Information Processing Letters 111 (2010) 64–67

Contents lists available at ScienceDirect

Information Processing Letters

www.elsevier.com/locate/ipl

Exact exponential-time algorithms for finding bicliques

Daniel Binkele-Raible [a], Henning Fernau [a], Serge Gaspers [b,1], Mathieu Liedloff [c,*]

[a] Universität Trier, FB 4 - Abteilung Informatik, D-54286 Trier, Germany
[b] Institute of Information Systems, Vienna University of Technology, Vienna, Austria
[c] LIFO, Université d'Orléans, 45067 Orléans Cedex 2, France

ARTICLE INFO

Article history:
Received 24 August 2010
Received in revised form 27 October 2010
Accepted 27 October 2010
Available online 29 October 2010
Communicated by Ł. Kowalik

Keywords:
Graph algorithms
Exact exponential-time algorithms
NP-hard problem
Complete bipartite subgraphs

ABSTRACT

Due to a large number of applications, bicliques of graphs have been widely considered in the literature. This paper focuses on non-induced bicliques. Given a graph $G = (V, E)$ on n vertices, a pair (X, Y), with $X, Y \subseteq V$, $X \cap Y = \emptyset$, is a non-induced biclique if $\{x, y\} \in E$ for any $x \in X$ and $y \in Y$. The NP-complete problem of finding a non-induced (k_1, k_2)-biclique asks to decide whether G contains a non-induced biclique (X, Y) such that $|X| = k_1$ and $|Y| = k_2$. In this paper, we design a polynomial-space $\mathcal{O}(1.6914^n)$-time algorithm for this problem. It is based on an algorithm for bipartite graphs that runs in time $\mathcal{O}(1.30052^n)$. In deriving this algorithm, we also exhibit a relation to the spare allocation problem known from memory chip fabrication. As a byproduct, we show that the constraint bipartite vertex cover problem can be solved in time $\mathcal{O}(1.30052^n)$.

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 an induced biclique of G. Notice that if G is a bipartite graph, then every non-induced biclique of G is also an induced one. Let the pair (X, Y) be an induced or non-induced biclique of G. Then 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 general or bipartite graphs) in various different areas such as data mining, automata and language theory, artificial intelligence and biology; see for example [1]. Therefore, bicliques and algorithmic problems about bicliques have been studied extensively.

1.1. Known results

Already in [9], the complexity of finding certain bicliques has been considered. For example, deciding whether a bipartite graph has a (k, k) biclique, also known as a balanced biclique of size (at least) k, is NP-complete ([GT24] in [9]). A maximum cardinality induced biclique can be computed in polynomial time on bipartite graphs [4], whereas this problem is NP-complete for general graphs [18]. A related problem that asks to compute a non-induced biclique with a maximum number of edges is also known to be NP-hard [16]. Studies on the approximability of these problems are presented in [13].

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

* Corresponding author.
E-mail addresses: raible@informatik.uni-trier.de (D. Binkele-Raible), fernau@uni-trier.de (H. Fernau), gaspers@kr.tuwien.ac.at (S. Gaspers), liedloff@univ-orleans.fr (M. Liedloff).

1 Serge Gaspers acknowledges partial support from Conicyt Chile via the project Basal-CMM.

0020-0190/$ – see front matter
doi:10.1016/j.ipl.2010.10.020

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

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

Bipartite $(\mathbf{k_1}, \mathbf{k_2})$ **Biclique**
Input: An undirected bipartite graph G, positive integers k_1 and k_2.
Question: Does G have a (k_1, k_2) biclique?

Observe that any biclique in a bipartite graph is induced.

There is a trivial $\mathcal{O}^*(3^n)$ algorithm for finding and also for enumerating all induced and non-induced (k_1, k_2) bicliques of a graph, respectively.[2] It considers all partitions of the vertex set into X, Y and $V \setminus (X \cup Y)$ and verifies for each whether (X, Y) fulfills all conditions.

1.2. Our results

For enumerating all non-induced (k_1, k_2) bicliques, note that there is no hope in obtaining a faster algorithm than the above-described $\mathcal{O}^*(3^n)$ algorithm, as a complete graph on n vertices has $\Theta^*(3^n)$ 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 $\mathcal{O}(1.6914^n)$ time algorithm, based on a polynomial-space $\mathcal{O}(1.30052^n)$ time algorithm for **Bipartite** $(\mathbf{k_1}, \mathbf{k_2})$ **Biclique**. That algorithm in turn employs a relation to a specific application, namely to **Spare Allocation**, which is inspired by memory chip fabrication. This relation may be interesting on its own. As a byproduct, we also show that **Constraint Bipartite Vertex Cover** can be solved in time $\mathcal{O}(1.30052^n)$.

Observe that there is also an $\mathcal{O}^*(3^{n/3}) = \mathcal{O}(1.4423^n)$ 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 [6] and on the fact that an n-vertex graph has $\mathcal{O}^*(3^{n/3})$ maximal induced bicliques [11,12].

In this note we improve algorithms presented in [7], where the following results were presented: (1) a polynomial-space $\mathcal{O}(1.8899^n)$ time algorithm and (2) an exponential-space $\mathcal{O}(1.8458^n)$ time algorithm for solving the **Non-Induced** $(\mathbf{k_1}, \mathbf{k_2})$ **Biclique** problem. Our new results make use of connections to a problem called **Constraint Bipartite Vertex Cover** for which a sophisticated branching algorithm was described in [2,3,8] that has been analyzed from the viewpoint of parameterized complexity.

2. Finding bicliques in bipartite graphs

As we will exhibit, the **Bipartite** $(\mathbf{k_1}, \mathbf{k_2})$ **Biclique** problem is closely related to the following problem that comes up (with certain variants till today) in the fabrication process of memory elements.

[2] Throughout the paper we write $f(n) = \mathcal{O}^*(g(n))$ if $f(n) \leqslant p(n) \cdot g(n)$ for some polynomial $p(n)$.

An instance of **Spare Allocation (SAP)** is given by a $n_1 \times n_2$ binary matrix A representing an erroneous chip with $A[r, c] = 1$ if and only if the chip is faulty at position $[r, c]$, and the parameters: positive integers k_1 and k_2. The task is: Is there a *reconfiguration strategy* that repairs all faults and uses at most k_1 spare rows and at most k_2 spare columns?

With reconfiguration strategy we mean a prescription which rows and columns from A have to be replaced by spares. Kuo and Fuchs [14] provide a fundamental study of that problem. A review on the according literature is given in [2,3]. Put concisely, the “most widely used approach to reconfigurable VLSI” uses spare rows and columns to tolerate failures in rectangular arrays of identical computational elements, which may be as simple as memory cells or as complex as processor units. If a faulty cell is detected, the entire row or column is replaced by a spare one.

The following graph-theoretic problem can easily be seen to be equivalent to the previous problem via the adjacency matrix of a bipartite graph:

An instance of **Constraint Bipartite Vertex Cover (CBVC)** is given by a bipartite graph $G = (V_1, V_2, E)$, and the parameters: positive integers k_1 and k_2. The task is: Is there a *vertex cover* $C \subseteq V_1 \cup V_2$ with $|C \cap V_i| \leqslant k_i$ for $i = 1, 2$?

It is known [8] that CBVC admits a quadratic kernel and a search tree algorithm with running time $1.3999^k n^{\mathcal{O}(1)}$ using polynomial space, where $k = k_1 + k_2$.

Let us call a valid solution of a CBVC instance a (k_1, k_2) *vertex cover*.

Parameterized duality is usually defined by reparameterizing, say a vertex-selection problem on graphs, by considering $n - k$ instead of k as the parameter, where n is the number of vertices in the graph and k is the solution size. For example, thanks to Gallai’s identity, the parameterized dual of **Vertex Cover** is **Independent Set**. For vertex-selection problems on bipartite graphs, where we face two parameters k_1, k_2 corresponding to the number of vertices in each part of the vertex bipartition (V_1, V_2) that are in the solution, it is natural to consider the reparameterization given by $(n_1 - k_1, n_2 - k_2)$, where $n_i = |V_i|$, as the dual parameterization. Considering $k = k_1 + k_2$ and $n = n_1 + n_2$, one can see that this definition corresponds to the widely used notion of parameterized duality for one-parametric problems.

The *bipartite complement* of a bipartite graph $G = (V_1, V_2, E)$ is the bipartite graph $G^C = (V_1, V_2, E^C)$, where E^C contains all edges between V_1-vertices and V_2-vertices that are not contained in E.

The following lemma formalizes that **Spare Allocation** can be solved in $c^{k_1+k_2} n^{\mathcal{O}(1)}$ time if and only if the parameterized dual of **Bipartite** $(\mathbf{k_1}, \mathbf{k_2})$ **Biclique** can be solved in $c^{k_1+k_2} n^{\mathcal{O}(1)}$ time.

Lemma 1. *The* **Spare Allocation** *problem is polynomially equivalent to the parameterized dual of* **Bipartite** $(\mathbf{k_1}, \mathbf{k_2})$ **Biclique**.

Proof. As noticed above, we can consider a CBVC instance to start with, that is, a bipartite graph $G = (V_1, V_2, E)$, together with parameters k_1, k_2. The parameterized dual asks to find an independent set $I_1 \cup I_2$ in G with $I_i \subseteq V_i$ and

$|I_i| \geqslant k'_i = |V_i| - k_i$. Now, $C_1 \cup C_2$ is a (k_1, k_2) vertex cover of G if and only if $(V_1 \setminus C_1) \cup (V_2 \setminus C_2)$ is an independent set in G with $|V_1| - k_1$ vertices in V_1 and $|V_2| - k_2$ vertices in V_2 if and only if $(V_1 \setminus C_1) \cup (V_2 \setminus C_2)$ is a $(|V_1| - k_1, |V_2| - k_2)$ biclique of the bipartite complement of G. □

Clearly, the "trivial barrier" for moderately exponential-time algorithms for CBVC is $\mathcal{O}^*(2^{n/2})$ rather than $\mathcal{O}^*(2^n)$, since it is enough to consider all subsets of the smaller set of V_1 and V_2. Our result is the first exact algorithm for **Constraint Bipartite Vertex Cover** that breaks the trivial $\Theta^*(2^{n/2})$-barrier.

Theorem 2. Constraint Bipartite Vertex Cover *and* **Bipartite $(\mathbf{k_1}, \mathbf{k_2})$ Biclique** *can be solved in time $\mathcal{O}(1.30052^n)$, using polynomial space.*

Proof. Let $G = (V_1, V_2, E)$ be an instance for **Bipartite $(\mathbf{k_1}, \mathbf{k_2})$ Biclique**. Let $n_1 = |V_1|$, $n_2 = |V_2|$, and $\alpha := 0.2189$. We will consider two algorithmic possibilities depending on $k_1 + k_2$:

1. First, suppose $k_1 + k_2 \geqslant \alpha n$. According to Lemma 1 the problem of finding a (k_1, k_2) biclique is equivalent to finding a (k'_1, k'_2) vertex cover in the bipartite complement with $k'_1 = n_1 - k_1$ and $k'_2 = n_2 - k_2$. The algorithm in [8] solves CBVC in time $\mathcal{O}(1.3999^{k'_1+k'_2}) = \mathcal{O}(1.3999^{n-(k_1+k_2)})$ using polynomial space. As $k_1 + k_2 \geqslant \alpha n$ the running time is $\mathcal{O}(1.3999^{(1-\alpha)n}) = \mathcal{O}(1.30052^n)$.
2. Now, suppose $k_1 + k_2 < \alpha n$. Depending on the values of k_1, k_2, n_1, and n_2, the algorithm either (i) enumerates all subsets $V'_1 \subseteq V_1$ of size k_1 and checks whether $|\bigcap_{v \in V'_1} N(v)| \geqslant k_2$ holds, or (ii) it enumerates all subsets $V'_2 \subseteq V_2$ of size k_2 and checks whether $|\bigcap_{v \in V'_2} N(v)| \geqslant k_1$ holds. If this is the case we clearly have found a (k_1, k_2) biclique. The enumeration (i) is done if and only if $\binom{n_1}{k_1} \leqslant \binom{n_2}{k_2}$ (otherwise enumeration (ii) is performed). This step can thus be done in time $\mathcal{O}^*(\min\{\binom{n_1}{k_1}; \binom{n_2}{k_2}\})$. By Vandermonde's identity, we have that for any $x, y, z \in \mathbb{N}$, $\binom{x+y}{z} = \sum_{i=0}^{z} \binom{x}{i}\binom{y}{z-i}$. In particular, $\binom{n}{k} = \binom{n_1+n_2}{k_1+k_2} \geqslant \binom{n_1}{k_1} \cdot \binom{n_2}{k_2}$. Thus, $\min\{\binom{n_1}{k_1}; \binom{n_2}{k_2}\} \leqslant \sqrt{\binom{n}{k}}$ and the running time of this step is upper bounded by $\mathcal{O}^*(\binom{n}{\alpha n}^{1/2})$ which, by Stirling formula [17], is upper bounded by $\mathcal{O}(1.30052^n)$.

This shows that **Bipartite $(\mathbf{k_1}, \mathbf{k_2})$ Biclique** can be solved in $\mathcal{O}(1.30052^n)$ time. By Lemma 1, **Constraint Bipartite Vertex Cover** can also be solved in $\mathcal{O}(1.30052^n)$ time. □

It is tempting to replace the trivial brute-force enumeration in the second part of the algorithm presented in the preceding proof by a more elaborate technique. Notice that this might be not so easy, since it would probably lead to a parameterized algorithm for the **Non-Induced $(\mathbf{k_1}, \mathbf{k_2})$ Biclique** problem, hence solving a long-standing open problem in parameterized complexity.

Corollary 3. Non-Induced $(\mathbf{k_1}, \mathbf{k_2})$ Biclique *can be solved in time $\mathcal{O}(1.6914^n)$, using polynomial space.*

Proof. Given a graph $G = (V, E)$, construct a bipartite graph $G' = (V, V', E')$ where V' is a copy of V. To each vertex $v \in V$, its copy in V' is denoted by v'. Then $\{u, v'\} \in E'$ iff $\{u, v\} \in E$. Observe that (X, Y) is a non-induced (k_1, k_2) biclique in G if and only if (X, Y') is a bipartite (k_1, k_2) biclique in G' where Y' is the copy of Y in V'. Now apply the preceding theorem to obtain a running time of $\mathcal{O}(1.30052^{2n}) = \mathcal{O}(1.6914^n)$. □

3. Conclusions

We already mentioned that CBVC, the parameterized dual of **Non-Induced $(\mathbf{k_1}, \mathbf{k_2})$ Biclique**, is in FPT. It is a natural question to ask whether **Non-Induced $(\mathbf{k_1}, \mathbf{k_2})$ Biclique** is in FPT as well. The proof of Kuo and Fuchs [14] (showing the NP-hardness of CBVC) is not parameter-preserving and hence does not answer this question. The only thing that can be (relatively easily) seen is membership in W[1]. As already mentioned in [5,10,15], this poses an interesting open problem in parameterized complexity, even when restricted to bipartite graphs as in our sketched application and even when $k_1 = k_2$. We finally mention that the experiments with an implementation of CBVC described in [2,3] show that the approach described in this paper might be feasible for many practical situations.

References

[1] J. Amilhastre, M.-C. Vilarem, P. Janssen, Complexity of minimum biclique cover and minimum biclique decomposition for bipartite domino-free graphs, Discrete Applied Mathematics 86 (1998) 125–144.

[2] G. Bai, H. Fernau, Constraint bipartite vertex cover: Simpler exact algorithms and implementations, in: F.P. Preparata, X. Wu, J. Yin (Eds.), Frontiers in Algorithmics FAW, in: LNCS, vol. 5059, Springer, 2008, pp. 67–78.

[3] G. Bai, H. Fernau, Constraint bipartite vertex cover: Simpler exact algorithms and implementations, Journal of Combinatorial Optimization, in press, doi:10.1007/s10878-010-9289-7.

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

[5] E. Demaine, G.Z. Gutin, D. Marx, U. Stege, 07281 Open problems, in: Structure Theory and FPT Algorithmics for Graphs, Digraphs and Hypergraphs, in: Dagstuhl Seminar Proceedings, no. 07281, Dagstuhl, Germany, 2007 IBFI.

[6] V.M. Dias, C.M. de Figueiredo, J.L. Szwarcfiter, Generating bicliques of a graph in lexicographic order, Theoretical Computer Science 337 (1–3) (2005) 240–248.

[7] H. Fernau, S. Gaspers, D. Kratsch, M. Liedloff, D. Raible, Exact exponential-time algorithms for finding bicliques in a graph, in: S. Cafieri, A. Mucherino, G. Nannicini, F. Tarissan, L. Liberti (Eds.), CTW, 2009, pp. 205–209.

[8] H. Fernau, R. Niedermeier, An efficient exact algorithm for constraint bipartite vertex cover, Journal of Algorithms 38 (2) (2001) 374–410.

[9] M.R. Garey, D.S. Johnson, Computers and Intractability: A Guide to the Theory of NP-Completeness, Freeman, 1979.

[10] S. Gaspers, Exponential time algorithms: Structures, measures, and bounds, PhD thesis, Department of Informatics, University of Bergen, 2008.

[11] S. Gaspers, D. Kratsch, M. Liedloff, On independent sets and bicliques in graphs, in: H. Broersma, T. Erlebach, T. Friedetzky, D. Paulusma (Eds.), Graph-Theoretic Concepts in Computer Science WG, in: LNCS, vol. 5344, 2008, pp. 171–182.

[12] S. Gaspers, D. Kratsch, M. Liedloff, On independent sets and bicliques in graphs, Algorithmica, in press, doi:10.1007/s00453-010-9474-1.
[13] D. Hochbaum, Approximating clique and biclique problems, Journal of Algorithms 29 (1998) 174–200.
[14] S.-Y. Kuo, W.K. Fuchs, Efficient spare allocation for reconfigurable arrays, IEEE Design and Test 4 (1987) 24–31.
[15] D. Lokshtanov, New methods in parameterized algorithms and complexity, PhD thesis, Department of Informatics, University of Bergen, 2009.
[16] R. Peeters, The maximum edge biclique problem is NP-complete, Discrete Applied Mathematics 131 (2003) 651–654.
[17] J. Stirling, Methodus differentialis, sive tractatus de summatione et interpolatione serierum infinitarum, London, 1730.
[18] M. Yannakakis, Node and edge-deletion NP-complete problems, in: Proc. 10th Annual ACM Symposium on Theory of Computing STOC, 1978, pp. 253–264.