R U T C O R
R E S E A R C H
R E P O R T

CONSENSUS ALGORITHMS FOR THE GENERATION OF ALL MAXIMAL BICLIQUES

Gabriela Alexe[a] Sorin Alexe[b]
Stephan Foldes[c] Peter L. Hammer[d]
Bruno Simeone[e]

RRR 23-2000, April, 2000

RUTCOR
Rutgers Center for
Operations Research
Rutgers University
640 Bartholomew Road
Piscataway, New Jersey
08854-8003
Telephone: 732-445-3804
Telefax: 732-445-5472
Email: rrr@rutcor.rutgers.edu
http://rutcor.rutgers.edu/~rrr

---

[a] RUTCOR, Rutgers University, 640 Bartholomew Road, Piscataway, NJ 08854-8003, USA. e-mail: alexe@rutcor.rutgers.edu
[b] RUTCOR, Rutgers University of New Jersey, 640 Bartholomew Road, Piscataway, NJ 08854-8003, USA.
e-mail: salexe@rutcor.rutgers.edu
[c] RUTCOR, Rutgers University, 640 Bartholomew Road, Piscataway, NJ 08854-8003, USA. e-mail: sfoldes@rutcor.rutgers.edu
[d] RUTCOR, Rutgers University, 640 Bartholomew Road, Piscataway, NJ 08854-8003, USA. e-mail: hammer@rutcor.rutgers.edu
[e] Department of Statistics, "La Sapienza" University, Piazzale Aldo Moro 5, 00185, Rome, Italy. e-mail: marsalis@rosd.sta.uniroma1.it

RUTCOR RESEARCH REPORT
RRR23-2000, April, 2000

CONSENSUS ALGORITHMS FOR THE GENERATION OF ALL MAXIMAL BICLIQUES

Gabriela Alexe Sorin Alexe
Stephan Foldes Peter L. Hammer
Bruno Simeone

Abstract. We describe a consensus-type algorithm for determining all the maximal complete bipartite (not necessarily induced) subgraphs of a graph. We show that by imposing a particular order in which the consensus type operations should be executed, this algorithm becomes totally polynomial. By imposing a further restriction on the way the algorithm has to be executed, we derive an improved variant of it, the complexity of which is bounded by a polynomial which is cubic in the input size, and only linear in the output size, and show its high efficiency on numerous computational experiments on randomly generated graphs with up to 1000 vertices and 6000 edges.

Acknowledgements: This work was partially supported by NSF grant DMS-9806389, and ONR grant N00014-92-J-1375.

1 Introduction

The problem of covering the edge set of a graph with a family of complete bipartite (not necessarily induced) subgraphs, initiated by Chung [Chu81], has received considerable attention in graph theory. It was proved in [Chu80] that the conjecture of Bermond [Ber78] asserting that $\lim_{n \to \infty} \frac{\rho(n)}{n} = 1$; where $\rho(n)$ is the minimum number of complete bipartite subgraphs that cover the edges of a graph with n vertices, is true. In [CES83], Chung, Erdős, and Spencer examined the similar problem of partitioning the edge set of a graph into complete bipartite subgraphs. Among the many contributions to this direction of research we mention Tuza [Tuz83], Idzik [BIK99], Tayur [DKT97], Hochbaum [Hoc98], Lundgren [DLS99].

It was noticed in [Ham78] that to every covering of the edge set of a graph G with a family F of complete bipartite subgraphs we can associate a pseudo-Boolean function, i.e. a real valued function with 0-1 variables, whose literals are in one-to-one correspondence with the complete bipartite subgraphs of F, and whose maximum equals the stability number of G. This observation lead to the establishment of various links between problems in graphs theory and pseudo-Boolean functions (see [HS79], [BBHS83], [BHS80], [CH89]). In particular, quadratic graphs were defined in [HS79] as graphs with the property that among the associated pseudo-Boolean functions there is at least one which can be represented as a degree two polynomial. Also, the same observation was central in the elaboration of the struction method [EHW84] for calculating the stability number of a graph; it was proved in [HMW85_1], [HMW85_2], and [GH88] that the struction method provides polynomial solutions for finding the stability number of certain classes of graphs. Also, Billionnet and Minoux [BM85] have used the same observation to show that the problem of maximizing a cubic supermodular function can be reduced to that of finding the weighted stability number of a bipartite graph.

Given a polynomial expression of a pseudo-Boolean function f, it is easy to construct a graph G whose stability number is equal to the maximum of f. Moreover, every family B of complete bipartite subgraphs of G covering its edge set defines a pseudo-Boolean function f_B; such that the maximum value of f and the maximum value of f_B are the same, for every family B covering the edge set of G. The number of variables of the pseudo-Boolean function f_B being equal to the number of complete bipartite subgraphs of the family $\mathcal{B}$, it is clearly advantageous to determine families $\mathcal{B}$ consisting of the smallest possible number of complete bipartite subgraphs covering the edge set of G. It is clear also that in order to achieve this goal we can restrict our attention to coverings which use only maximal bipartite subgraphs.

The goal of this paper is to describe an algorithm for generating all the maximal complete bipartite subgraphs of a graph. Let $G = (V, E)$ be a graph without loops and multiple edges, having the vertex set $V = \{1, 2, \ldots, n\}$ and the edge set E. In this paper we shall call the (maximal) bipartite – not necessarily induced – subgraphs of a graph, its (maximal) bicliques.

The Maximal Biclique Enumeration Problem (MBEP) consists in finding all the maximal bicliques of a given graph. The MBEP cannot be solved in polynomial time with respect to the input size, since the size of the output can be exponentially large. Indeed, consider for example a complete graph with n vertices. Since each proper partition of its vertex set into two subsets induces exactly one maximal biclique, the number of its maximal bicliques is $2^{n-1} - 1$. Also notice that the existence of an algorithm for solving MBEP polynomially in the input size only would imply the existence of a polynomial algorithm for solving the so-called Edge Biclique Problem, i.e. the problem of finding a maximum weight biclique of an edge weighted graph. This would contradict the NP-completeness of this problem, proved in [DKT97].

In this paper we shall describe a consensus-type algorithm for determining all the maximal complete bipartite (not necessarily induced) subgraphs of a graph. We shall show that by imposing a particular order in which the consensus type operations should be executed, this algorithm becomes totally polynomial. By imposing a further restriction on the way the algorithm has to be executed, we shall derive an improved variant of it, the complexity of which is bounded by a polynomial which is cubic in the input size, and only linear in the output size, and show its high efficiency on numerous computational experiments on randomly generated graphs with up to 1000 vertices and 6000 edges. In section 2 we develop a consensus type algorithm for MBEP, closely resembling the Blake and Quine classic methods for finding all the prime implicants of a Boolean function. In section 3 we show how to reduce the general MBEP to the special case of covering the edge set of a bipartite graph by bicliques. We show in section 4 that by applying the steps of the consensus method in a specific order we can reduce the exponential time algorithm given in section 2 to one of total polynomial complexity. In section 5 we further streamline the operations of the modified consensus type algorithm of the previous section and show that this variant of it produces all the maximal bicliques of the given graph in $O(n^3\beta)$ time, where n is the number of vertices of the graph and β is the number of its maximal bicliques. After describing in section 6 several classes of graphs for which the algorithm provides a polynomial solution to MBEP, we present in section 7 the results of a large set of computational experiments which clearly prove the efficiency of the proposed procedure.

2 A consensus-type algorithm

We shall describe below a method for solving MBEP in a way resembling the consensus method of Blake [Bla37] and Quine [Qui55] for finding the prime implicants of a Boolean function. According to [KP77], Malgrange has developed a consensus approach to find in a $0-1$ matrix the maximal submatrices consisting only of ones — a problem which is clearly equivalent to the MBEP in the case of bipartite graphs.

In order to define a consensus algorithm for finding the maximal bicliques we shall need some definitions.

Let G be a graph and X and Y two disjoint non-empty subsets of the vertex set with the property that every vertex in X is linked to every vertex in Y by an edge. The biclique of G having the bipartition sets X and Y will be denoted by (X, Y).

Let $B_1 = (X_1, Y_1)$ and $B_2 = (X_2, Y_2)$ be two bicliques of G. If $Y_1 \cap Y_2 \neq \emptyset$, we shall call $(X_1 \cup X_2, Y_1 \cap Y_2)$ a consensus of B_1 and B_2; symmetrically, if any of the conditions $X_1 \cap X_2 \neq \emptyset$, $X_1 \cap Y_2 \neq \emptyset$, $Y_1 \cap X_2 \neq \emptyset$ hold, we shall call the corresponding biclique $(X_1 \cap X_2, Y_1 \cup Y_2)$, $(Y_1 \cup X_2, X_1 \cap Y_2)$, $(X_1 \cup Y_2, Y_1 \cap X_2)$, a consensus of B_1 and B_2. In this way, a pair of bicliques may have $0, 1, 2, 3$, or 4 bicliques as its consensus.

The consensus-type algorithm will start with a collection C of bicliques covering the edge set of a graph G. Such a collection is easily available, for instance by simply considering all the individual edges of the graph, viewed as bicliques. A similar straightforward way of obtaining C is to define it as the collection of all the stars centered in the vertices of the graph G.

Using the above terminology we can now define a consensus algorithm as a sequence of transformations on the collection C. The method allows only two transformations, the absorption and the consensus adjunction — described below — and stops when none of these steps can be applied.

(i) Absorption: If the biclique (X_1, Y_1) in C contains the biclique (X_2, Y_2)

in C, i.e. $X_2 \subseteq X_1$ and $Y_2 \subseteq Y_1$; then remove $(X_2; Y_2)$ from C.

(ii) Consensus adjunction: For any two bicliques $B_1 = (X_1; Y_1)$ and $B_2 = (X_2; Y_2)$ in C, if any of the consensuses of B_1 and B_2 exists and is not absorbed by a biclique already in C, it will be added to C.

Two trivial observations are in order. First, if the collection C covers the edge set (i.e. every edge of G is contained in at least one of the bicliques of C), then this property will be preserved by both of the transformations above. Second, it is clear that the repeated application of the transformations above will always produce collections consisting only of bicliques of G.

The validity of the above described algorithm is based on the following:

Theorem 1. *If C is a collection of bicliques of the graph G which covers the edge set of G, and if $\hat{C}$ is the collection of bicliques obtained from C by repeating the transformations in the consensus type algorithm described above as many times as possible, then $\hat{C}$ consists of all the maximal bicliques of G.*

Proof. First of all, let us notice that the algorithm terminates after a finite number of steps since the total number of bicliques of G is finite, and a biclique which was removed from C (because of absorption) can never re-enter C.

Let us assume that after the termination of the algorithm the collection C does not contain all the maximal bicliques of G. Let then $B^* = (X^*; Y^*)$ be a maximal biclique of G, not contained in C, and having the minimum number of edges among all the maximal bicliques which are not in C. Since B^* is a biclique, none of the sets X^* and Y^* can be empty. Moreover, at least one of these two sets, say X^*, consists of two elements, since otherwise B^* would be an edge, and would be either in $\hat{C}$, or would be absorbed by a biclique in C.

Let us partition the set X^* into two proper subsets X' and X''. Clearly, $(X'; Y^*)$ and $(X''; Y^*)$ are bicliques having strictly fewer edges than B^*. By the minimality assumption it follows that both of these two bicliques are either contained in $\hat{C}$ or included in bicliques of $\hat{C}$. Clearly, their consensus $(X' \cup X''; Y^* \cap Y^*)$ is either contained in $\hat{C}$ or included in one of the bicliques of $\hat{C}$. However, this consensus is simply B^*.

It follows from the above that $\hat{C}$ contains all the maximal bicliques of G. Clearly it cannot contain any other biclique since that would be absorbed by one of the maximal ones. $\blacksquare$

Example 1. Consider the graph $G = (V, E)$, having the vertex set $V = \{a, b, c, d, e, f\}$ and the edge set $E = \{(a, b), (a, c), (a, d), (b, c), (b, e), (c, f)\}$ (see figure 1).

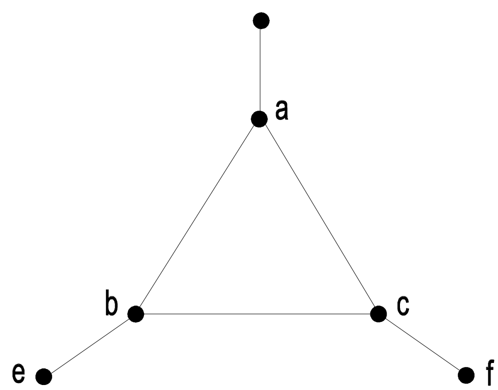

Figure 1: The graph G

Let us consider the following family of bicliques covering the edge set of G: $C = \{B_1, B_2, B_3\}$, where $B_1 = (\{a\}, \{b, d\})$, $B_2 = (\{b\}, \{e\})$, and $B_3 = (\{c\}, \{a, b, f\})$. Starting with C, the application of the above algorithm will result in the following steps:

1. The only consensus of B_1 and B_2 is $B_4 = (\{b\}, \{a, e\})$, which we shall add to C.
2. B_2 being absorbed by B_4, will be removed from C.
3. The consensuses of B_1 and B_3 are $B_5 = (\{a\}, \{b, c, d\})$, $B_6 = (\{b\}, \{a, c\})$, which will be added to C.
4. B_1 will be removed from C, being absorbed by B_5.
5. The consensuses of B_3 and B_4 are $(\{a\}, \{b, c\})$ (which being absorbed by B_5 will not be added to C) and $B_7 = (\{b\}, \{a, c, e\})$, which will be added to C.
6. We remove B_4 and B_6 from C, both being absorbed by B_7.

Any consensus of any of the remaining bicliques B_3, B_5, and B_7 is absorbed by one of these bicliques, and cannot be added to C.

Therefore, the algorithm terminates with the collection of all maximal bicliques of G: $\hat{C} = \{(\{a\};\{b;c;d\});(\{b\};\{a;c;e\});(\{c\};\{a;b;f\})\}$.

Obviously, the consensus adjunction and the absorption operations could have been applied in another order, but the resulting collection $\hat{C}$ – as shown by theorem 1 – would have been the same. We should remark that the above consensus algorithm may have an exponential running time, since as it will be shown in the example below, it can produce along the way a large number of non-maximal bicliques. However, as it will be seen in Section 4, by an adequate specification of the order in which the transformations of the consensus algorithm have to be carried out, the complexity of the algorithm can be guaranteed to be total polynomial in its input (number of vertices and of edges of G); and output (number of maximal bicliques).

Taking as C the collection of all the individual edges of $K_{n;n}$ (each viewed as a complete bipartite subgraph), we can apply the consensus method in such a way that in the first sequence of transformations we produce all the bicliques having exactly n vertices. However, the number of those bicliques is $\sum_{i=1}^{n-1}\binom{n}{i}\binom{n}{n-i} = 2^n - 2$; and therefore, the application of the above algorithm with this particular order of transformations runs in exponential time.

On the other hand, let $V' = \{v'_1;\ldots;v'_n\}$ and $V'' = \{v''_1;\ldots;v''_n\}$ be the vertices in the two bipartitions of $K_{n;n}$. By forming the consensus of $(\{v'_1\};\{v''_1\})$ with $(\{v'_1\};\{v''_2\})$; we obtain the biclique $(\{v'_1\};\{v''_1;v''_2\})$. In the next steps we shall form in a similar way the bicliques $(\{v'_1\};\{v''_1;v''_2;v''_3\})$ and so on, until we form $(\{v'_1\};V'')$. In a similar way we shall produce all the bicliques $(\{v'_i\};V'')$;
$i = 2;\ldots;n$. The number of transformations in this process is of $n(n-1)$ consensus adjunctions and $2n(n-1)$ absorptions. Forming now the consensuses of one of two $K_{1;n}$'s we obtain a $K_{2;n}$. The consensus of this $K_{2;n}$ with another $K_{1;n}$ is a $K_{3;n}$. Continuing in this way, in $n-1$ steps of consensus adjunction and $2(n-1)$ steps of absorptions, we shall produce $K_{n;n}$; showing that with a proper order of transformations, the running time of the algorithm is total polynomial (and in this particular case polynomial).

3 Reduction to the bipartite case

We shall discuss in this section a straightforward reduction of the MBEP to the special case of finding the maximal complete bipartite subgraphs of a bipartite graph. It will also be seen that this transformation will make it possible to define specific versions of the general consensus type algorithm described in Section 2, having guaranteed total polynomial complexity.

Let us consider a bipartite graph $G = (L, R, A)$, where L and R are two disjoint sets of elements called nodes, and A is a set of arcs of the type (l, r), with $l \in L$ and $r \in R$. Without loss of generality we shall always assume $|L| \le |R|$, and we shall call L the left set of nodes and R the right set of nodes. We shall also assume that the arcs are always oriented from L to R. A bipartite graph G satisfying the above assumptions on the cardinalities of its bipartition sets and on the orientation of its arcs will be simply called a bigraph. We shall denote the total number $|L| + |R|$ of vertices of a bigraph by n, the number $|A|$ of its arcs by m, and the number of its maximal bicliques by β.

Two non-empty subsets L' of L and R' of R will be called strongly adjacent, denoted as $L' ? R'$, if $(l, r) \in A$ for all $l \in L'$ and all $r \in R'$.

A biclique of G induced by two strongly adjacent subsets L' and R' of L and R respectively, will be simply denoted by (L', R'). Also, we shall call $L' = L(B)$ the left side of B, and $R' = R(B)$ the right side of B. The set of all maximal bicliques of the bigraph G will be denoted as $\mathcal{B}(G)$.

Given a graph $G = (V, E)$, with $V = \{v_1, \ldots, v_n\}$, we shall associate to it its double cover, i.e. a bigraph $2G := (L, R, A)$, where $L = \{l_1, \ldots, l_n\}$ and $R = \{r_1, \ldots, r_n\}$ are two disjoint copies of V, and where the oriented arc set A is $\{(l_i, r_j), (l_j, r_i) : i, j \in \{1, \ldots, n\}, (i, j) \in E\}$.

We remark that there is a one-to-one correspondence between the bicliques of G and the pairs of symmetric bicliques of $2G$. Indeed, let us consider a biclique (X, Y) of G and let us denote the copies of the sets X and Y in L and R as L_X, L_Y, R_X, R_Y, respectively. It is easy to notice now that there is a one-to-one correspondence between the bicliques (X, Y) of G and the pairs of symmetric bicliques $((L_X, R_Y), (L_Y, R_X))$. This remark allows us to reduce the general MBEP for a graph G to the problem of enumerating the maximal bicliques of the double cover $2G$ of G in linear time. In view of this remark, the algorithms to be described below will be formulated only for bigraphs, although it is clear that they could also be presented directly in terms of arbitrary graphs.

Let $B_1 = (L_1, R_1)$, $B_2 = (L_2, R_2)$ be two bicliques of the bigraph G. Since $L_1 \cap R_2$ and $L_2 \cap R_1$ are both empty, at most two consensuses of the bicliques B_1 and B_2 may be defined, namely the right consensus

$$B_1 \vee B_2 = (L_1 \cup L_2, R_1 \cap R_2) \quad (1)$$

if $R_1 \cap R_2 \neq \emptyset$, and the left consensus

$$B_1 \wedge B_2 = (L_1 \cap L_2, R_1 \cup R_2) \quad (2)$$

if $L_1 \cap L_2 \neq \emptyset$. Dulmage and Mendelsohn [DM 58] pointed out that (1) and (2) are the join and the meet, respectively, of a distributive lattice.

Let us consider a subset S of L and let us associate to it its neighborhood $\Gamma(S)$ in R. Similarly, let us associate to an arbitrary subset T of R its neighborhood $\Delta(T)$ in L. With these notations, it is clear that for any non-empty subset S of L, the sets $\Gamma(S)$ and $\Delta(\Gamma(S))$ are strongly adjacent; the bipartite subbigraph of G induced by $\Delta(\Gamma(S))$ and $\Gamma(S)$ will be called the biclique generated by S and denoted $B_\Gamma(S)$ (or $B(S)$ for short). In a symmetric way, for any non-empty subset T of R, the bipartite subbigraph of G induced by $\Delta(T)$ and $\Gamma(\Delta(T))$ will be called the biclique generated by T and denoted $B_\Delta(T)$ (or $B(T)$ for short). It is clear that different subsets S', S'' of L may define the same biclique $B(S') = B(S'')$. It is also clear that for every biclique $B = (L(B), R(B))$, there exist subsets S of L such that $B = B_\Gamma(S)$ (e.g. $S = L(B)$).

Given a maximal biclique B of G, we shall define the rank $\rho(B)$ by

$$\rho(B) = \min_{B = B_\Gamma(S)} |S| \quad (3)$$

and we shall denote the set of all maximal bicliques of rank l as $B_l(G)$. The rank of the bigraph G, denoted $\rho(G)$, is defined as $\rho(G) = \max\{\rho(B) : B$ maximal biclique of $G\}$.

The bicliques of rank one, which are the left centered stars $B_\Gamma(\{x\})$, $x \in L$, will play a special role in the method described below.

Remark 1. If L' and R' are non-empty subsets of L and R respectively, then the following statements are equivalent:

(i) $L' \perp R'$;

(ii) $R' \subseteq \Gamma(L')$;

(iii) $L' \subseteq \Delta(R')$.

Remark 2. If L', L'' and R', R'' are non-empty subsets of L and R, respectively, then the following relations can be easily established:

$$\Gamma(L' \cup L'') = \Gamma(L') \cap \Gamma(L'') \tag{4}$$

$$\Gamma(L' \cap L'') \supseteq \Gamma(L') \cup \Gamma(L'') \tag{5}$$

$$\Delta(R' \cup R'') = \Delta(R') \cap \Delta(R'') \tag{6}$$

$$\Delta(R' \cap R'') \supseteq \Delta(R') \cup \Delta(R'') \tag{7}$$

$$\Gamma(L') = \Gamma(\Delta(\Gamma(L'))) \tag{8}$$

$$\Delta(R') = \Delta(\Gamma(\Delta(R'))). \tag{9}$$

Note that a biclique B is maximal if and only if $L(B) = \Delta(\Gamma(L(B)))$ and $R(B) = \Gamma(L(B))$.

4 A totally polynomial version of the consensus algorithm

In this section we shall show that by appropriately specifying the order in which the consensus operations have to be carried out, the algorithm of the preceding section becomes totally polynomial. Other totally polynomial algorithms for generating different kinds of combinatorial objects are known in the literature, see e.g. Paull and Unger [PU59], and Tsukiyama, Ide, Ariyoshi, Shirakawa [TIAS77], Johnson, Yannakakis, and Papadimitriou [JYP88] (stable sets), Lawler, Lenstra, and Rinnooy Kan [LLK80] (independent sets in some classes of independence systems, and in particular induced complete $k-$partite subgraphs).

The new version of the consensus algorithm – which we shall call the modular consensus algorithm (MCA) assumes the knowledge of a family $\mathcal{C}$ of maximal bicliques covering the arc set of the bigraph; as it will be seen below, such a family can be found in linear time starting with an arbitrary covering of the arcs. Starting with the given family $\mathcal{C}$ of maximal bicliques, we select two such maximal bicliques B' and B'', form their consensuses $\bar{B}$ and $\hat{B}$, if they exist, and check whether they are contained in $\mathcal{C}$. If one of them, say $\bar{B}$, is not in $\mathcal{C}$, instead of simply adding $\bar{B}$ to $\mathcal{C}$, we shall first execute a sequence of consensus operations involving $\bar{B}$ and certain rank one bicliques of G. At each such step i, the biclique $\bar{B}_i$ will be absorbed by the resulting new biclique $\bar{B}_{i+1}$, and at the next step $\bar{B}_{i+1}$ will be absorbed by the new $\bar{B}_{i+2}$ etc. This "module" of consensus operations on $\bar{B}$ and its descendants ends when the sequence $\bar{B}, \bar{B}_1, \ldots,$ produces a maximal biclique $\bar{B}_*$. If $\bar{B}_*$ is not yet in $\mathcal{C}$ we include it. We repeat now all of the above steps for $\hat{B}$. At the end of the sequence, if $\hat{B}_*$ is not in $\mathcal{C}$ we add it, and repeat the above complete module by starting with another pair of bicliques in the family $\mathcal{C}' = \mathcal{C} \cup \{\bar{B}_*, \hat{B}_*\}$. The process ends when the consensus of no pair of bicliques in this family leads to the formation of new maximal bicliques.

In order to develop the details of the algorithm, we shall define now the modular consensus operations for two bicliques B' and B'' of a bigraph G in the following way:

- the right modular consensus

$$B' \sqcup B'' = (\cdot(R(B') \cap R(B'')), R(B') \cap R(B'')) \qquad (10)$$

  is defined if $R(B') \cap R(B'') \neq \emptyset$;

- the left modular consensus

$$B' \sqcap B'' = (L(B') \cap L(B''), -(L(B') \cap L(B''))) \qquad (11)$$

is defined if $L(B') \cap L(B'') \neq \emptyset$.

Lemma 1. Given the collection of all left centered stars $(B_{\Gamma}(\{x\}) : x \in L)$ of a bigraph $G = (L; R; A)$ having $m = |A|$ arcs, the right modular consensus $B' \sqcup B''$ of any two bicliques B', B'' of G can be obtained by performing at most n' right consensus adjunctions and n' absorptions, where $n' = \max\{|L|, |R|\} + 1$. The order in which these operations are carried out has no influence on the result, or on the complexity.

Proof. If the collection of the rank one maximal bicliques $(B_{\Gamma}(\{x\}) : x \in L)$ is given, then the right modular consensus $B' \sqcup B''$ of two bicliques B' and B'' can be obtained in polynomial time by performing the following right consensus adjunctions and absorptions:

- determine the right consensus $B^0 = B' \vee B''$
- initialize $i := 0$
- for every node $x \in L$:
    - if $R(B^i) \subseteq \Gamma(\{x\})$ then
        - determine the right consensus $B^{i+1} = B^i \vee B(\{x\})$
        - absorb B^i by B^{i+1}
        - increase i by 1

Note that on the one hand $R(B^0) = R(B^1) = \ldots = R(B^i)$; and on the other hand, $L(B^0) \subset L(B^1) \subset \ldots \subset L(B^i)$; where all inclusions are strict. Clearly, $L(B^i)$ is maximal with respect to the property that every vertex in it is adjacent to every vertex of $R(B^i)$; therefore $B^i = B' \sqcup B''$. The number of consensus adjunctions and absorptions in this algorithm cannot exceed n', respectively. The order in which the right consensus operations are performed does not influence neither the result, nor the complexity of the procedure. □

Similarly to lemma 1, we have:

Lemma 2. Given the collection of all right centered stars $(B_{\Delta}(\{y\}) : y \in R)$ of a bigraph $G = (L; R; A)$ having $m = |A|$ arcs, the left modular consensus $B' \sqcap B''$ of any two bicliques B', B'' of G can be obtained by performing at most n' left consensus adjunctions and n' absorptions, where $n' = \max\{|L|, |R|\} + 1$. The order in which these operations are carried out has no influence on the result, or on the complexity.

Remark 3. The right modular consensus $B' \sqcup B''$ can be determined directly in $O(m)$ time. For each vertex x in L we determine the number of its neighbors in $R(B' \sqcup B'')$. This number equals $|R(B' \sqcup B'')|$ if and only if $x \in ¢(R(B' \sqcup B''))$. A similar remark holds for left modular consensuses.

Remark 4. Having described how to obtain the two modular consensuses, if any, of two bicliques B' and B'', let us see now how to use these results for transforming a family of bicliques covering the edge set of a bigraph to a family of maximal bicliques with the same property. If B is a biclique of G, then $(B \sqcup B) \sqcap (B \sqcup B) = B_{¢}(R(B))$ and $(B \sqcap B) \sqcup (B \sqcap B) = B_{¡}(L(B))$ are both maximal bicliques of G, containing B, and it is easy to check that if $R(B) = ¡(L(B))$ or if $L(B) = ¢(R(B))$, then $(B \sqcup B) \sqcap (B \sqcup B) = (B \sqcap B) \sqcup (B \sqcap B)$.

It follows that given a family C of bicliques that covers the arcs of the bigraph G, we may obtain in polynomial time a family of maximal bicliques that cover the arcs of G by replacing each biclique B in C either with $(B \sqcup B) \sqcap (B \sqcup B)$ or with $(B \sqcap B) \sqcup (B \sqcap B)$. Note that given a family C of p bicliques that cover the arcs of G, we may obtain in $O(mp)$ a subfamily of at most m bicliques with the same property. Also, the family of all bicliques of rank one can be obtained from C in polynomial time.

Modular consensus algorithm (MCA)

- Start with a collection C of at most m bicliques that cover the arcs of the bigraph G
- Replace the collection C by a collection of maximal bicliques that cover the arcs of G, as in Remark 4
- For every pair of distinct bicliques B' and B'' in C determine their left and right modular consensuses, if any, and add them to C, unless already contained in C.

Thus, the modular consensus algorithm is a specialized version of the consensus algorithm which starts with a collection C of maximal bicliques covering the arcs of G, and in which the consensus operations are grouped in modules which have to be completed before starting any other consensus operation. Being a specialized version of the consensus algorithm, this algorithm terminates after a finite number of steps producing the collection of all maximal bicliques.

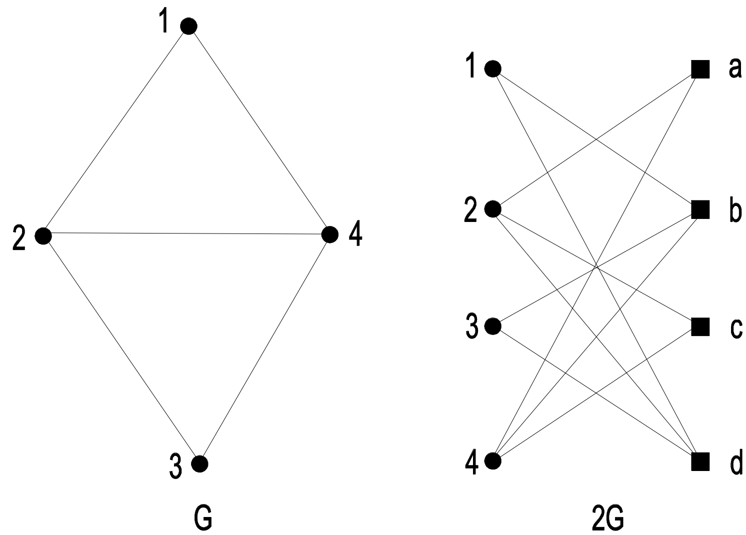

Figure 2: The graph G and its double cover 2G

Example 2. Consider the graph $G = (V; E)$ with $V = \{1; \ldots; 4\}$ and $E = \{(1;2); (1;4); (2;3); (2;4); (3;4)\}$. Its double cover is the bigraph $2G = (L; R; A)$; where $L = \{1; 2; 3; 4\}$; $R = \{a; b; c; d\}$, $A = \{(1;b); (1;d); (2;a); (2;c); (2;d); (3;b); (3;d); (4;a); (4;b); (4;c)\}$ (see figure 2). The modular consensus algorithm will consist of the following steps:

1. Start with the following collection of bicliques that cover the arcs of $2G$; $C = \{B_1; \ldots; B_4\}$; where $B_1 = (\{1; 3\}; \{d\})$; $B_2 = (\{2\}; \{a; c; d\})$; $B_3 = (\{1; 3; 4\}; \{b\})$; $B_4 = (\{4\}; \{a; c\})$.
2. Replace each biclique B from C by the maximal biclique $(B \sqcup B) \sqcap (B \sqcup B)$: $B_1 := (\{1; 3\}; \{b; d\})$; $B_2 := (\{2\}; \{a; c; d\})$; $B_3 := (\{1; 3; 4\}; \{b\})$; $B_4 := (\{2; 4\}; \{a; c\})$.
3. The only modular consensus of B_1 and B_2 is $B_5 = (\{1; 2; 3\}; \{d\})$ and we add it to C.
4. Both modular consensuses of B_1 are B_3 are already contained in C.
5. The pair of bicliques $B_1; B_4$ has no modular consensus.

6. Both modular consensuses of $\{B_1$ are $B_5\}$ are already contained in C.
7. The pair of bicliques B_2, B_3 has no modular consensus.
8. Both modular consensuses of $\{B_2$ are $B_4\}$ are already contained in C.
9. Both modular consensuses of $\{B_2$ are $B_5\}$ are already contained in C.
10. The pair of bicliques B_3, B_4 has no modular consensus.
11. The pair of bicliques B_3, B_5 has only one modular consensus, which is already in C.
12. The pair of bicliques B_4, B_5 has only one modular consensus, which is already in C.

The algorithm terminates with the collection $\hat{C} = \{B_1, \ldots, B_5\}$ of all maximal bicliques of G.

Theorem 2. The modular consensus algorithm runs in $O\left(\beta^2(m + n\log_2\beta)\right)$ time.

Proof. Since each maximal biclique can be obtained in at most $O(m)$ time, the transformation of the initial collection C into a collection of maximal bicliques that cover the arcs of G can be performed in at most $O(m^2)$ time. The operations (10) and (11) applied to maximal bicliques B' and B'' produce maximal bicliques, therefore at every step of the algorithm the collection C contains only maximal bicliques. The number of pairs of bicliques used for consensus formation is $O(\beta^2)$, and the number of operations needed for each pair of bicliques is of order $O(m + n)$. In order to avoid duplicates, for each new modular consensus it must be checked whether it has been already generated, and using binary search this can be done in $O(n\log_2\beta)$ time. Therefore, the total complexity is $O\left(\beta^2(m + n\log_2\beta)\right)$. Since $\beta \le 2^n$, it follows that $\log_2\beta \le O(n)$, and consequently, the complexity of the modular consensus algorithm is bounded by $O(\beta^2 n^2)$. □

Remark 5. We may obtain a computational improvement of the modular consensus algorithm if we initialize the collection C with the collection $\{B_-(\{x\}) : x \in L\}$ of all bicliques of rank one of G. Any maximal biclique B of G may be obtained by performing a sequence of right consensuses $(\ldots(B_-(\{x_1\}) \vee B_-(\{x_2\})) \vee \ldots) \vee B_-(\{x_p\})$, where $\{x_1, \ldots, x_p\} = L(B)$. Therefore, the right modular consensus operations are sufficient to generate all the maximal bicliques of G.

5 Modular input consensus algorithm (MICA)

A substantial improvement in the modular consensus algorithm can be obtained by noticing that many of the consensus adjunction operations produce the same outputs, and are therefore redundant. We shall show below that by giving proper care to the exact order in which the maximal bicliques of the collection C are listed, we can restrict the consensus adjunctions only to those situations in which one of the two "parents" is a biclique of rank one.

Let us remind that a biclique B is said to have rank l if the minimum cardinality of a set S such that $B = (\Delta(\Gamma(S)), \Gamma(S))$ is l.

The algorithm to be described in this section is essentially based on the following

Lemma 3. The right modular consensus of a biclique of rank l and of a biclique of rank 1 is a biclique of rank at most $l+1$.

Proof. Let $B' = B_\Gamma(A)$, $|A| = l$, be a biclique of rank l, and let $B'' = B_\Gamma(\{x\})$ be a left centered star of G, such that $R(B') \cap R(B'') \neq \emptyset$. Then $B' \sqcup B'' = (\Delta(\Gamma(A \cup \{x\})), \Gamma(A \cup \{x\}))$, and it follows that $\rho(B' \sqcup B'') \leq l+1$.

Conversely, let $B = B_\Gamma(A)$, $|A| = l+1$ be a biclique of rank $l+1$ of G. Let a be an arbitrary element of A and let $A' = A - \{a\}$. Note that both $\Gamma(\{a\})$ and $\Gamma(A')$ are nonempty, since $\Gamma(\{a\}) \cap \Gamma(A') = \Gamma(A)$, and therefore both bicliques $B_\Gamma(\{a\})$ and $B_\Gamma(A')$ are well-defined. Clearly, the rank of $B_\Gamma(\{a\})$ is 1 and the rank of $B_\Gamma(A')$ is at most l. Note also that if the rank of $B_\Gamma(A')$ is strictly less than l, then it follows from the first part of the proof that the rank of B is at most l. Therefore, $\rho(B_\Gamma(A')) = l$, completing the proof. ∎

The above lemma shows that we can replace the modular consensus algorithm by an accelerated version in which we only produce consensuses using pairs of maximal bicliques, at least one of which is a left centered star. Following the Boolean literature on similar consensus algorithms, we shall call this accelerated technique a modular input consensus algorithm.

As before, we shall denote by $B_l(G)$ the collection of maximal bicliques of rank l. Let us also denote $\bigcup_{i=1}^{l} B_i(G)$ by C_l. Within the framework of using the input consensus algorithm for constructing the list of maximal bicliques of rank l of a bigraph, we shall proceed successively by creating the lists $B_1(G), B_2(G), \ldots$ etc. Assume that at a certain step we have constructed all the maximal bicliques up to rank $l-1$, and a part of those of rank l, say $B_l^0(G)$. The lists we shall make use of to construct the remaining bicliques of rank l (if any) are $B_1(G)$, $B_l(G)$, C_l and $B_l^0(G)$.

Now we can state the

Modular input consensus algorithm (MICA)

- Start with the collection C of bicliques of rank one of G: $\{B_i(\{x\}) : x \in L\}$
- Assuming that the bicliques of rank $1, \ldots, l-1$ were constructed, construct the bicliques of rank l by sequentially forming all the right modular consensuses of bicliques of rank $l-1$ and bicliques of rank one, and add them to C, unless already there.

Theorem 3. *Given a bigraph G, the modular input consensus algorithm produces all the maximal bicliques of G.*

The validity of this theorem follows from the validity of the modular consensus algorithm and lemma 3.

Example 3. Consider the graph G and its double cover $2G$ as in Example 2.

1. Start with the collection of bicliques of rank one of $2G$:
$B_1(2G) = \{B_1, B_2, B_3\}$, where $B_1 = (\{1,3\}, \{b,d\})$, $B_2 = (\{2\}, \{a,c,d\})$, $B_3 = (\{4\}, \{a,b,c\})$; initialize C as $B_1(2G)$.
2. Initialize $B_2(2G)$ as $\emptyset$.
3. The right modular consensus of B_1 and B_2 is $B_4 = (\{1,2,3\}, \{d\})$ and we add it to $B_2(2G)$.
4. The right modular consensus of B_1 and B_3 is $B_5 = (\{1,3,4\}, \{b\})$ and we add it to $B_2(2G)$.
5. The right modular consensus of B_2 and B_3 is $(\{1,3,4\}, \{b\})$ and it is already contained in $B_2(2G)$.
6. Update C as $B_1(2G) \cup B_2(2G) = \{B_1, \ldots, B_5\}$.
7. Initialize $B_3(2G)$ as $\emptyset$.
8. The right consensuses of any biclique in $B_1(2G)$ and any biclique in $B_2(2G)$ is already in C.

Therefore, the algorithm terminates with the collection $C = \{B_1, \ldots, B_5\}$ of all maximal bicliques of $2G$.

The complexity of the modular input consensus algorithm is given by the following result which is proved in the Appendix:

Theorem 4. The modular input consensus algorithm runs in $O(n^2 \beta \log \beta)$ time.

One may notice that since $\log \beta = O(n)$; the complexity of the algorithm is also $O(n^3 \beta)$: The number β of maximal bicliques of a bigraph is not polynomial in the size of the bigraph. If the number of bicliques of a bigraph would be polynomial in the number of its nodes, then the problem of determining the maximum size clique of a graph would also be polynomially. This follows immediately from the one-to-one correspondence which can be established between the cliques of a graph $G = (V; E)$ with $V = \{v_1; \ldots; v_n\}$ and the bicliques of the bigraph G' obtained by adding the arcs $(l_i; r_i)$; $i = 1; \ldots; n$; to the double cover $2G$ (see e.g. [DKT97]). Consequently, the complexity of the modular input consensus algorithm is smaller by a factor β than the complexity of the modular consensus algorithm.

6 Some classes of bigraphs for which MBEP can be solved polynomially in the input size

As it was seen in the introduction, the number of bicliques of a bigraph can be exponential in its size. It is also obvious that the family of the double covers of graphs having maximal cliques of size $O(n)$ has an exponential number of maximal bicliques. On the other hand, numerous classes of specially structured graphs are known to have polynomial numbers of maximal cliques; clearly, the double covers of these graphs will all have polynomial numbers of maximal bicliques. Therefore { in view of theorem 2 { we have a polynomial solution for MBEP: In this section we shall present two such special classes.

6.1 Bounded degree bigraphs

Let d be a positive integer and let F_d be the family of all bigraphs whose node degrees are bounded by d: Then:

Theorem 5. MBEP is polynomial for bounded degree bigraphs.

Proof. Let G be a bigraph having all node degrees bounded by the constant d. The theorem is based on two facts. First, the number of arcs of this bigraph is at most nd. The second fact concerns the number of maximal bicliques containing a given arc, say xy ($x \in L$, $y \in R$), of the bigraph G. Clearly, each of these maximal bicliques, say (L', R') will be such that $L' \subseteq \Delta(\{y\})$ and $R' \subseteq \Gamma(\{x\})$. Since L' determines R' and since at most $2^d - 1$ choices of the set L' are possible, it follows that the number of maximal bicliques containing xy is at most $2^d - 1$.

The two rules above show that the total number β of maximal bicliques in a bounded degree bigraph is at most $nd2^d$. From theorem 4 it follows that the complexity of the modular input consensus is $O(n^2 \beta \log \beta)$, which for our class of graphs is $O(n^3 \log n)$. □

6.2 Convex bigraphs

Let G be a bigraph. G is called R-convex (see [Glo67]) if the nodes in R can be ordered in such a way that $\Gamma(\{x\})$, for any $x \in L$, is an interval consisting of all $\{y_{i(x)}, y_{i(x)+1}, \ldots, y_{j(x)}\}$.

Theorem 6. *The rank $\rho(B)$ of any maximal biclique of an R-convex bigraph G is at most 2, and therefore MBEP can be solved in polynomial time.*

Proof. Define

$$l = \max_{x \in L(B)} \min\{i : y_i \in \Gamma(\{x\})\} \tag{12}$$

and

$$u = \min_{x \in L(B)} \max\{j : y_j \in \Gamma(\{x\})\}. \tag{13}$$

Let $a \in L(B)$ be such that $l = \min\{i : y_i \in \Gamma(\{a\})\}$ and let $b \in L(B)$, such that $u = \max\{j : y_j \in \Gamma(\{b\})\}$. Since $R(B) = \bigcap_{x \in L(B)} \Gamma(\{x\}) = \Gamma(\{a\}) \cap \Gamma(\{b\}) = R(B_\Gamma(\{a, b\}))$, we have $B = B_\Gamma(\{a, b\})$. Therefore, $\rho(B) \leq 2$. □

Analogously, we can define an L-convex bigraph. In a symmetric way with $\rho(B)$ we may also define $\rho'(B)$ as

$$\rho'(B) = \min_{B = B_\Delta(T)} |T| \tag{14}$$

Similarly to theorem 6, we have:

Theorem 7. The rank $\rho'(B)$ of any maximal biclique of an L-convex bigraph G is at most 2; and therefore MBEP can be solved in polynomial time.

7 Computational results

We have run a large number of tests using as input the double covers of randomly generated graphs with 20 to 1000 vertices and 100 to 6000 edges. We have run experiments in three batches.

The purpose of the experiments was to determine the complexity coefficients which appear in the worst case evaluations of $O(\beta^2(m + n \log_2 \beta))$ for the modular consensus algorithm and $O(n^2 \beta \log \beta)$ for the modular input consensus algorithm.

The results of these experiments are presented in tables 1 - 4. In these tables, $n; m$ represent respectively the number of vertices and edges in the randomly generated graphs, while β represents the number of maximal bicliques in their double covers. We have denoted by k_1 and k_2 respectively, the number of comparisons between pairs of bicliques generated during the execution of the two algorithms. Similarly, q_1 and q_2 represent the number of consensus operations executed by the two algorithms.

Based on the number of operations $k_1 n + q_1 m$ executed in these experiments by the modular consensus algorithm MCA; we have determined an approximate value of the constant c_1 appearing in the worst case bound $c_1\beta^2(m + n \log_2 \beta)$ on the number of operations. In a similar way, based on the number of operations $k_2 n + q_2 m$ executed in these experiments by the modular input consensus algorithm MICA; we have determined an approximate value of the constant c_2 appearing in the worst case bound $c_2 n^2 \beta \log \beta$ on the number of operations needed.

One of the most surprising conclusions of these calculations, as it can be seen in tables 1 - 4, is that the values of the constants c_1 and c_2 remained practically unchanged along these experiments, both of them being extremely low in all the experiments, without any exception. For example, c_1 maintained a value very close to 2.7 in all the 20 experiments involving graphs with 20 vertices and 100 edges. Its value was always very close to 1.8 in all the 20 experiments involving 25 vertices and 135 – 157 edges. The constant c_2 behaved in a very similar way, maintaining a value close to 1.4 in all the experiments in table 1, very close to 1.34 in all the experiments in table 2; close to 0.8 in all the experiments in table 3 and to 0.13 in all experiments in table 4.

Another interesting conclusion from these experiments is the clear computational superiority of the algorithm MICA, in comparison with MCA. As a matter of fact, the ratio of the number of operations required by the two algorithms varies between 20 and 60, depending on the graph.

However, it is clear that the most important conclusion of the experiments is the extreme efficiency of MICA in handling very large graphs. It should be added that the computational times required even by the largest graphs experimented with were extremely reasonable.

	Input parameters		Output parameter	Modular Consensus Algorithm			Modular Input Consensus Algorithm			
experiment	**n = # of vertices**	**m = # of edges**	**beta = # of maximal bicliques**	**K1 = # comparisons (in thousands)**	**Q1 = # consensuses (in thousands)**	$C_1 = \frac{K_1 * n + Q_1 * m}{O\left(b\left(n^2 \log_2 b + mb\right)\right)}$	**K2 = # comparisons (in thousands)**	**Q2 = # consensuses (in thousands)**	$C_2 = \frac{K_2 * n + Q_2 * n}{O\left(n^2 b \log_2 b\right)}$	$\frac{K_1 n + Q_1 m}{K_2 n + Q_2 n}$
1	20	100	600	3496	361	2.774	137	15	1.374	35
2	20	100	646	4007	418	2.762	158	16	1.438	35
3	20	100	482	2110	233	2.625	109	12	1.419	27
4	20	100	688	4457	474	2.735	178	18	1.510	35
5	20	100	478	1969	229	2.537	125	14	1.630	22
6	20	100	434	1713	189	2.610	90	11	1.329	26
7	20	100	574	3216	330	2.777	128	14	1.348	34
8	20	100	580	3226	337	2.746	128	14	1.333	35
9	20	100	682	4418	466	2.749	168	17	1.444	36
10	20	100	574	3168	330	2.750	126	14	1.327	35
11	20	100	538	2688	290	2.679	129	13	1.463	29
12	20	100	574	3208	330	2.772	128	14	1.356	34
13	20	100	624	3754	390	2.765	141	16	1.348	37
14	20	100	542	2815	294	2.735	117	13	1.319	33
15	20	100	488	2129	239	2.600	110	13	1.411	27
16	20	100	498	2205	249	2.594	125	13	1.553	25
17	20	100	582	3237	339	2.740	131	14	1.358	34
18	20	100	562	2905	316	2.668	142	15	1.529	29
19	20	100	514	2420	265	2.649	115	12	1.374	29
20	20	100	670	4269	450	2.749	165	16	1.443	36
		Mean	567	3070	326	**2.701**	133	14	**1.415**	**31.631**
		St dev	72	830	82	**0.074**	22	2	**0.087**	**4.349**

Table 1: Comparative analysis of MCA=MICA for random graphs with $n = 20; m = 100$

	Input parameters		Output parameter	Modular Consensus Algorithm			Modular Input Consensus Algorithm			
experiment	n = # of vertices	m = # of edges	beta = # of maximal bicliques	K1 = # comparisons (in thousands)	Q1 = # consensuses (in thousands)	$C_1 = \frac{K_1 * n + Q_1 * m}{O(b(n^2 \log_2 b + mb))}$	K2 = # comparisons (in thousands)	Q2 = # consensuses (in thousands)	$C_2 = \frac{K_2 * n + Q_2 * n}{O(n^2 b \log_2 b)}$	$\frac{K_1 n + Q_1 m}{K_2 n + Q_2 n}$
1	25	157	1672	17373	2791	1.938	620	57	1.511	52
2	25	146	1174	7133	1375	1.816	370	37	1.358	37
3	25	130	800	2667	638	1.699	209	23	1.202	26
4	25	139	932	4222	866	1.786	279	30	1.342	29
5	25	155	1622	13520	2626	1.780	501	48	1.270	54
6	25	142	1008	4339	1013	1.676	280	30	1.231	33
7	25	135	776	2866	600	1.776	228	24	1.354	24
8	25	151	1292	10295	1665	1.954	446	43	1.463	42
9	25	140	962	3720	923	1.639	260	27	1.206	31
10	25	154	1336	10920	1781	1.930	458	44	1.446	44
11	25	155	1728	15162	2981	1.773	535	51	1.261	57
12	25	143	1062	6586	1125	1.938	359	36	1.479	33
13	25	156	1596	14501	2543	1.861	555	52	1.430	50
14	25	156	1506	12138	2264	1.805	488	48	1.348	49
15	25	157	1836	19958	3365	1.897	643	58	1.408	59
16	25	146	1252	7315	1564	1.736	364	36	1.244	41
17	25	146	1142	6365	1301	1.766	345	35	1.311	37
18	25	144	1020	4698	1037	1.708	301	31	1.303	32
19	25	147	1242	8303	1539	1.848	391	39	1.348	40
20	25	138	1014	4742	1025	1.754	295	30	1.285	32
Mean		147	1249	8841	1651	**1.804**	396	39	**1.340**	**40.085**
St dev		8	317	5130	830	**0.093**	129	11	**0.093**	**10.473**

Table 2: Comparative analysis of MCA/MICA for random graphs with $n = 25$

experiment	Input parameters: n = # of vertices	m = # of edges	Output parameter: beta = # of maximal bicliques	Modular Input Consensus Algorithm: K2 = # comparisons (in thousands)	Q2 = # consensuses (in thousands)	$C_2 = \frac{K_2 * n + Q_2 * n}{O(n^2 b \log b)}$
1	100	1000	11802	11541	1282	0.803
2	100	1000	12368	12783	1367	0.842
3	100	1000	11742	11408	1271	0.799
4	100	1000	12042	12032	1316	0.818
5	100	1000	11922	11812	1298	0.812
6	100	1000	11970	12046	1317	0.824
7	100	1000	11940	11913	1309	0.818
8	100	1000	11728	11428	1278	0.801
9	100	1000	11744	11471	1279	0.803
10	100	1000	11906	11840	1301	0.815
		Mean	11916	11827	1302	**0.813**
		St dev	192	416	28	**0.013**

Table 3: Computations with MICA for random graphs with n = 100; m = 1000

experiment	Input parameters: n = # of vertices	m = # of edges	Output parameter: beta = # of maximal bicliques	Modular Input Consensus Algorithm: K2 = # comparisons (in thousands)	Q2 = # consensuses (in thousands)	$C_2 = \frac{K_2 * n + Q_2 * n}{O(n^2 b \log b)}$
1	1000	6000	6438	4622	6435	0.136
2	1000	6000	6114	4436	6111	0.137
3	1000	6000	6284	4491	6282	0.136
4	1000	6000	6518	4607	6516	0.135
5	1000	6000	6398	4584	6396	0.136
6	1000	6000	6372	4558	6369	0.136
7	1000	6000	6370	4565	6368	0.136
8	1000	6000	6274	4522	6272	0.136
9	1000	6000	6244	4513	6242	0.137
10	1000	6000	6258	4481	6256	0.136
		Mean	6327	4538	6325	**0.136**
		St dev	115	59	115	**0.001**

Table 4: Computations with MICA for random gaphs with n = 1000; m = 6000

8 Appendix

In the following we shall provide the pseudo-code for the implementation of the modular input consensus algorithm. In this implementation, a biclique B is recorded as a set $\Gamma(X)$, where X is a set of minimum size such that $B = B_{\Gamma}(X)$. The various lists of bicliques used in the process are lexicographically ordered with respect to the corresponding sets $\Gamma(X)$.

The implementation uses the following procedures and functions:

found searches if a biclique B is in a certain list of maximal bicliques; *found* returns true if B is in the list, and false otherwise;

add an element B to a lexicographically ordered list L: uses *found* in order to determine if B is in L and if *found* returns false, *add* updates the list L, by inserting the element B in L, while maintaining the lexicographic order;

interclass a list Q with a list L: creates a new lexicographically ordered list by merging the lists Q and L.

Modular input consensus algorithm

```
Step 1
begin
    initiate l := 1;
    define C_l := B_1(G);
end;

Step 2
while B_l(G) is not empty do
    begin
        l := l + 1;
        initiate B'_l(G) to empty list;
        for every biclique B' in B_1(G) do
            for every biclique B'' in B_l(G) do
                if R(B') ∩ R(B'') ≠ ∅, then
                    B := B' ⊔ B'';
                    if B not found in C_{l-1} then
                        add B to B'_l(G);
        C_l := interclass(B'_l(G), C_{l-1});
        B_l(G) := B'_l(G);
```

end;
define $C := C_l$;

Theorem 4. The modular input consensus algorithm runs in $O(n^2\beta\log\beta)$ time.

Proof. Let n be the number of nodes in a bipartition set of G; m the number of arcs of G, r the rank of G; β_l the size of the set $B_l(G)$; and β the number of maximal bicliques of the bigraph G. Clearly, $\sum_{l=1}^{r}\beta_l = \beta$ and $r \le n$.

The complexities of the basic procedures we perform are the following:

initiate: $O(1)$;

found by binary search an element of size u in a list of size v: $O(u\log_2 v)$;

add an element of size u to a list of size v (without duplicates) – the same as found: $O(u\log_2 v)$;

define a list of size u as a given list Q of size u: $O(u)$;

intersection of two sets of size u: $O(u)$;

subset of a set of size u: $O(u)$;

interclass two lists of size u and v, respectively: $O(u+v)$.

Next we evaluate the complexities of the two steps.

The complexity of Step 1 is $O(\text{initiate} + n\cdot(\text{found} + \text{add}) + 2\cdot\text{copy}) = O(1 + 2n^2\log_2 n + 2n) = O(n^2\log n)$.

The complexity of Step 2 is

$$O\left(\sum_{l=1}^{r}(\text{initiate} + n\beta_l(\text{subset} + \text{intersection} + \text{found} + \text{add} + \text{interclass} + \text{define}))\right) =$$

$$O\left(\sum_{l=1}^{r}\left(1 + n\beta_l\left[2n + n\left[\log_2\beta_{l+1} + \log_2\beta\right] + n(\beta + \beta_l) + \beta_{l+1}\right)\right)\right) = O(r + 2n^2\beta + n^2\beta\log\beta + rn\beta + n\beta + \beta) = O(n^2\beta\log\beta).$$

Therefore, the complexity of the accelerated algorithm is $O(n^2\beta\log\beta)$.

References

[BBHS83] Benzaken, Cl.; Boyd, S.; Hammer, P. L.; Simeone, B. Adjoints of pure bidirected graphs. Congressus Numerantium 39 (1983), 123-144.

[BHS80] Benzaken, Cl.; Hammer, P. L.; Simeone, B. Some remarks on conflict graphs of quadratic pseudo-Boolean functions. Internat.

Ser. Numer. Math Birkhäuser, Basel-Boston, Mass., 55 (1980), 9–30.

[Ber78] Bermond, J. C. Couvertures des arrêtes d'un graphe par des graphes bipartis complets. Preprint, Univ. De Paris Sud, Centre d'Orsay, Rapport de Recherche No. 10 (June 1978).

[BM85] Billionnet, A.; Minoux, M. Maximizing a supermodular pseudo-Boolean function: a polynomial algorithm for supermodular cubic functions. Discrete Appl. Math. 12 (1985), no. 1, 1–11.

[Bla37] Blake, A. Canonical expressions in Boolean algebra, Ph.D. Thesis, University of Chicago, 1937.

[BIK99] Bylka, S.; Idzik, A.; Komar, J. Bipartite Subgraphs of Graphs with Maximum Degree Three, Graphs and Combinatorics 15 (1999), 129-136.

[Chu80] Chung, F. R. K. On the coverings of graphs. Discrete Math. 30 (1980), no. 2, 89–93.

[Chu81] Chung, F. R. K. On the decomposition of graphs. SIAM J. Algebraic Discrete Methods 2 (1981), no. 1, 1–12.

[CES83] Chung, F. R. K.; Erdős, P.; Spencer, J. On the decomposition of graphs into complete bipartite subgraphs. Studies in pure mathematics, Birkhäuser, Basel-Boston, Mass. (1983), 95–101.

[CH89] Crama, Y.; Hammer, P. L. Recognition of quadratic graphs and adjoints of bidirected graphs. Ann. New York Acad. Sci. New York Acad. Sci., New York, 555 (1989), 140–149.

[DKT97] Dawande, M.; Keskinocak, P.; Tayur, S. On the Complexity of the biclique problem in bipartite graphs, GSIA Working Paper 1996-04, Carnegie-Melon University, 1997.

[DLS99] Doherty, F. C. C.; Lundgren, J. R.; Siewert, D. J. Biclique Covers and Partitions of Bipartite Graphs and Digraphs and Related Matrix Ranks of {0,1}-Matrices, Congressus Numerantium 136 (1999), 73-96.

[DM58] Dulmage, A. L.; Mendelsohn, N. S. Coverings of bipartite graphs. Canadian J. Math. 10 (1958), 517-534.

[EHW84] Ebenegger, Ch.; Hammer, P. L.; de Werra, D. Pseudo-Boolean functions and stability of graphs. Annals of Discrete Mathematics 19 (1984), 83-97.

[Glo67] Glover, F. Maximum Matching in a Convex Bipartite Graph. Naval Res. Logist. Quart. 14 (1967), 313-316.

[GH88] Golumbic, M. C.; Hammer, P. L. Stability in circular arc graphs. J. Algorithms 9 (1988), no. 3, 314–320.

[Ham78] Hammer, P. L. The conflict graph of a pseudo-Boolean function. Technical Report, Bell Laboratories, West Long Branch, NJ, 1978.

[HS79] Hammer, P. L.; Simeone, B. Quasimonotone Boolean functions and bistellar graphs. Ann. Discrete Math. 9 (1980), 107–119.

[HMW85_1] Hammer, P. L.; Mahadev, N. V. R.; de Werra, D. The struction of a graph: application to CN-free graphs. Combinatorica 5 (1985), no. 2, 141–147.

[HMW85_2] Hammer, P. L.; Mahadev, N. V. R.; de Werra, D. Stability in CAN-free graphs. J. Combin. Theory Ser. B 38 (1985), no. 1, 23–30.

[Hoc98] Hochbaum, D. S. Approximating clique and biclique problems. J. Algorithms 29 (1998), no. 1, 174–200.

[JYP88] Johnson, D. S.; Yannakakis, M.; Papadimitriou, C. H. On generating all maximal independent sets. Inform. Process. Lett. 27 (1988), no. 3, 119–123.

[KP77] Kaufmann, A.; Pichat, E. Méthodes mathématiques non numériques et leurs algorithmes, Tome I, Algorithmes de recherche des éléments maximaux, Masson, Paris, 1977.

[LLK80] Lawler, E. L.; Lenstra, J. K.; Rinnooy Kan, A. H. G. Generating all maximal independent sets: NP-hardness and polynomial-time algorithms. SIAM J. Comput. 9 (1980), no. 3, 558–565.

[Mal62] Malgrange, Y. Recherche des sous-matrices premieres d'une matrice a coefficients binaires. Applications a certains problemes de graphe, dans Deuxieme Congres de l'AFCALTI, octobre 1961, Gauthier-Villars, 1962, 231-242.

[PU59] Paull, M.C.; Unger, S.H. Minimizing the number of states in incompletely specified sequential switching functions. IRE Trans. Electronic Computers, EC-8 (1959), 356-367.

[Qui55] Quine, W. A way to simplify truth functions, American Mathematical Monthly, 62, (1955), 627-631.

[TIAS77] Tsukiyama, S.; Ide, M.; Ariyoshi, H.; Shirakawa, I. A new algorithm for generating all the maximal independent sets. SIAM J. Comput. 6 (1977), no. 3, 505{517.

[Tuz83] Tuza, Z.: Covering of graphs by complete bipartite subgraphs; complexity of 0-1 matrices, Combinatorica 4 (1984), no 1, 111-116.