A Simple Reduction from Maximum Weight Matching to Maximum Cardinality Matching[☆]

S. Pettie[a]

[a] *University of Michigan, Dept. of Electrical Engineering and Computer Science, 2260 Hayward St., Ann Arbor, MI 48109*

Abstract

Let $\textsc{mcm}(m,n)$ and $\textsc{mwm}(m,n,N)$ be the complexities of computing a maximum cardinality matching and a maximum weight matching, and let $\textsc{mcm}_{\mathrm{bi}}, \textsc{mwm}_{\mathrm{bi}}$ be their counterparts for bipartite graphs, where m, n, and N are the edge count, vertex count, and maximum integer edge weight. Kao, Lam, Sung, and Ting [1] gave a general reduction showing $\textsc{mwm}_{\mathrm{bi}}(m,n,N) = O(N \cdot \textsc{mcm}_{\mathrm{bi}}(m,n))$ and Huang and Kavitha [2] recently proved the analogous result for general graphs, that $\textsc{mwm}(m,n,N) = O(N \cdot \textsc{mcm}(m,n))$.

We show that Gabow's $\textsc{mwm}_{\mathrm{bi}}$ and $\textsc{mwm}$ algorithms from 1983 and 1985 [3, 4] can be modified to replicate the results of Kao et al. and Huang and Kavitha, but with dramatically simpler proofs. We also show that our reduction leads to new bounds on the complexity of $\textsc{mwm}$ on sparse graph classes, e.g., (bipartite) planar graphs, bounded genus graphs, and H-minor-free graphs.

Keywords: Graph algorithms, maximum matching

1. Introduction

We are given an integer-weighted graph $G = (V, E, w)$ and asked to find a maximum weight matching ($\textsc{mwm}$), that is, a set of vertex-disjoint edges M for which $\sum_{e \in M} w(e)$ is maximized. This problem is distinct from, but closely related to, the problem of finding a maximum (or minimum) weight *perfect* matching ($\textsc{mwpm}$), in which all vertices are matched. There are simple reductions between these two problems (see [5, 6]) showing that $\textsc{mwm}(m,n,N) = O(\textsc{mwpm}(2m+n, 2n, N))$ and $\textsc{mwpm}(m,n,N) = O(\textsc{mwm}(m,n,nN))$. Note that the first reduction preserves the graph parameters but the second blows up the maximum edge weight.

The complexity of the $\textsc{mwm}$ and $\textsc{mwpm}$ problems depend on the graph density, the relative sizes of N and n, the exponent ω of square matrix multiplication, the word size $w = \Omega(\log n)$, and the complexity of maximum cardinality matching ($\textsc{mcm}$). For both bipartite and general graphs we have $\textsc{mcm}(m,n), \textsc{mcm}_{\mathrm{bi}}(m,n) = O(m\sqrt{n}\frac{\log(n^2/m)}{\log n})$ (deterministically) and $O(n^\omega)$ (randomized) [7–11].[1]. Furthermore, on bipartite graphs $\textsc{mcm}_{\mathrm{bi}}(m,n) = O(n^2 + n^{5/2}/w)$ (deterministically) [18], which is faster on dense graphs with $w = \omega(\log n)$. The running times of the best weighted matching algorithms are given below. See [6] for a more detailed discussion of these and other matching algorithms. The citations [19–21] are integer priority queues, which can be used to efficiently implement the Hungarian algorithm.

[☆] This work is supported by NSF CAREER grant no. CCF-0746673 and a grant from the US-Israel Binational Science Foundation.

Email address: `pettie@umich.edu, Ph: +1 (734) 972 0375, Fax: +1 (734) 763-1260` (S. Pettie)

[1] Note that the first bound improves on the older $O(m\sqrt{n})$-time algorithms of Hopcroft and Karp [12], Dinic and Karzanov [13, 14], Micali and Vazirani [15, 16], and Gabow and Tarjan [17] only when $m = n^{2-o(1)}$.

$$\textsc{mwm}_{\text{bi}}(m,n,N) = \begin{cases} O(mn + n^2 \log\log n) \;\; \text{(indep. of } N\text{)} & [19, 20] \\ O(mn) \;\; \text{(rand., indep. of } N\text{)} & [21] \\ O(m\sqrt{n}\log N) & [5, 6] \\ O(Nn^\omega) \;\; \text{(rand.)} & [22] \\ O(N \cdot \textsc{mcm}_{\text{bi}}(m,n)) & [1] \\ \quad = \begin{cases} O(N \cdot m\sqrt{n}\frac{\log(n^2/m)}{\log n}) & [9] \\ O(N \cdot (n^2 + n^{5/2}/w)) & [18] \\ O(N \cdot n^\omega) \;\; \text{(rand.)} & [10] \end{cases} \end{cases}$$

$$\textsc{mwpm}_{\text{bi}}(m,n,N) = \begin{cases} O(mn + n^2 \log\log n) \;\; \text{(indep. of } N\text{)} & [19, 20] \\ O(mn) \;\; \text{(rand., indep. of } N\text{)} & [21] \\ O(m\sqrt{n}\log(nN)) & [5, 6, 8, 23, 24] \\ O(Nn^\omega) \;\; \text{(rand.)} & [22] \end{cases}$$

$$\textsc{mwm}(m,n,N) = \begin{cases} O(mn + n^2 \log n) \;\; \text{(indep. of } N\text{)} & [25] \\ O(m\sqrt{n\log n}\log(nN)) & [17] \\ O(N \cdot \textsc{mcm}(m,n)) & [2] \\ \quad = \begin{cases} O(N \cdot m\sqrt{n}\frac{\log(n^2/m)}{\log n}) & [7] \\ O(Nn^\omega) \;\; \text{(rand.)} & [10, 11] \end{cases} \end{cases}$$

$$\textsc{mwpm}(m,n,N) = \begin{cases} O(mn + n^2 \log n) \;\; \text{(indep. of } N\text{)} & [25] \\ O(m\sqrt{n\log n}\log(nN)) & [17] \end{cases}$$

In the mid-1980s Gabow introduced the *scaling* technique to the weighted matching problem and gave MWPM algorithms for both bipartite [3] and general graphs [4] running in $O(mn^{3/4}\log N)$ time. In a generally overlooked passage [3, pp. 159–160] Gabow noted that his $\textsc{mwpm}_{\text{bi}}$ algorithm for bipartite graphs could be modified to solve $\textsc{mwm}_{\text{bi}}$ in $O(Nm\sqrt{n})$ time, and stated without proof that the same bound could be obtained for MWM on general graphs. Using a rather different approach, Kao et al. [1] proved that $\textsc{mwm}_{\text{bi}}$ could be solved with N black-box applications of a bipartite maximum cardinality matching algorithm. This improved on Gabow's algorithm[2] when the graph is somewhat dense. Very recently Huang and Kavitha [2] generalized Kao et al.'s reduction to general graphs.

New Results. In this paper we provide a simplified presentation of Gabow's original algorithms and show that they can be expressed as reductions from $\textsc{mwm}/\textsc{mwm}_{\text{bi}}$ to N executions of $\textsc{mcm}/\textsc{mcm}_{\text{bi}}$. The resulting algorithms and proofs of correctness are dramatically simpler than those of Kao et al. [1] and Huang and Kavitha [2]. Our reduction (and those of [1, 2]) also work on all minor-closed graph classes. Together with the cardinality matching algorithms of Mucha and Sankowski [26], Yuster and Zwick [27], and Borradaile et al. [28], our reduction yields new MWM algorithms running in time $O(N \cdot n^{\omega/2})$ on bounded genus and planar graphs, $O(N \cdot n^{3\omega/(3+\omega)})$ on H-minor-free graphs, and $O(N \cdot n\log^3 n)$ on bipartite planar graphs.

2. Preliminaries

2.1. The Maximum Weight Matching LP

Let $\mathcal{V}_{\text{odd}}$ be the set of all odd-size subsets of $V(G)$. Edmonds [29, 30] proved that the basic solutions to the following LPs are integral. More specifically, if M is a MWM and x its incidence vector ($x(e) = 1$ if

[2] Kao et al. do not cite Gabow's $O(Nm\sqrt{n})$-time algorithm. They compare their time bound of $N \cdot \textsc{mcm}_{\text{bi}}(m,n)$ against Gabow and Tarjan's bound of $O(m\sqrt{n}\log(nN))$.

$e \in M$, 0 if $e \notin M$) then x is optimal for (1):

$$\begin{array}{lll} \text{maximize} & \displaystyle\sum_{e \in E(G)} w(e)x(e) & \\ \text{subject to} & 0 \le x(e) \le 1 & \forall e \in E(G) \\ & \displaystyle\sum_{e=(u,u') \in E(G)} x(e) \le 1 & \forall u \in V(G) \\ & \displaystyle\sum_{e=(u,v)\,:\, u,v \in B} x(e) \le (|B|-1)/2 & \forall B \in \mathcal{V}_{\text{odd}} \end{array} \tag{1}$$

The dual of (1) is given below, where $y : V(G) \to \mathbb{R}$ and $z : \mathcal{V}_{\text{odd}} \to \mathbb{R}$ are the dual variables for vertices and odd sets.

$$\begin{array}{lll} \text{minimize} & \displaystyle\sum_{u \in V(G)} y(u) + \sum_{B \in \mathcal{V}_{\text{odd}}} z(B)(|B|-1)/2 & \\ \text{subject to} & yz(e) \ge w(e) & \forall e \in E(G) \\ & y(u) \ge 0 & \forall u \in V(G) \\ & z(B) \ge 0 & \forall B \in \mathcal{V}_{\text{odd}} \\ \text{where, by definition,} & yz(u,v) \stackrel{\text{def}}{=} y(u) + y(v) + \displaystyle\sum_{B \in \mathcal{V}_{\text{odd}}\,:\, u,v \in B} z(B) & \end{array} \tag{2}$$

2.2. Matchings and Blossoms

An *alternating* path or cycle w.r.t. a matching M is one whose edges alternate between M and $E(G)\backslash M$. An alternating path is *augmenting* if it begins and ends at free vertices. If M is a matching and P an augmenting path, $M \oplus P = (M\backslash P) \cup (P\backslash M)$ is a matching with $|M \oplus P| = |M| + 1$.

Blossoms are formed inductively as follows. A trivial blossom consists of a singleton vertex set $\{v\}$ and no edges. Suppose $A_0, \ldots, A_{\ell-1}$ are vertex sets containing blossoms $E_{A_0}, \ldots, E_{A_{\ell-1}}$. If there exist edges $e_0, \ldots, e_{\ell-1}$ where $e_i \in A_i \times A_{i+1}$ (modulo ℓ) and $e_i \in M$ if and only if i is odd, then $B = \bigcup_{0 \le i < \ell} A_i$ is a vertex set containing the blossom $E_B = \bigcup_{0 \le i < \ell} E_{A_i} \cup \{e_0, \ldots, e_{\ell-1}\}$. The unique unmatched vertex in E_B is called the *base* of B. See Figure 1 for an example. Matching algorithms usually maintain a dynamically changing matching M together with a hierarchically nested set Ω of weighted blossoms (those assigned non-zero z-values). We say M *respects* Ω if for each $B \in \Omega$, $|E_B \cap M| = (|B|-1)/2$. Note that the graph induced by B generally contains more edges than E_B.

The blossom set Ω is represented as a forest of rooted trees. Leaves correspond to trivial blossoms (vertices) and root blossoms are those not contained in any other blossom. Using the terminology above, if B is a node with children $A_0, \ldots, A_{\ell-1}$, B stores a pointer to the child containing the base of B and each node A_i keeps a pointer to the successor edge $e_i \in A_i \times A_{i+1}$. We often refer to a blossom by its vertex set, e.g., $B \in \Omega$ asserts that some blossom E_B on $B \subseteq V$ is in Ω.

Blossoms can often be treated like single vertices. Let G/Ω be the graph obtained by contracting all root blossoms in Ω. Observe that if M is a matching in G, M/Ω is also a matching in G/Ω. If P' is an augmenting path in G/Ω then P' extends to an augmenting path P in G, that is, P is obtained by substituting a path through E_B for each non-trivial root blossom B in P'. Furthermore, the augmented matching $M \oplus P$ still respects Ω, though augmentation can change the bases of some blossoms in Ω. See Figure 1.

Property 2.1 lists the standard complementary slackness invariants for MWM, which are maintained throughout the algorithm described in Section 3.

Property 2.1. Recall that $yz(u,v) = y(u) + y(v) + \sum_{B \in \mathcal{V}_{\text{odd}}\,:\, u,v \in B} z(B)$ is defined to be the dual of edge (u,v). Let M be a matching respecting Ω.

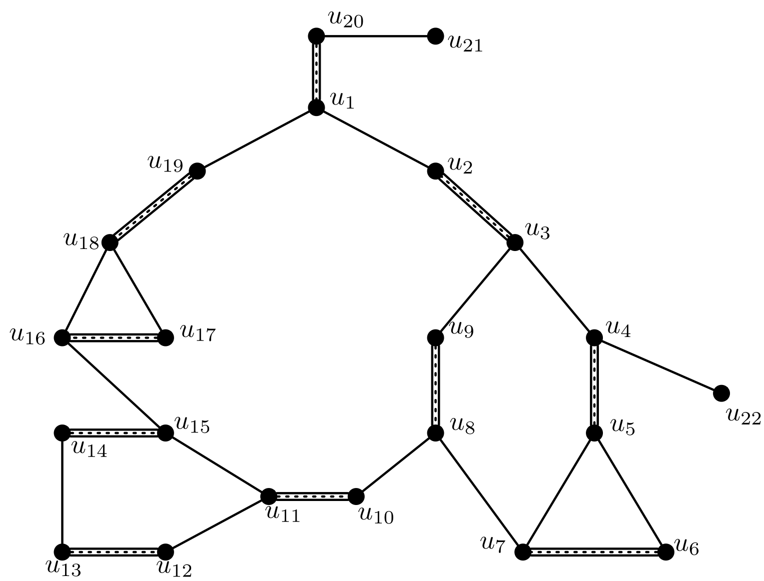


Figure 1: Matched edges in M are thick and dashed, unmatched edges thin. A blossom $B_1 = (u_1, u_2, B_4, u_{10}, B_3, B_2, u_{19})$ contains blossoms $B_2 = (u_{18}, u_{16}, u_{17})$, $B_3 = (u_{11}, u_{12}, u_{13}, u_{14}, u_{15})$ and $B_4 = (u_3, u_4, B_5, u_8, u_9)$, which contains the blossom $B_5 = (u_5, u_6, u_7)$. Augmenting along the path $(u_{21}, u_{20}, B_1, u_{22})$ in G/B_1, corresponding to the path $(u_{21}, u_{20}, u_1, u_2, u_3, u_9, u_8, u_7, u_6, u_5, u_4, u_{22})$ in G, repositions the bases of B_1 and B_4 to u_4, and the base of B_5 to u_7.

1. Non-negativity. All y- and z-values are non-negative. The set Ω consists of nested blossoms, and includes all blossoms with non-zero z-values. Root blossoms in Ω have non-zero z-values, though non-root blossoms may have zero z-values.
2. Domination. $yz(e) \geq w(e)$ for all $e \in E(G)$.
3. Tightness. If $e \in M \cup \bigcup_{B \in \Omega} E_B$ then $yz(e) = w(e)$.
4. Free vertices. The y-values of free vertices are equal and strictly less than all matched vertices.

Lemma 2.2 is well known. See [6] for a short proof of it and related statements.

Lemma 2.2. *If Property 2.1 holds for a matching M and the y-values of free vertices are zero, then M is a maximum weight matching.*

3. A Maximum Weight Matching Algorithm

What follows is a simplified presentation of Gabow's algorithm [3, 4], using the notation and terminology from [6].

Initialization. Initially M and Ω are empty and $y(u) = N/2$ for all $u \in V(G)$. This clearly satisfies Property 2.1: the non-negativity, domination, and free vertex conditions are immediate and the tightness condition is vacuous. Let $G_{\text{tight}} = (V, E_{\text{tight}})$ be the *tight subgraph*, where $E_{\text{tight}} = \{e \in E(G) \mid yz(e) = w(e)\}$. By Property 2.1, $M \cup \bigcup_{B \in \Omega} E_B \subseteq E_{\text{tight}}$. Note that after initialization all edges with weight N are tight.

After initialization we repeatedly execute *Augmentation*, *Blossom Formation*, *Dual Adjustment*, and *Blossom Dissolution* steps until the y-values of free vertices are zero. This requires exactly N iterations. Figure 2 illustrates steps of Augmentation and Blossom Formation.

Augmentation. Extend M to a maximum cardinality matching in G_{tight} respecting Ω, that is, *matched vertices must remain matched.* This is tantamount to extending M/Ω to a maximum cardinality matching on G_{tight}/Ω. The matched edges inside a blossom are determined by the matching on G_{tight}/Ω.

Blossom Formation. Let $V_{\rm out} \subseteq V(G_{\rm tight}/\Omega)$ be the vertices of $G_{\rm tight}/\Omega$ reachable from free vertices by even-length alternating paths and let $V_{\rm in} \subseteq V(G_{\rm tight}/\Omega)\backslash V_{\rm out}$ be the non-$V_{\rm out}$ vertices reachable from free vertices by odd-length alternating paths. Let Ω' be a maximal set of possibly nested blossoms on $V_{\rm out}$, that is, if $(u,v) \in E(G_{\rm tight}/\Omega)$ and $u,v \in V_{\rm out}$ then u and v must belong to a common blossom in Ω'. Set $z(B) \leftarrow 0$ for all $B \in \Omega'$ and set $\Omega \leftarrow \Omega \cup \Omega'$.

Dual Adjustment. Let $\hat{V}_{\rm in}, \hat{V}_{\rm out} \subseteq V(G)$ be original vertices represented by vertices (that is, root blossoms) in $V_{\rm in}$ and $V_{\rm out}$. The y- and z-values for some vertices and root blossoms are adjusted:

$$y(u) \leftarrow y(u) - 1/2, \text{ for all } u \in \hat{V}_{\rm out}.$$
$$y(u) \leftarrow y(u) + 1/2, \text{ for all } u \in \hat{V}_{\rm in}.$$
$$z(B) \leftarrow z(B) + 1, \text{ for all root blossoms } B \in \Omega \text{ with } B \subseteq \hat{V}_{\rm out}.$$
$$z(B) \leftarrow z(B) - 1, \text{ for all root blossoms } B \in \Omega \text{ with } B \subseteq \hat{V}_{\rm in}.$$

Blossom Dissolution. After Dual Adjustment some root blossoms may have zero z-values. Remove such blossoms from Ω as long as they exist. Note that non-root blossoms can have zero z-values and should not be removed.

The Blossom Dissolution step can be implemented in $O(n)$ time by traversing the forest representing Ω, repeatedly removing roots whose z-values are zero.

The correctness of the algorithm follows from the fact that Property 2.1 is maintained after every iteration; see [6, Lemmas 3–5] for a short proof.[3] Note that the Blossom Dissolution step is critical. If this step is not performed then the z-values of blossoms may become negative in a subsequent Dual Adjustment step, which would not allow us to apply Lemma 2.2.

Since free vertices have their y-values decremented by 1/2 in each iteration, there are exactly N iterations until their y-values are zero. By Lemma 2.2 the resulting matching is a maximum weight matching. It is crucial that matched vertices do not become unmatched in the Augmentation step, for otherwise free vertices will not have equal and minimal y-values.

3.1. *Implementation and Efficiency*

The Blossom Formation step is easy to implement in $O(m)$ time using depth first search.[4] The Dual Adjustment and Blossom Dissolution steps are easy to implement in $O(m)$ time. For the latter, we process the forest representing Ω, repeatedly removing roots with zero z-values.

We can implement the Augmentation step using any maximum cardinality matching algorithm in a black-box fashion. First, find a maximum cardinality matching M' of $G_{\rm tight}$ respecting Ω, in $O(\textsc{mcm}(m,n))$ time. (I.e., find a maximum cardinality matching of $G_{\rm tight}/\Omega$ and extend it to a maximum cardinality matching of $G_{\rm tight}$.) The graph $M' \oplus M$ consists of even-length alternating cycles, even-length alternating paths (connecting a free vertex w.r.t. M to a free vertex w.r.t. M') and odd-length augmenting paths w.r.t. M. Let $P \subseteq M' \oplus M$ be the union of the augmenting paths. We set $M \leftarrow M \oplus P$. It follows that M is now a maximum cardinality matching respecting Ω and that matched vertices remain matched after augmentation.

Clearly $\textsc{mcm}(m,n) = \Omega(m)$. Thus, the dominant cost in this algorithm is the N applications of some MCM algorithm. Of course, if the original graph is bipartite then we never have to consider blossoms and can instead use any $\textsc{mcm}_{\rm bi}$ algorithm in the Augmentation step. Theorem 3.1 follows.

---

[3] The proof is merely a case analysis, the cases of which depend on whether zero, one, or both endpoints of an edge e are in $\hat{V}_{\rm out} \cup \hat{V}_{\rm in}$, whether e is in M or not, and whether e is tight or not. The only subtlety of the proof is this. It is conceivable that before Dual Adjustment $(u,v) \notin M$ is not tight and $yz(u,v) = w(u,v) + 1/2$. If *both* u and v are in $\hat{V}_{\rm out}$ then both will have their y-values decremented and $yz(u,v)$ will be $w(u,v) - 1/2$ after Dual Adjustment, violating domination. This situation, however, cannot happen. The y-values of all vertices reachable from free vertices have the same parity (as a multiple of 1/2) so if $u,v \in \hat{V}_{\rm out}$ and (u,v) is not tight then $yz(u,v) \geq w(e) + 1$.

[4] The complications of blossom manipulation in other matching algorithms (e.g., [3, 17, 25, 31, 32]) arise from the need to interleave graph searching and dual adjustments and to detect augmenting paths. However, the Blossom Formation step in this algorithm makes no dual adjustments and, by definition, cannot find an augmenting path in the tight subgraph.

Theorem 3.1. $\text{MWM}(m,n,N) = O(N \cdot \text{MCM}(m,n))$ *and* $\text{MWM}_{\text{bi}}(m,n,N) = O(N \cdot \text{MCM}_{\text{bi}}(m,n))$*. According to the known bounds on* MCM_{bi} *and* MCM *[7, 9–11, 18], both* $\text{MWM}(m,n,N)$ *and* $\text{MWM}_{\text{bi}}(m,n,N)$ *are* $O(N \cdot m\sqrt{n}\frac{\log(n^2/m)}{\log n})$ *(determistically) and* $O(N \cdot n^{\omega})$ *(randomized, w.h.p.) and, furthermore,* $\text{MWM}_{bi}(m,n,N) = O(N \cdot (n^2 + n^{5/2}/w))$ *(deterministically).*

Theorem 3.1 is only superior to the recent $O(m\sqrt{n}\log N)$-time MWM_{bi} algorithm of [6] when the graph is very dense, but it is superior to the $O(m\sqrt{n\log n}\log(nN))$-time MWM algorithm of [17] both when the graph is dense and when $N = o(\log^{3/2} n)$.

Call a class $\mathcal{C}$ of graphs *good* if it is closed under taking subgraphs, and, if $\mathcal{C}$ does not consist solely of bipartite graphs, if it is also closed under taking minors. We have actually shown that $\text{MWM}_{\mathcal{C}}(m,n,N) = O(N \cdot \text{MCM}_{\mathcal{C}}(m,n))$, where $\text{MWM}_{\mathcal{C}}$ and $\text{MCM}_{\mathcal{C}}$ are the complexities of the problems on any good $\mathcal{C}$. This follows from the fact that $G \in \mathcal{C}$ implies $G_{\text{tight}}/\Omega \in \mathcal{C}$ as well. Our reduction, together with existing algorithms for good graph classes [26–28] gives a number of new bounds on the complexity of MWM. Mucha and Sankowski [26] showed that MCM can be solved in $O(n^{\omega/2})$ time in general planar graphs. Yuster and Zwick [27] generalized this algorithm to graphs of bounded genus, and gave a new $O(n^{3\omega/(3+\omega)})$-time MCM algorithm for H-minor free graphs. Very recently Borradaile, Klein, Mozes, Nussbaum, and Wulff-Nilsen [28] gave an $O(n\log^3 n)$-time MCM algorithm for bipartite planar graphs. At the time of their publication, all of the algorithms cited above improved on the standard $\tilde{O}(n^{3/2})$-time algorithms. Theorem 3.2 follows immediately.

Theorem 3.2. $\text{MWM}_{\text{planar}}(n,N), \text{MWM}_{\text{low-genus}}(n,N) = O(N \cdot n^{\omega/2})$, $\text{MWM}_{\text{minor-free}}(n,N) = O(N \cdot n^{3\omega/(3+\omega)})$, *and* $\text{MWM}_{\text{bi-planar}}(n,N) = O(N \cdot n\log^3 n)$.

4. Conclusion

We have given simple reductions from maximum weight matching to maximum cardinality matching, though the loss in efficiency is *linear* in the maximum edge weight N. Is it possible to improve the dependence on N while still using cardinality matching algorithms in a similar black-box fashion? Is there an efficient reduction from maximum weight *perfect* matching to maximum cardinality matching?

References

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

[2] C.-C. Huang, T. Kavitha, Efficient algorithms for maximum weight matchings in general graphs with small edge weights, in: Proceedings 23rd ACM-SIAM Symposium on Discrete Algorithms (SODA), pp. 1400–1412.

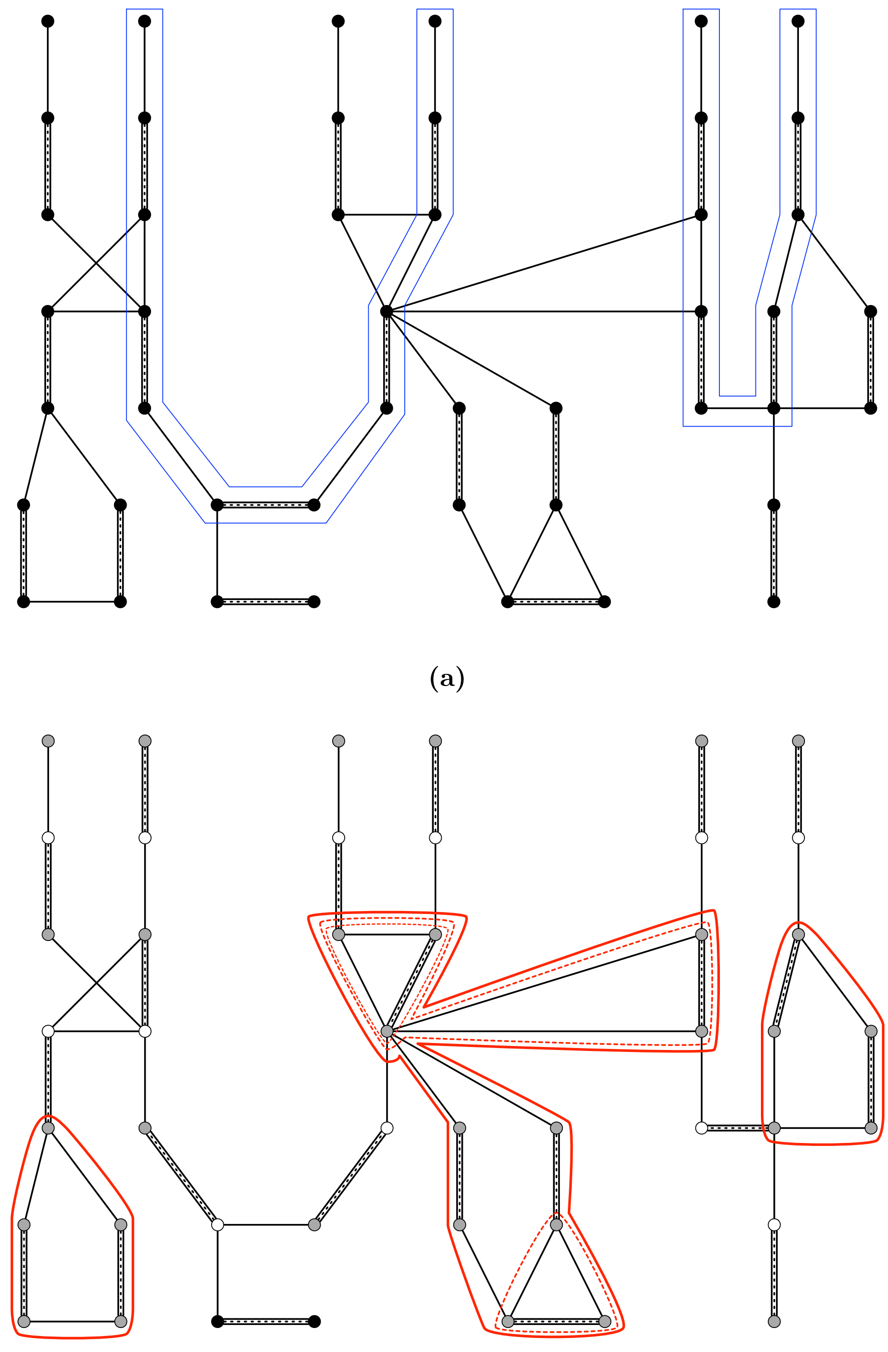
[3] H. N. Gabow, Scaling algorithms for network problems, J. Comput. Syst. Sci. 31 (1985) 148–168.

[4] H. N. Gabow, A scaling algorithm for weighted matching on general graphs, in: Proceedings 26th IEEE Symposium on Foundations of Computer Science (FOCS), pp. 90–100.

[5] R. Duan, H.-H. Su, A scaling algorithm for maximum weight matching in bipartite graphs, in: Proceedings 23rd ACM-SIAM Symposium on Discrete Algorithms (SODA), pp. 1413–1424.

[6] R. Duan, S. Pettie, H.-H. Su, Scaling algorithms for approximate and exact maximum weight matching, 2011. ArXiv:1112.0790.

[7] A. V. Goldberg, A. V. Karzanov, Maximum skew-symmetric flows and matchings, Math. Program., Ser. A 100 (2004) 537–568.



(a)

(b)

Figure 2: All depicted edges are in G_{tight}, where $\Omega = \emptyset$. Matched edges in M are thick and dashed, unmatched edges thin. (a) Two augmenting paths suffice to extend M to an MCM. (b) After Augmentation and Blossom Formation. Gray vertices are in V_{out}, white are in V_{in}, and black are in neither. Six blossoms are formed, three of which are root blossoms (marked with solid lines), whose z-values become 1 after Dual Adjustment.

[8] A. V. Goldberg, R. Kennedy, Global price updates help, SIAM J. Discrete Mathematics 10 (1997) 551–572.

[9] T. Feder, R. Motwani, Clique partitions, graph compression and speeding-up algorithms, J. Comput. Syst. Sci. 51 (1995) 261–272.

[10] M. Mucha, P. Sankowski, Maximum matchings via Gaussian elimination, in: Proceedings 45th IEEE Symposium on Foundations of Computer Science (FOCS), pp. 248–255.

[11] N. Harvey, Algebraic algorithms for matching and matroid problems, SIAM J. Comput. 39 (2009) 679–702.

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

[13] E. A. Dinic, Algorithm for solution of a problem of maximum flow in networks with power estimation, Soviet Math. Dokl. 11 (1970) 1277–1280.

[14] A. V. Karzanov, An exact estimate of an algorithm for finding a maximum flow, applied to the problem "on representatives" [in Russian], Problems in Cybernetics 5 (1973) 66–70. Announced at the Seminar on Combinatorial Mathematics (Moscow, 1971). English translation available at the author's website.

[15] S. Micali, V. V. Vazirani, An $O(\sqrt{|V|} \cdot |E|)$ algorithm for finding maximum matching in general graphs, in: Proceedings 21st IEEE Symposium on Foundations of Computer Science (FOCS), pp. 17–27.

[16] V. V. Vazirani, A theory of alternating paths and blossoms for proving correctness of the $O(\sqrt{V}E)$ general graph maximum matching algorithm, Combinatorica 14 (1994) 71–109.

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

[18] J. Cheriyan, K. Mehlhorn, Algorithms for dense graphs and networks on the random access computer, Algorithmica 15 (1996) 521–549.

[19] Y. Han, Deterministic sorting in $O(n \log \log n)$ time and linear space, in: Proceedings 34th ACM Symposium on Theory of Computing (STOC), pp. 602–608.

[20] M. Thorup, Integer priority queues with decrease key in constant time and the single source shortest paths problem, in: Proceedings 35th ACM Symposium on Theory of Computing (STOC), pp. 149–158.

[21] M. Thorup, Equivalence between priority queues and sorting, J. ACM 54 (2007).

[22] P. Sankowski, Maximum weight bipartite matching in matrix multiplication time, Theoretical Computer Science 410 (2009) 4480–4488.

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

[24] J. B. Orlin, R. K. Ahuja, New scaling algorithms for the assignment and minimum mean cycle problems, Math. Program. 54 (1992) 41–56.

[25] H. N. Gabow, Data structures for weighted matching and nearest common ancestors with linking, in: Proceedings First Annual ACM-SIAM Symposium on Discrete Algorithms (SODA), pp. 434–443.

[26] M. Mucha, P. Sankowski, Maximum matchings in planar graphs via Gaussian elimination, Algorithmica 45 (2006) 3–20.

[27] R. Yuster, U. Zwick, Maximum matching in graphs with an excluded minor, in: Proceedings 18th ACM-SIAM Symposium on Discrete Algorithms (SODA), pp. 108–117.

[28] G. Borradaile, P. N. Klein, S. Mozes, Y. Nussbaum, C. Wulff-Nilsen, Multiple-source multiple-sink maximum flow in directed planar graphs in near-linear time, in: Proceedings 52nd IEEE Symposium on Foundations of Computer Science (FOCS), pp. 170–179.

[29] J. Edmonds, Maximum matching and a polyhedron with $0,1$-vertices, J. Res. Nat. Bur. Standards Sect. B 69B (1965) 125–130.

[30] J. Edmonds, Paths, trees, and flowers, Canadian Journal of Mathematics 17 (1965) 449–467.

[31] Z. Galil, S. Micali, H. N. Gabow, An $O(EV \log V)$ algorithm for finding a maximal weighted matching in general graphs, SIAM J. Comput. 15 (1986) 120–130.

[32] H. N. Gabow, Z. Galil, T. H. Spencer, Efficient implementation of graph algorithms using contraction, J. ACM 36 (1989) 540–572.