lensch, H. Seidel	Tutorial Notes ACM SM 02 A Framework for the Acquisition, Processing and Interactive Display of High Quality 3D Models
MPI-I-2002-2-008	W. Charatonik, J. Talbot	Atomic Set Constraints with Projection
MPI-I-2002-2-007	W. Charatonik, H. Ganzinger	Symposium on the Effectiveness of Logic in Computer Science in Honour of Moshe Vardi
MPI-I-2002-1-008	P. Sanders, J.L. Träff	The Factor Algorithm for All-to-all Communication on Clusters of SMP Nodes
MPI-I-2002-1-005	M. Hoefer	Performance of heuristic and approximation algorithms for the uncapacitated facility location problem

MPI-I-2002-1-004	S. Hert, T. Polzin, L. Kettner, G. Schäfer	Exp Lab A Tool Set for Computational Experiments
MPI-I-2002-1-003	I. Katriel, P. Sanders, J.L. Träff	A Practical Minimum Scanning Tree Algorithm Using the Cycle Property
MPI-I-2002-1-002	F. Grandoni	Incrementally maintaining the number of l-cliques
MPI-I-2002-1-001	T. Polzin, S. Vahdati	Using (sub)graphs of small width for solving the Steiner problem
MPI-I-2001-4-005	H.P.A. Lensch, M. Goesele, H. Seidel	A Framework for the Acquisition, Processing and Interactive Display of High Quality 3D Models
MPI-I-2001-4-004	S.W. Choi, H. Seidel	Linear One-sided Stability of MAT for Weakly Injective Domain
MPI-I-2001-4-003	K. Daubert, W. Heidrich, J. Kautz, J. Dischler, H. Seidel	Efficient Light Transport Using Precomputed Visibility
MPI-I-2001-4-002	H.P.A. Lensch, J. Kautz, M. Goesele, H. Seidel	A Framework for the Acquisition, Processing, Transmission, and Interactive Display of High Quality 3D Models on the Web
MPI-I-2001-4-001	H.P.A. Lensch, J. Kautz, M. Goesele, W. Heidrich, H. Seidel	Image-Based Reconstruction of Spatially Varying Materials
MPI-I-2001-2-006	H. Nivelle, S. Schulz	Proceeding of the Second International Workshop of the Implementation of Logics
MPI-I-2001-2-005	V. Sofronie-Stokkermans	Resolution-based decision procedures for the universal theory of some classes of distributive lattices with operators
MPI-I-2001-2-004	H. de Nivelle	Translation of Resolution Proofs into Higher Order Natural Deduction using Type Theory
MPI-I-2001-2-003	S. Vorobyov	Experiments with Iterative Improvement Algorithms on Completely Unimodel Hypercubes
MPI-I-2001-2-002	P. Maier	A Set-Theoretic Framework for Assume-Guarantee Reasoning