Available online at www.sciencedirect.com
SCIENCE DIRECT®
Theoretical Computer Science 337 (2005) 240–248
Theoretical Computer Science
www.elsevier.com/locate/tcs

Generating bicliques of a graph in lexicographic order

Vânia M.F. Dias[a], Celina M.H. de Figueiredo[b,*], Jayme L. Szwarcfiter[c]

[a] *Universidade Federal do Rio de Janeiro, COPPE, Caixa Postal 68530, 21945-970 Rio de Janeiro, Brazil*
[b] *Universidade Federal do Rio de Janeiro, IM and COPPE, Caixa Postal 68530, 21945-970 Rio de Janeiro, Brazil*
[c] *Universidade Federal do Rio de Janeiro, IM, COPPE, and NCE, Caixa Postal 68511, 21945-970 Rio de Janeiro, Brazil*

Received 20 April 2004; received in revised form 25 November 2004; accepted 4 January 2005

Communicated by G. Ausiello

Abstract

An independent set of a graph is a subset of pairwise non-adjacent vertices. A complete bipartite set B is a subset of vertices admitting a bipartition $B = X \cup Y$, such that both X and Y are independent sets, and all vertices of X are adjacent to those of Y. If both $X, Y \neq \emptyset$, then B is called proper. A biclique is a maximal proper complete bipartite set of a graph. We present an algorithm that generates all bicliques of a graph in lexicographic order, with polynomial-time delay between the output of two successive bicliques. We also show that there is no polynomial-time delay algorithm for generating all bicliques in reverse lexicographic order, unless P = NP. The methods are based on those by Johnson, Papadimitriou and Yannakakis, in the solution of these two problems for independent sets, instead of bicliques.
© 2005 Elsevier B.V. All rights reserved.

Keywords: Algorithms; Biclique; Enumeration; NP-complete; Lexicographic order; Polynomial-time delay

* Corresponding author. Tel.: +55 21 2598 3393; fax: +55 21 2290 6626.
E-mail addresses: vaniad@cos.ufrj.br (V.M.F. Dias), celina@cos.ufrj.br (C.M.H. de Figueiredo), jayme@nce.ufrj.br (J.L. Szwarcfiter).

0304-3975/$ - see front matter © 2005 Elsevier B.V. All rights reserved.
doi:10.1016/j.tcs.2005.01.014

1. Introduction

Generating all configurations that satisfy a given specification is a well-studied problem in combinatorics and in graph theory suggesting many interesting problems. Among them, generating all maximal independent sets of a given graph is one that has attracted considerable attention [8,10,13,17].

We use the following notation. Let $G = (V, E)$ be a graph, and $V = \{1, \ldots, n\}$. Denote by N_i the set of neighbours of vertex i, and $\overline{N}_i = (V \setminus N_i) \setminus \{i\}$. Let $X \subseteq V$. The *focus* of X is the subset $F(X) = \cap_{i \in X} N_i$. When each vertex of X is adjacent (non-adjacent) to every other vertex of the set then call it a *complete* (*independent*) set. Let $X, Y \subseteq V$. Say that $B = X \cup Y$ is a *complete bipartite* set when both X and Y are independent sets and every vertex of X is adjacent to every vertex of Y, i.e. B induces a complete bipartite subgraph in G. If $X, Y \neq \emptyset$, then B is called *proper*, i.e. B induces a complete bipartite subgraph containing at least one edge of G, otherwise *degenerate*. An independent (complete, complete bipartite) set is *maximal*, when it is not properly contained in any such set. A maximal proper complete bipartite set of G is called a *biclique* of G. In the definition of biclique, when the requirement that X and Y are independent sets of G is dropped, we have a *non-induced biclique*. On the other hand, bicliques as above defined are the *induced bicliques*.

Bicliques have been studied in some different contexts: Applications in automata and language theories, graph compression, partial orders, artificial intelligence, and biology are discussed in [2]; applications to enumeration of cyber-communities are discussed in [9]; Tuza [18] proves bounds on the number of bicliques needed to cover the edges of a general graph; Orlin [12] proves for general bipartite graphs and Müller [11] for chordal bipartite graphs that computing the minimum cardinality of a biclique cover is NP-hard; Prisner [15] shows that bicliques are the key structure in certain classes of graphs, and how generating all bicliques of such graphs seems an approach for recognition algorithms; Prisner [16] gives upper bounds on the number of bicliques in bipartite graphs and general graphs, exhibits examples of classes of graphs where the number of bicliques is exponential, and characterizes classes of graphs where the number of bicliques is polynomial in the number of vertices of the graph. The complexity of deciding whether a graph contains a biclique of a certain size is first mentioned in [5] with the NP-completeness of the balanced complete bipartite subgraph problem, whereas Dawande et al. [4] show the maximum biclique problem is polynomial for bipartite graphs, and Yannakakis [19] shows the problem is NP-complete for general graphs; the NP-completeness of the weighted maximum edge biclique problem for bipartite graphs is established by Dawande et al. [4], and more recently for the non-weighted version by Peeters [14]; upper bounds for the maximum number of edges in a biclique, and approximation algorithms for the edge or node deletion biclique problems are presented by Hochbaum [7]. Bicliques have also been employed in the study of absolute bipartite retracts [3], and for characterizing chordal bipartite graphs [6]. Alexe et al. [1] describe an algorithm for generating all non-induced bicliques. On the other hand, there is no algorithm for generating all induced bicliques of a general graph. In the present paper, we propose such an algorithm.

Alexe et al. [1] observed that any algorithm for generating all maximal independent sets can be used for generating all non-induced bicliques of a graph $G = (V, E)$ as follows. Let G' be a graph consisting of two disjoint complete sets V_1 and V_2, each on $|V|$ vertices,

where each $v_i \in V$ is labelled as $v'_i \in V_1$, and as $v''_i \in V_2$, and where v'_i and v''_j are adjacent in G' if and only if v_i and v_j are so in G. There is a 2–1 correspondence between the cliques (i.e. the maximal complete sets) of G' and the non-induced bicliques of G, except for the two cliques V_1 and V_2. Consequently, non-induced bicliques can be generated in polynomial-time delay.

A similar approach can be used for generating all maximal complete bipartite sets, proper and degenerate, of a graph G. Let G' be the graph obtained from G as above, except that V_1 and V_2, instead of being complete sets, form complements of the graph G. There is a 2–1 correspondence between the cliques of G' and maximal complete bipartite sets of G, implying that the latter sets can be generated by a polynomial-time delay algorithm, with polynomial space.

Clearly, the above method can be used for generating all maximal complete bipartite sets which are proper, i.e. the bicliques of the graph G. It is simple to distinguish between the proper and the degenerate maximal complete bipartite sets. Therefore the algorithm would generate all cliques of G', ignoring those corresponding to maximal independent sets of G, not contained in bicliques. However, the number of such maximal independent sets of G might be exponential in the number of its bicliques. For example, when G consists of n isolated edges then G has n bicliques and 2^n maximal independent sets, implying that the algorithm would not obey polynomial-time delay.

Given two subsets S and T of an ordered set, say that S is *lexicographically smaller* than T, if one of the two following conditions occurs: (i) the least at which S and T of disagree is in S, or (ii) the least $|S|$ elements of T coincide with those of S and $|S| < |T|$. Represent by B^* the least lexicographic biclique of G. Johnson et al. [8] showed that there is no polynomial-delay algorithm for generating all maximal independent sets of a given graph in reverse lexicographic order, unless P = NP. Nevertheless, they presented an algorithm that generates all maximal independent sets of a graph in lexicographic order, with only polynomial delay between the output of two successive independent sets.

The goal of the present paper is to extend the results of [8] to bicliques. We show in Section 2 that there is no polynomial-delay algorithm for generating all bicliques in reverse lexicographic order, unless P = NP. In Section 3 we present an $O(|V|^2)$ algorithm that generates the lexicographically least biclique containing a given complete bipartite set. In particular, this algorithm can generate the least biclique of G. The algorithm is used in Section 4 as part of an algorithm that generates all bicliques of a graph in lexicographic order, with only polynomial-time delay between the output of two successive bicliques.

2. NP-completeness of the lexicographically largest biclique

In this section we establish the NP-completeness of deciding whether a given biclique is the lexicographically largest biclique.

SATISFIABILITY

Instance: Set $X = \{x_1, \ldots, x_k\}$ of k Boolean variables, collection $C = \{c_1, \ldots, c_n\}$ of $n > 1$ clauses over literals of X.

Question: Is there a truth assignment satisfying C?

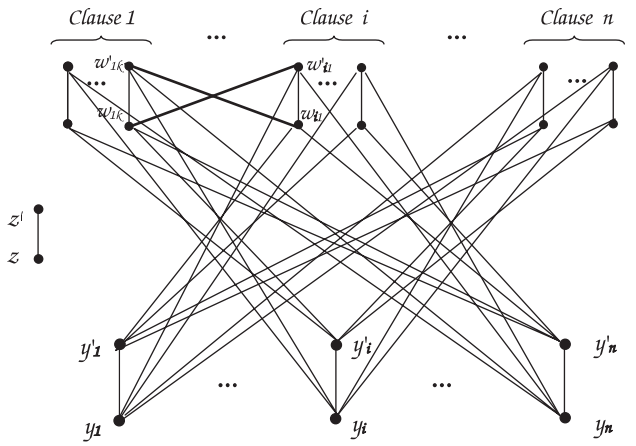

Fig. 1. Constructed graph G. Literals ℓ_{1k} and ℓ_{i1} are not contradicting literals.

LEXICOGRAPHICALLY LARGEST BICLIQUE
Instance: Graph $G = (V, E)$, biclique B, order on V.
Question: Is there a biclique B' of G such that B' is lexicographically larger than B?

Theorem 1. *Given a graph $G = (V, E)$, a biclique B, and an order on V, it is* coNP-*complete to decide whether B is the lexicographically largest biclique.*

Proof. Problem LEXICOGRAPHICALLY LARGEST BICLIQUE is in coNP since a short certificate is a biclique B' that is lexicographically larger than B.

To show completeness, we give a polynomial transformation from SATISFIABILITY. Given an instance (X, C) of SATISFIABILITY we construct in polynomial time a graph $G = (V, E)$, a biclique B, and an order on V, such that there is a truth assignment satisfying C if and only if B is not the lexicographically largest biclique.

Let ℓ_{ij} be the jth literal of clause c_i. Vertex set V is the union $V = Y \cup Z \cup W$, where $Y = \{y_1, \ldots, y_n, y'_1, \ldots, y'_n\}$, set Y contains a pair of vertices y_i, y'_i corresponding to each clause c_i; $Z = \{z, z'\}$; set W contains a pair of vertices w_{ij}, w'_{ij} corresponding to each ℓ_{ij}. Edge set E is the union $E = Y^* \cup Z^* \cup W^*$, where $Y^* = \{(y_i, y'_i) : i = 1, \ldots, n\} \cup \{(y_i, w'_{pq}) : i = 1, \ldots, n \text{ and } p \neq i\} \cup \{(y'_i, w_{pq}) : i = 1, \ldots, n \text{ and } p \neq i\}$; $Z^* = \{(z, z')\}$; $W^* = \{(w_{ij}, w'_{ij}) : w_{ij}, w'_{ij} \in W\} \cup \{(w_{ij}, w'_{pq}), (w'_{ij}, w_{pq}) : i \neq p \text{ and } \ell_{ij} \neq \neg\ell_{pq}\}$.

Define the biclique $B = \{z'\} \cup \{z\}$.

Finally, the order on V is defined as follows. Each vertex y'_i has label i, $i = 1, \ldots, n$, each vertex y_i has label $n + i$, $i = 1, \ldots, n$; vertices z and z' have labels $2n + 1$ and $2n + 2$, respectively; each vertex $w \in W$ has a label $j \geqslant 2n + 3$. The construction is completed (see Fig. 1).

The structure of the graph G gives the following equivalence. There is a biclique of G larger than B if and only if there is a truth assignment satisfying C.

Suppose G contains a biclique $B' = S \cup R$ lexicographically larger than B. Hence $B' \subseteq W$. Since B' is a proper complete bipartite set, we have $S, R \neq \emptyset$, i.e. B' contains

adjacent vertices $w_{kp} \in S$ and $w'_{jq} \in R$. Since $y_k \notin B'$, the biclique B' is maximal, and $(y_k, w_{kp}) \notin E$, we have that B' contains $w'_{kr} \in R$. Now $w_{kp}, w'_{kr} \in B'$, with $w_{kp} \in S$ and $w'_{kr} \in R$, imply $r = p$, i.e. adjacent vertices $w_{kp}, w'_{kp} \in B'$. Hence, for each clause c_i, $i = 1, \ldots, n$, the pair $y_i, y'_i \notin B'$, implies a pair of adjacent vertices $w_{it}, w'_{it} \in B'$, with $w_{it} \in S$ and $w'_{it} \in R$. Now vertices corresponding to contradicting literals of distinct clauses are non-adjacent in G. So B' consists of n pairs w_{ip_i}, w'_{ip_i} of adjacent vertices, for $i = 1, \ldots, n$, of W, and these n pairs correspond to n literals (not necessarily distinct), one in each of the n clauses of C, which may assume value true simultaneously. This defines a satisfying truth assignment. Conversely, any satisfying truth assignment corresponds to a biclique of G entirely contained in W, meaning that it is lexicographically larger than B. □

Corollary 2. *Given a graph $G = (V, E)$, a biclique B, and an order on V, it is* coNP-*complete to test if B is the lexicographically largest non-induced biclique.*

Proof. The graph constructed in the proof of Theorem 1 is a bipartite graph. In a bipartite graph, every non-induced biclique is actually a (induced) biclique. □

3. Greedy generation of the lexicographically least biclique

Let $G = (V, E)$ be a graph, with an order on $V = \{1, \ldots, n\}$. We describe an $O(n^2)$ greedy algorithm to generate the lexicographically least biclique containing a given complete bipartite set. Given a complete bipartite set $B = X \cup Y$, with $X \neq \emptyset$, the algorithm finds the least biclique B' of G containing B, or answers $B' = \emptyset$, whenever B is contained in no biclique of G.

Algorithm: Least biclique
Input: Graph $G = (V, E)$, order on V, complete bipartite set $B = X \cup Y$
Output: Least biclique B' containing B, if it exists, and $B' = \emptyset$, if it does not exist
1. If $Y = \emptyset$ then
    if $F(X) = \emptyset$, then answer $B' = \emptyset$
    else $i \leftarrow 1$
        repeat $j \leftarrow i$-th least vertex of $V \setminus B$
            if $\overline{N}_j \supseteq X$ and $F(X) \cap N_j \neq \emptyset$ then $X \leftarrow X \cup \{j\}$
            if $N_j \supseteq X$ then $Y \leftarrow \{j\}$
            $i \leftarrow i + 1$
        until $Y \neq \emptyset$
2. $X' \leftarrow X$; $Y' \leftarrow Y$
    for each $k \in V \setminus (X \cup Y)$, $k > 1$, in increasing order do
        if $\overline{N}_k \supseteq X'$ and $N_k \supseteq Y'$ then $X' \leftarrow X' \cup \{k\}$
        if $\overline{N}_k \supseteq Y'$ and $N_k \supseteq X'$ then $Y' \leftarrow Y' \cup \{k\}$
    answer $B' = X' \cup Y'$

Given a graph $G = (V, E)$, with an order on V, and given a complete bipartite set $B = X \cup Y$, Step 1 of Algorithm Least biclique finds the least proper complete bipartite set containing B, if any, while Step 2 extends it to a maximal one.

To find B^* (the least biclique of G), apply Algorithm Least biclique with $B = X \cup Y$, $X = \{k\}$, where k is the least non-isolated vertex of G, and $Y = \emptyset$.

The complexity of Algorithm Least biclique set is $O(n^2)$.

4. Generation in lexicographic order with polynomial-time delay

Let $G = (V, E)$ be a graph, with an order on $V = \{1, \ldots, n\}$. Write $B_j = B \cap \{1, \ldots, j\}$, $X_j = X \cap \{1, \ldots, j\}$, and $Y_j = Y \cap \{1, \ldots, j\}$. The following algorithm lists all bicliques of G in lexicographic order. Lemma 3 characterizes the lexicographically least biclique by a maximality condition, and Theorem 4 shows how to use this condition to ensure that all bicliques are correctly output in lexicographic order by the algorithm.

Algorithm: Lex Biclique generation
Input: Graph $G = (V, E)$, order on V
Output: List of all bicliques of G in lexicographic order
Find the least biclique B^* of G
$Q \leftarrow \emptyset$
insert B^* in the queue Q
while $Q \neq \emptyset$ do
    find the least biclique $B = X \cup Y$ of Q
    remove B from Q and output it
    for each vertex $j \in V \setminus B$ do
        $X_j \leftarrow X \cap \{1, \ldots, j\}$; $Y_j \leftarrow Y \cap \{1, \ldots, j\}$
        repeat twice
            if $X_j \cap N_j \neq \emptyset$ or $Y_j \cap \overline{N}_j \neq \emptyset$ then
                $X'_j \leftarrow (X_j \setminus N_j) \cup \{j\}$; $Y'_j \leftarrow Y_j \setminus \overline{N}_j$
                if there exists no $\ell \in \{1, \ldots, j\} \setminus B_j$ such that $X'_j \cup Y'_j \cup \{\ell\}$
                extends to a biclique of G then
                    find the least biclique B' of G containing $X'_j \cup Y'_j$, if any
                    if $B' \neq \emptyset$ and $B' \notin Q$ then include B' in Q
            swap the contents of X_j and Y_j

Lemma 3. *Let $B = X \cup Y$ be a biclique of G. Then $B = B^*$, if and only if no biclique of G contains $B_j \cup \{\ell\}$, for any $j \in \{1, \ldots, n\}$ and $\ell \in \{1, \ldots, j\} \setminus B_j$.*

Proof. Let $B = X \cup Y$ and assume $B = B^*$. By contradiction, suppose that G contains a biclique $B' \supseteq B_j \cup \{\ell\}$, $\ell \in \{1, \ldots, j\} \setminus B_j$. Then B and B' first disagree on some vertex $\ell' \in B'$, such that $\ell' \leqslant \ell \leqslant j$. Consequently, $B' < B$, which is impossible. Conversely, by hypothesis no biclique of G contains $B_j \cup \{\ell\}$, for any $j \in \{1, \ldots, n\}$ and $\ell \in \{1, \ldots, j\}$. Again by contradiction, suppose that $B \neq B^*$. Let j be the smallest index such that $B_j \neq B^*_j$.

Then $j \in B^*$ and $j \notin B$. Consequently, B^* is a biclique containing $B_j \cup \{j\}$. The latter is impossible, which completes the proof. □

Theorem 4. *Given a graph G, the algorithm generates all bicliques of G in lexicographic order.*

Proof. In the algorithm, first examine one of the main iterations of the while loop, where the biclique $B = X \cup Y$ has been output. Assume that biclique B' is included in Q, in this step. Then B' is the least biclique of G containing $X'_j \cup Y'_j$, where $X'_j = (X_j \setminus N_j) \cup \{j\}$ and $Y'_j = Y_j \setminus \overline{N}_j$, for some $j \in V \setminus B$, satisfying $X_j \cap N_j \neq \emptyset$ or $Y_j \cap \overline{N}_j \neq \emptyset$. Since there is no $\ell \in \{1, \ldots, j\} \setminus B_j$ such that $X'_j \cup Y'_j \cup \{\ell\}$ extends to a biclique of G, we conclude that $B'_j = X'_j \cup Y'_j$. Because $X_j \cap N_j \neq \emptyset$ or $Y_j \cap \overline{N}_j \neq \emptyset$, the bicliques B and B' first disagree lexicographically in some vertex $i < j$, such that $i \in B \setminus B'$. Consequently B' is larger than B, in lexicographic order. Besides, the first biclique to be output is chosen as the least of Q. Therefore the bicliques are output in strictly increasing lexicographic order. That is, no biclique is output twice.

It remains to prove that all bicliques are output. The proof is by induction on the length of the output sequence. The first biclique output by the algorithm is B^*. In general, let $B = X \cup Y$, with $B \neq B^*$ be the next biclique in the lexicographic order of the bicliques of G. We show that $B \in Q$ at this occasion, implying that B would be chosen by the algorithm as the next one to be output, which would complete the proof. The idea is to identify a biclique $\tilde{B}$, such that in the iteration $\tilde{B}$ is output the algorithm includes B in Q, if so far $B \notin Q$.

Denote by j the maximal integer such that $B_j \cup \{\ell\}$ extends to some biclique of G, for some $\ell \in \{1, \ldots, j\} \setminus B_j$. Because of Lemma 3, there exists such $j > 0$. Since B is itself a biclique, $j < n$. Denote by $\tilde{B}$ a biclique of G containing $B_j \cup \{\ell\}$.

Then $\tilde{B}_j \supseteq B_j \cup \{\ell\}$. Write $\tilde{B}_j = \tilde{X}_j \cup \tilde{Y}_j$, with $\tilde{X}_j \supseteq X_j$ and $\tilde{Y}_j \supseteq Y_j$. Moreover, $\tilde{X}_j \neq X_j$ or $\tilde{Y}_j \neq Y_j$, because $\ell \in \tilde{B}_j \setminus B_j$. By the maximality of j, we know that $j+1 \in B$. Without loss of generality, assume $j + 1 \in X$. Again by the maximality of j and knowing that $j + 1 \in B$, it follows $N_{j+1} \supseteq \tilde{X}_j \setminus X_j$ and $\overline{N}_{j+1} \supseteq \tilde{Y}_j \setminus Y_j$. Since $\tilde{X}_j \neq X_j$ or $\tilde{Y}_j \neq Y_j$, we have that B and $\tilde{B}$ first disagree lexicographically in some vertex $i \in \tilde{B} \setminus B$. That is, $\tilde{B}$ is smaller than B. By induction, $\tilde{B}$ has already been output by the algorithm. In the sequel, we show that in the iteration in which $\tilde{B}$ was output, the algorithm includes in Q a biclique $\tilde{B}'$ (provided $\tilde{B}'$ was not already in Q), satisfying $\tilde{B}' = B$.

In the iteration in which $\tilde{B}$ has been output, examine the vertex $j + 1$. Clearly $j + 1 \in V \setminus \tilde{B}$, because $N_{j+1} \supseteq \tilde{X}_j \setminus X_j$ and $\overline{N}_{j+1} \supseteq \tilde{Y}_j \setminus Y_j$, while $\tilde{X}_j \neq X_j$ or $\tilde{Y}_j \neq Y_j$. The latter implies that there exists a step corresponding to $j + 1$. For the same reason, we know that it is impossible simultaneously to occur $\tilde{X}_j \cap N_j = \tilde{Y}_j \cap \overline{N}_j = \emptyset$ and $\tilde{Y}_j \cap N_j = \tilde{X}_j \cap \overline{N}_j = \emptyset$. Without loss of generality we can assume that $\tilde{X}_j \cap N_j \neq \emptyset$ or $\tilde{Y}_j \cap \overline{N}_j \neq \emptyset$. The latter implies that at most one of the iterations of the block repeat twice of the algorithm is executed for $j + 1$. Let $\tilde{X}'_{j+1} = (\tilde{X}_{j+1} \setminus N_{j+1}) \cup \{j + 1\}$ and $\tilde{Y}'_{j+1} = \tilde{Y}_{j+1} \setminus \overline{N}_{j+1}$. Clearly, $\tilde{X}'_{j+1} = X_j \cup \{j+1\}$ and $\tilde{Y}'_{j+1} = Y_j$. By the maximality of j, there is no $\ell \in \{1, \ldots, j+1\} \setminus B_{j+1}$, such that $\tilde{X}_{j+1} \cup \tilde{Y}_{j+1} \cup \{\ell\}$ extends to a biclique of G.

Hence the algorithm computes the least biclique $\tilde{B}'$ of G containing $\tilde{X}'_{j+1} \cup \tilde{Y}'_{j+1} = B_{j+1}$. Since $B_{j+1} \subseteq B$, we know that $\tilde{B}' \neq \emptyset$. The algorithm includes $\tilde{B}'$ in Q, if so far $\tilde{B}' \notin Q$.

To complete the proof, we show that $\tilde{B}' = B$. We already know that $\tilde{B}_{j+1} = B_{j+1}$. By contradiction, suppose that $\tilde{B}' \neq B$. Because $\tilde{B}'$ is the least biclique containing B_{j+1}, $\tilde{B}'$ must be lexicographically smaller than B. Consequently, $\tilde{B}'$ and B first disagree in some vertex $k > j$, such that $k \in \tilde{B}'$ and $k \notin B$. The latter implies that $B_k \cup \{k\}$ extends to some biclique of G, which contradicts the maximality of j. Hence $\tilde{B}' = B$, as required. □

In order to evaluate the complexity of the algorithm, observe the size of Q is that of the number of bicliques of G, that is $O(2^n)$. Consequently, the operations of verifying if a biclique $B \in Q$, inserting B in Q and removing B from Q could be done in $O(n)$ time, implementing Q as a priority queue. When a biclique B is output, there could be $O(n)$ attempts to generate bicliques B', using Algorithm Least biclique of Section 3. Consequently, the algorithm requires $O(n)^3$ time between the outputs of two consecutive bicliques of G. The same bound applies for the time needed before the output of the least biclique and after the largest one. Consequently, the bicliques are generated in delay time $O(n^3)$.

The space required by the algorithm is the size of Q, that is $O(2^n)$. Similarly as for independent sets, it remains open whether the bicliques can be generated in lexicographic order, in polynomial-time delay, and polynomial space. However, for bicliques the question is more basic, to describe an algorithm for generating them all, in any ordering, obeying polynomial-time delay, and polynomial space.

Acknowledgements

We are grateful to the referee for his careful reading, suggestions and corrections, which helped to improve the contents and organization of the paper. This research was partially supported by Conselho Nacional de Desenvolvimento Científico e Tecnológico-CNPq, Fundação de Amparo à Pesquisa do Estado do Rio de Janeiro-FAPERJ, Coordenação de Aperfeiçoamento de Pessoal de nível Superior-CAPES, Brazilian research agencies.

References

[1] G. Alexe, S. Alexe, Y. Crama, S. Foldes, P. Hammer, B. Simeone, Consensus algorithms for the generation of all maximal bicliques, Discrete Appl. Math. 145 (1) (2004) 11–21.
[2] J. Amilhastre, M.C. Vilarem, P. Janssen, Complexity of minimum biclique cover and minimum biclique decomposition for bipartite domino-free graphs, Discrete Appl. Math. 86 (2–3) (1998) 125–144.
[3] H.-J. Bandelt, M. Farber, P. Hell, Absolute reflexive retracts and absolute bipartite retracts, Discrete Appl. Math. 44 (1–3) (1993) 9–20.
[4] M. Dawande, P. Keskinocak, J.M. Swaminathan, S. Tayur, On bipartite and multipartite clique problems, J. Algorithms 41 (2) (2001) 388–403.
[5] M. Garey, D. Johnson, Computers and Intractability, a Guide to the Theory of NP-Completeness, W.H. Freeman and Company, New York, 1979.
[6] M.C. Golumbic, C.F. Goss, Perfect elimination and chordal bipartite graphs, J. Graph Theory 2 (2) (1978) 155–163.
[7] D.S. Hochbaum, Approximating clique and biclique problems, J. Algorithms 29 (1) (1998) 174–200.

[8] D.S. Johnson, M. Yannakakis, C.H. Papadimitriou, On generating all maximal independent sets, Inform. Process. Lett. 27 (3) (1988) 119–123.
[9] R. Kumar, P. Raghavan, S. Rajagopalan, A. Tomkins, Trawling the web for emerging cyber communities, in: Proc. Eighth Internat. Conf. on World Wide Web, Elsevier, North-Holland, 1999, pp. 1481–1493.
[10] E.L. Lawler, J.K. Lenstra, A.H.G. Rinnooy Kan, Generating all maximal independent sets: NP-hardness and polynomial-time algorithms, SIAM J. Comput. 9 (3) (1980) 558–565.
[11] H. Müller, On edge perfectness and classes of bipartite graphs, Discrete Math. 149 (1–3) (1996) 159–187.
[12] J. Orlin, Containment in graph theory: covering graphs with cliques, Nederl. Akad. Wetensch. Indag. Math. 39 (1997) 211–218.
[13] M.C. Paull, S.H. Unger, Minimizing the number of states in incompletely specified sequential switching functions, IRE Trans. Electron. Comput. EC-8 (1959) 356–367.
[14] R. Peeters, The maximum edge biclique problem is NP-complete, Discrete Appl. Math. 131 (3) (2003) 651–654.
[15] E. Prisner, Bicliques in graphs. II. Recognizing k-path graphs and underlying graphs of line digraphs, Graph-theoretic Concepts in Computer Science (Berlin, 1997), Lecture Notes in Computer Science, Vol. 1335, Springer, Berlin, 1997, pp. 273–287.
[16] E. Prisner, Bicliques in graphs. I. Bounds on their number, Combinatorica 20 (1) (2000) 109–117.
[17] S. Tsukiyama, M. Ide, H. Ariyoshi, I. Shirakawa, A new algorithm for generating all the maximal independent sets, SIAM J. Comput. 6 (3) (1977) 505–517.
[18] Z. Tuza, Covering of graphs by complete bipartite subgraphs: complexity of 0–1 matrices, Combinatorica 4 (1) (1984) 111–116.
[19] M. Yannakakis, Node- and edge-deletion NP-complete problems, in: Conf. Record of the Tenth Ann. ACM Symp. on Theory of Computing (San Diego, California, 1978), ACM, New York, 1978, pp. 253–264.