Information Processing Letters 112 (2012) 893–898

Contents lists available at SciVerse ScienceDirect

Information Processing Letters

www.elsevier.com/locate/ipl

A simple reduction from maximum weight matching to maximum cardinality matching [☆]

S. Pettie [*]

University of Michigan, Dept. of Electrical Engineering and Computer Science, 2260 Hayward St., Ann Arbor, MI 48109, United States

ARTICLE INFO

Article history:
Received 31 December 2011
Received in revised form 15 March 2012
Accepted 20 August 2012
Available online 29 August 2012
Communicated by Tsan-sheng Hsu

Keywords:
Graph algorithms
Maximum matching

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 (2001) [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 (2012) [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 [3] and 1985 [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.

© 2012 Elsevier B.V. All rights reserved.

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.

* Tel.: +1 (734) 972 0375; fax: +1 (734) 763 1260.
E-mail address: pettie@umich.edu.

[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 $m = n^{2-o(1)}$.

0020-0190/$ – see front matter © 2012 Elsevier B.V. All rights reserved.
http://dx.doi.org/10.1016/j.ipl.2012.08.010

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

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

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

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

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 MWPM_{bi} algorithm for bipartite graphs could be modified to solve MWM_{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 MWM_{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 MWM/MWM_{bi} to N executions of MCM/MCM_{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 works 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] Kao et al. do not cite Gabow's $O(Nm\sqrt{n})$-time algorithm. They compare their time bound of $N \cdot \text{MCM}_{\text{bi}}(m,n)$ against Gabow and Tarjan's bound of $O(m\sqrt{n}\log(nN))$.

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

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

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.

$$\begin{aligned} \text{minimize} \quad & \sum_{u \in V(G)} y(u) + \sum_{B \in \mathcal{V}_{\text{odd}}} z(B)(|B|-1)/2 & \\ \text{subject to} \quad & yz(e) \geqslant w(e) & \forall e \in E(G) \\ & y(u) \geqslant 0 & \forall u \in V(G) \\ & z(B) \geqslant 0 & \forall B \in \mathcal{V}_{\text{odd}} \end{aligned} \tag{2}$$

where, by definition,

$$yz(u,v) \stackrel{\text{def}}{=} y(u) + y(v) + \sum_{B \in \mathcal{V}_{\text{odd}}:\, u,v \in B} z(B)$$

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 $B = \bigcup_{0 \leqslant i < \ell} A_i$ is a vertex set containing the blossom $E_B = \bigcup_{0 \leqslant i < \ell} E_{A_i} \cup \{e_0, \ldots, e_{\ell-1}\}$. The unique unmatched vertex in E_B is called the *base* of B. See Fig. 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

Download English Version:

https://daneshyari.com/en/article/427670

Download Persian Version:

https://daneshyari.com/article/427670

Daneshyari.com