Bisimplicial Edges in Bipartite Graphs ⋆

Matthijs Bomhoff * Bodo Manthey

Faculty of Electrical Engineering, Mathematics and Computer Science
University of Twente, P.O. Box 217, 7500 AE Enschede, The Netherlands

Abstract

Bisimplicial edges in bipartite graphs are closely related to pivots in Gaussian elimination that avoid turning zeroes into non-zeroes. We present a new deterministic algorithm to find such edges in bipartite graphs. The expected time complexity of our new algorithm is $O\left(n^2 \log n\right)$ on random bipartite graphs in which each edge is present with a fixed probability p, a polynomial improvement over the fastest algorithm found in the existing literature.

Key words: bipartite graphs, random graphs, algorithms, Gaussian elimination

1 Introduction

When applying Gaussian elimination to a square $n \times n$ matrix M containing some elements with value zero, the choice of pivots can often determine the amount of zeroes turned into non-zeroes during the process, the so called *fill-in*. Some matrices even allow Gaussian elimination without any fill-in. Avoiding fill-in has the nice property of bounding the required space for intermediate results of the Gaussian elimination to the space required for storing the input matrix M. This can be important for processing very large sparse matrices. Even when fill-in cannot be completely avoided, it may still be worthwhile to avoid it for several iterations, motivating the search for pivots that avoid fill-in.

⋆ This work was supported by the Dutch Innovation Oriented Research Program "Integral Product Creation and Realisation (IOP-IPCR)" of the Dutch Ministry of Economic Affairs.

* Corresponding Author
Email addresses: m.j.bomhoff@utwente.nl (Matthijs Bomhoff), b.manthey@utwente.nl (Bodo Manthey).

If we assume subtracting a multiple of one row of M from another turns at most one non-zero into a zero, we may restrict ourselves to considering only $\{0,1\}$ matrices. Given such a square matrix M, we can construct the bipartite graph $G[M]$ with vertices corresponding to the rows and columns in M, where vertices i and j are adjacent if and only if $M_{i,j}$ is nonzero. The $\{0,1\}$ matrices that allow Gaussian elimination without fill-in correspond to the class of *perfect elimination bipartite* graphs [1]. Central to the recognition of this class of graphs is the notion of a *bisimplicial* edge that corresponds to an element of M that can be used as a pivot without causing fill-in. The fastest algorithm for finding bisimplicial edges in the existing literature has a time complexity equal to that of matrix multiplication [2,3], i.e., $O\left(n^{2.376}\right)$ [4]. We present a new deterministic algorithm for finding a bisimplicial edge in a bipartite graph, if one exists, and show that its expected time complexity on random bipartite graphs where each edge is present with some fixed probability p is $O\left(n^2 \log n\right)$.

2 Bisimplicial Edges

We denote by $\Gamma(u)$ the neighbors of a vertex u.

Definition 1 *An edge uv of a bipartite graph $G=(U,V,E)$ is called* bisimplicial*, if the induced subgraph $G[\Gamma(u)\cup\Gamma(v)]$ is a complete bipartite graph.*

Clearly, for a given edge uv we can determine in $O(|E|)$ time if it is a bisimplicial edge by simply checking all edges adjacent to it. So a simple algorithm to find a bisimplicial edge in a bipartite graph G, if one exists, takes $O\left(|E|^2\right)$ time.

Goh and Rotem [2] present a faster algorithm based on the following: A row $M_{a,*}$ is said to *majorize* a row $M_{b,*}$ if for each $1 \le j \le n$ we have $M_{a,j} \ge M_{b,j}$. According to this definition, every row majorizes itself.

Theorem 2 (Goh and Rotem [2]) *Let M be an $n\times n$ $\{0,1\}$ matrix representing a bipartite graph $G=(U,V,E)$. Let ℓ_i be the number of rows in M that majorize row i and let s_j be the sum of the entries in column j of M. Then $M_{i,j}=1$ and $\ell_i = s_j$ if and only if the edge u_iv_j is a bisimplicial edge of G.*

Let $Q = MM^T$, this implies ℓ_i is equal to the number of elements in the row $Q_{i,*}$ that are equal to $Q_{i,i}$ (including $Q_{i,i}$ itself). Clearly, the time complexity of finding a bisimplicial edge in $G[M]$ is bounded by the time complexity of the matrix multiplication. The fastest currently known algorithm for this has a time complexity of $O\left(n^{2.376}\right)$ [4].

To improve on this, our new approach first selects a set of candidate edges. If a bisimplicial edge exists, then one of our candidates is bisimplicial. Thus we can restrict ourselves to checking the candidate edges for bisimpliciality. By bounding the number of candidates we achieve an improved expected time complexity. The following observation is the basis of our candidate selection procedure.

Lemma 3 *If an edge uv of a bipartite graph $G = (U, V, E)$ is bisimplicial, we must have $\delta(u) = \min_{u' \in \Gamma(v)} \delta(u')$ and $\delta(v) = \min_{v' \in \Gamma(u)} \delta(v')$.*

Translated to the matrix M, this means that if $M_{i,j} = 1$, it can only correspond to a bisimplicial edge if row i has a minimal number of ones over all the rows that have a 1 in column j and column j has a minimal number of ones over all the columns having a 1 in row i. In what follows, we will call the row (column) in M with the minimal number of ones over all the rows (columns) in M the *smallest* row (column). Using this observation, we construct an algorithm to pick candidate edges that may be bisimplicial:

Algorithm 1 *(1) Determine the row and column sums (a_i and b_j) for each row and column of M.*

(2) Determine for each row i the index c_i of the smallest column with $M_{i,c_i} = 1$ (breaking ties by favoring the lowest index); or $c_i = 0$ if row i has no one.

(3) Determine for each column j the index r_j of the smallest row with $M_{r_j,j} = 1$ (breaking ties by favoring the lowest index); or $r_j = 0$ if column j has no one.

(4) Mark $M_{i,j}$ as a candidate edge if $c_i = j$ and $r_j = i$.

Clearly, all steps in the algorithm can be performed in $O(n^2)$ time. Furthermore, the last step will mark at most n candidate edges (and at least 1).

Theorem 4 *If $G = (U, V, E)$ contains a bisimplicial edge, at least one of the candidates marked by the algorithm will be bisimplicial.*

As each of the candidates can subsequently be checked for bisimpliciality in $O(n^2)$, we obtain a $O(n^3)$ algorithm that finds a single bisimplicial edge in a bipartite graph G, if one exists, without using matrix multiplication. By itself, this is not really interesting, as the worst case time complexity is not an improvement over previously known algorithms. However, for random bipartite graphs, our new algorithm performs significantly better.

3 Asymptotic Expected Behavior on Random Graphs

For a fixed value of $p \in (0,1)$, we consider random bipartite graphs from the $G_{n,n,p}$ model: i.e., we have n vertices in each vertex class, and each edge is present with probability p. Such a random graph corresponds to a stochastic $n \times n$ $\{0,1\}$ matrix M with $\mathbb{P}[M_{i,j} = 1] = p$. We denote by random variable X_i the $\{0,1\}$ vector that forms row i of M and use $|X_i|$ to denote the sum of its elements. If we order the X_i vectors according to the number of ones they contain (breaking ties by favoring lower values of i), we denote by $X_{(1)}$ the row with the least number of ones, by $X_{(2)}$ the row with the second-to-least number etc.

Lemma 5 *For any $\varepsilon > 0$ and sufficiently large n, we have*

$$\mathbb{P}\left[|X_{(1)}| < (1-\varepsilon)pn\right] \leq \frac{1}{n}.$$

Lemma 6 *For any p', k with $0 < p' < p$ and $k \geq 1$ and sufficiently large n, we have*

$$\mathbb{P}\left[\textit{Algorithm 1 selects more than } k \textit{ candidates}\right] \leq n(1-p')^k + \frac{1}{n}.$$

From this, we immediately get a bound on the expected number of candidates that our algorithm selects.

Theorem 7 *For any p' with $0 < p' < p$ and sufficiently large n, we have*

$$\mathbb{E}\left[\textit{number of candidates selected by Algorithm 1}\right] \leq 2 + 2\log_{(1-p')}\frac{1}{n}.$$

Corollary 8 *For any fixed p, the expected time complexity of finding a bisimplicial edge or deciding there is none using Algorithm 1 is $O\left(n^2 \log n\right)$.*

References

[1] Martin Charles Golumbic and Clinton F. Goss. Perfect elimination and chordal bipartite graphs. *J. Graph Theory*, 2(2):155–163, 1978.

[2] L. Goh and D. Rotem. Recognition of perfect elimination bipartite graphs. *Inform. Process. Lett.*, 15(4):179–182, 1982.

[3] Jeremy P. Spinrad. Recognizing quasi-triangulated graphs. *Discrete Appl. Math.*, 138(1-2):203–213, 2004.

[4] Don Coppersmith and Shmuel Winograd. Matrix multiplication via arithmetic progressions. *J. Symbolic Comput.*, 9(3):251–280, 1990.