Exact exponential-time algorithms for finding bicliques in a graph

Henning Fernau [a] Serge Gaspers [b] Dieter Kratsch [c]
Mathieu Liedloff [d,*] Daniel Raible [a]

[a] *Universität Trier, FB 4—Abteilung Informatik, D-54286 Trier, Germany*

[b] *LIRMM – University of Montpellier 2, CNRS, 34392 Montpellier, France*

[c] *LITA, Université Paul Verlaine - Metz, 57045 Metz Cedex 01, France*

[d] *LIFO, Université d'Orléans, 45067 Orléans Cedex 2, France*

Key words: graph, exact exponential-time algorithm, NP-hard problem, biclique

1 Introduction

Throughout the paper all graphs $G = (V, E)$ are undirected and simple. An *induced biclique* of G is a complete bipartite induced subgraph of G. A *non-induced biclique* is a complete bipartite (not necessarily induced) subgraph of G. Equivalently, the pair (X, Y) of disjoint vertex subsets $X \subseteq V$ and $Y \subseteq V$ is a non-induced biclique of G if $\{x, y\} \in E$ for all $x \in X$ and $y \in Y$. If, additionally, X and Y are independent sets, then (X, Y) is also an induced biclique of G. Let the pair (X, Y) be an induced or non-induced biclique. We call it a (k_1, k_2) biclique if $|X| = k_1$ and $|Y| = k_2$. Its cardinality is $|X| + |Y|$.

The literature dealing with bicliques is rich and diverse. There are applications of bicliques (induced or non-induced on bipartite graphs or general graphs) in various different areas such as data mining, automata and language theory, artificial intelligence and biology, see e.g. [1]. Therefore bicliques and algorithmic problems about bicliques have been studied extensively.

* Corresponding author.
Email addresses: `fernau@uni-trier.de` (Henning Fernau), `gaspers@lirmm.fr` (Serge Gaspers), `kratsch@univ-metz.fr` (Dieter Kratsch), `liedloff@univ-orleans.fr` (Mathieu Liedloff), `raible@uni-trier.de` (Daniel Raible).

Known results. Already in [3], the complexity of finding certain bicliques has been considered. For example, deciding whether a bipartite graph has a balanced biclique of size (at least) k is NP-complete ([GT24] in [3]). A maximum cardinality induced biclique can be computed in polynomial time on bipartite graphs [2], wheras this problem is NP-complete for general graphs [7]. Another related problem that asks to compute a non-induced biclique with a maximum number of edges is also known to be NP-hard [6].

The above-mentioned NP-completeness of the balanced biclique problem on bipartite graphs implies the NP-completeness of the following two problems about the existence of induced and non-induced bicliques, respectively.

Induced $(\mathbf{k_1}, \mathbf{k_2})$ Biclique
Input: An undirected graph $G = (V, E)$, positive integers k_1 and k_2.
Question: Does G have an induced (k_1, k_2) biclique (X, Y)?

Non-Induced $(\mathbf{k_1}, \mathbf{k_2})$ Biclique
Input: An undirected graph $G = (V, E)$, positive integers k_1 and k_2.
Question: Does G have a non-induced (k_1, k_2) biclique (X, Y)?

There is a trivial $O^*(3^n)$ algorithm for finding and also for enumerating all induced and non-induced (k_1, k_2) bicliques, respectively. [1] It considers all partitions of the vertex set into X, Y and $V \setminus (X \cup Y)$ and verifies for each whether (X, Y) fulfils all conditions.

Our results. For generating all non-induced (k_1, k_2) bicliques, note that there is no hope in obtaining a faster algorithm than the above-described $O^*(3^n)$ algorithm, as a complete graph on n vertices has $3^n \mathsf{poly}(n)$ non-induced $(\lfloor n/3 \rfloor, \lfloor n/3 \rfloor)$ bicliques. For solving the **Non-Induced $(\mathbf{k_1}, \mathbf{k_2})$ Biclique** problem, however, we give a polynomial-space $O(1.8899^n)$ algorithm and an exponential-space $O(1.8458^n)$ algorithm.

There is also an $O^*(3^{n/3})$ time algorithm to solve **Induced $(\mathbf{k_1}, \mathbf{k_2})$ Biclique**. This algorithm is based on enumerating all maximal induced bicliques of the graph with a polynomial delay algorithm and on the fact that an n-vertex graph has $O^*(3^{n/3})$ maximal induced bicliques [4].

2 Polynomial-space algorithms for finding a non-induced biclique

We start by describing two simple $O^*(2^n)$ time algorithms for the **Non-Induced $(\mathbf{k_1}, \mathbf{k_2})$ Biclique** problem. We will use these algorithms as sub-

[1] Throughout the paper we write $f(n) = O^*(g(n))$ if $f(n) \le p(n) \cdot g(n)$ for some polynomial $p(n)$.

routines in our third algorithm with running-time $O(1.8899^n)$ and polynomial space usage.

The first algorithm, `NIB1`, verifies all sets $X_c \subseteq V$ with $|X_c| = k_1$ as candidates for being the set X in the pair (X, Y). It computes for each X_c the set $B(X_c) := \{v \in V \setminus X_c \mid \forall x \in X_c : v \in N(x)\}$. If $|B(X_c)| < k_2$ then the algorithm rejects the candidate X_c. Otherwise it picks an arbitrary set $Y_c \subseteq B(X_c)$ with $|Y_c| = k_2$, and clearly (X_c, Y_c) is a non-induced (k_1, k_2) biclique. The only exponential part is the enumeration step and thus the running-time is $O^*(2^n)$.

The second algorithm, `NIB2`, verifies all sets $U \subseteq V$ with $|U| = k_1 + k_2$ and checks for each whether there is a non-induced biclique (X, Y) such that $|X| = k_1$ and $|Y| = k_2$. This can be done in polynomial time by computing the connected components of the complement of $G[U]$. If $s_1, s_2, \ldots, s_t$ are the sizes of those components, then there is a non-induced biclique as described above iff there is an $I \subseteq \{1, 2, \ldots, t\}$ such that $\sum_{i \in I} s_i = k_1$. Such a Subset Sum problem can be solved in time $O(nW)$ by dynamic programming, where $W = \max_i s_i$. The only exponential part is the enumeration step and thus the running-time is $O^*(2^n)$.

For the third algorithm, `NIB3`, suppose w.l.o.g. that $k_1 \leq k_2$. If $k_1 \leq \lfloor n/3 \rfloor$, then run `NIB1`, otherwise run `NIB2`. Thus, the running-time of `NIB3` is at most $O^*\left(\binom{n}{n/3}\right) = O(1.8899^n)$ if `NIB1` is executed and at most $O^*\left(\binom{n}{2n/3}\right) = O(1.8899^n)$ if `NIB2` is executed.

Theorem 2.1 *Algorithm* `NIB3` *solves the* **Non-Induced $(\mathbf{k_1}, \mathbf{k_2})$ Biclique** *problem in time $O(1.8899^n)$ and polynomial space.*

3 Exponential-space algorithm for finding a non-induced biclique

In this section we provide an exponential-space algorithm for the the **Non-Induced $(\mathbf{k_1}, \mathbf{k_2})$ Biclique** problem in time $O(1.8458^n)$. The algorithm relies on a preprocessing involving a dynamic programming approach. It is described by the forthcoming three steps called *Partitioning*, *Preprocessing* and *Computing.*

3.1 Description of the algorithm

Partitioning Step. Let α be a constant to be determined. Given the graph $G = (V, E)$, compute an arbitrary partition of the vertex set into two subsets L and R such that $|R| = \lceil \alpha n \rceil$ and $|L| = \lfloor (1 - \alpha) n \rfloor$.

Preprocessing Step. This step focuses on the vertices of R. For any two (not necessarily disjoint) subsets $X, Y \subseteq R$ and any two integers i and j, $0 \leq i, j \leq |R|$, we compute the value of the boolean $\textsf{R-biclique}[X, i, Y, j]$ which is true iff there exist two subsets $X' \subseteq X$ and $Y' \subseteq Y$ such that (X', Y') is a non-induced (i, j) biclique. To compute the values of R-biclique, the sets X, Y and the integers i, j are considered by increasing cardinality and order.

For any $X, Y \subseteq R$ and i, j, such that $0 \leq i, j \leq |R|$, $\textsf{R-biclique}[X, 0, Y, 0]$ is clearly true. Obviously $\textsf{R-biclique}[\emptyset, i, Y, j]$ is true iff $i = 0$ and $\textsf{R-biclique}[X, i, \emptyset, j]$ is true iff $j = 0$. For any other value, $\textsf{R-biclique}[X, i, Y, j]$ is true iff

$$\bigvee_{v \in X} \Big(\textsf{R-biclique}[X \setminus \{v\}, i, Y, j] \vee \textsf{R-biclique}[X \setminus \{v\}, i-1, N(v) \cap Y, j]\Big) \vee$$
$$\bigvee_{v \in Y} \Big(\textsf{R-biclique}[X, i, Y \setminus \{v\}, j] \vee \textsf{R-biclique}[N(v) \cap X, i, Y \setminus \{v\}, j-1]\Big).$$

Computing step. If the graph admits a non-induced (k_1, k_2) biclique, then it is found during this final step. For every two disjoint subsets $X_L, Y_L \subseteq L$ for which (X_L, Y_L) is a non-induced biclique with $|X_L| \leq k_1$, $|Y_L| \leq k_2$, let $X'_R = \{v \in R : v \text{ is adjacent to every vertex of } Y_L\}$ and $Y'_R = \{v \in R : v \text{ is adjacent to every vertex of } X_L\}$; if $\textsf{R-biclique}[X'_R, k_1 - |X_L|, Y'_R, k_2 - |Y_L|]$ is true then the graph has a non-induced (k_1, k_2) biclique and "Yes" is returned.

If the algorithm was not able to find any X_L, Y_L such that $\textsf{R-biclique}[X'_R, k_1 - |X_L|, Y'_R, k_2 - |Y_L|]$ is true, then the graph has no non-induced (k_1, k_2) biclique and it returns "No". The correctness of the algorithm is shown in the next section. Note that instead of returning "Yes" or "No", our algorithm can easily be modified (by standard backtracking techniques) to indeed return a non-induced (k_1, k_2) biclique if one exists.

3.2 Correctness of the algorithm

Assume that G has a non-induced (k_1, k_2) biclique and let (X, Y) be such a biclique. Since (L, R) is a partition of V, it holds that $X = X_L \cup X_R$ and $Y = Y_L \cup Y_R$ where $X_L = X \cap L$, $X_R = X \cap R$, $Y_L = Y \cap L$ and $Y_R = Y \cap R$. Since (X, Y) is a biclique, note that $X_R \subseteq X'_R$ and $Y_R \subseteq Y'_R$ where $X'_R = \{v \in R : v \text{ is adjacent to every vertex of } Y_L\}$ and $Y'_R = \{v \in R : v \text{ is adjacent to every vertex of } X_L\}$. Moreover $|X_R| = k_1 - |X_L|$ and $|Y_R| = k_2 - |Y_L|$.

Thus, assuming that X_L and Y_L are given, by definition of R-biclique it is sufficient to know whether $\textsf{R-biclique}[X'_R, k_1 - |X_L|, Y'_R, k_2 - |Y_L|]$ is true. Since the Computing step goes through all possible choices for X_L and Y_L, it remains to show that the formula of the Preprocessing step is correct. Clearly, the base cases are correct. Let us consider the inductive step. The value $\textsf{R-biclique}[X, i, Y, j]$ is true iff there exists a vertex $v \in X$ (the same argument can be used if v is a vertex of Y) such that $\textsf{R-biclique}[X \setminus \{v\}, i, Y, j]$ is true (i.e. v does not belong to the (i, j)-biclique and thus it is removed from X) or R-biclique$[X \setminus \{v\}, i-1, N(v) \cap Y, j]$ is true (i.e. v is a vertex of the (i, j)-biclique and thus it remains to find a $(i-1, j)$-biclique from $X \setminus \{v\}$ and $\{u \in Y \setminus \{v\} : \{u, v\} \in E\}$).

3.3 Analysis of the running-time

The Partitioning step can clearly be done in polynomial time. During the Preprocessing step we need to compute R-biclique$[X, i, Y, j]$ for any (not necessarily disjoint) subsets $X, Y \subseteq R$ and any integers i and j, $0 \leq i, j \leq |R|$. For each such 4-tuple, R-biclique can be evaluated in polynomial time: go through all vertices of $X \cup Y$ and use the previously computed values of R-biclique – recall that X and Y are considered by increasing cardinality. Thus, to enumerate all X and Y, the algorithm needs $O^*(4^{\alpha n})$ time. The Computing step needs to consider all disjoint subsets $X_L, Y_L \subseteq L$ and then to look for already computed values of R-biclique. Consequently, it needs $O^*(3^{(1-\alpha)n})$ time. Finally the value of α is choosen to balance the running-time of the last two steps. By setting $\alpha = \log(3)/(2 + \log(3)) \approx 0.44211...$, our main theorem follows.

Theorem 3.1 *The described algorithm solves the* **Non-Induced $(\mathbf{k_1}, \mathbf{k_2})$ Biclique** *problem in time $O(1.8458^n)$ and exponential space.*

References

[1] J. Amilhastre, M.C. Vilarem, P. Janssen, Complexity of minimum biclique cover and minimum biclique decomposition for bipartite dominofree graphs. Discrete Appl. Math. 86, pp. 125–144 (1998).

[2] M. Dawande, J. Swaminathan, P. Keskinocak, S. Tayur, On bipartite and multipartite clique problems. J. Algorithms 41, pp. 388–403 (2001).

[3] M.R. Garey, D.S. Johnson, *Computers and Intractability: A guide to the Theory of NP-completeness.* Freeman, New York, 1979.

[4] S. Gaspers, D. Kratsch, M. Liedloff, On Independent Sets and Bicliques in Graphs, Proceedings of WG 2008, Springer, LNCS 5344, pp. 171-182.

[5] D.S. Hochbaum, Approximating clique and biclique problems, J. Algorithms 29, pp. 174–200 (1998).

[6] R. Peeters, The maximum edge biclique problem is NP-complete, Discrete Appl. Math. 131, pp. 651–654 (2003).

[7] M. Yannakakis, Node and edge deletion NP-complete problems, Proceedings of STOC 78, ACM, pp. 253-264 (1978).