Journal of Discrete Algorithms 7 (2009) 464–468

Contents lists available at ScienceDirect

Journal of Discrete Algorithms

www.elsevier.com/locate/jda

An exact algorithm for subgraph homeomorphism

Andrzej Lingas, Martin Wahlen [*]

Department of Computer Science, Lund University, Box 118, S-22100 Lund, Sweden

A R T I C L E I N F O

Article history:
Received 29 October 2007
Received in revised form 31 March 2008
Accepted 10 October 2008
Available online 28 October 2008

Keywords:
Subgraph homeomorphism
Disjoint paths
Time complexity
Space complexity

A B S T R A C T

The *subgraph homeomorphism* problem is to decide if there is an injective mapping of the vertices of a pattern graph into vertices of a host graph so that the edges of the pattern graph can be mapped into (internally) vertex-disjoint paths in the host graph. The restriction of subgraph homeomorphism where an injective mapping of the vertices of the pattern graph into vertices of the host graph is already given in the input instance is termed *fixed-vertex subgraph homeomorphism*.

We show that fixed-vertex subgraph homeomorphism for a pattern graph on p vertices and a host graph on n vertices can be solved in time $2^{n-p}n^{O(1)}$ or in time $3^{n-p}n^{O(1)}$ and polynomial space. In effect, we obtain new non-trivial upper bounds on the time complexity of the problem of finding k vertex-disjoint paths and general subgraph homeomorphism.

1. Introduction

Regarded as an injective mapping, the *subgraph isomorphism* of a pattern graph P into a host graph H consists of a mapping of vertices of P into vertices of H so that edges of P map to corresponding edges of H. Generalizations of this mapping include *subgraph homeomorphism*, also termed as *topological embedding* or *topological containment*, where vertices of P map to vertices of H and edges of P map to (internally) vertex-disjoint paths in H, and *minor containment*, where vertices of P map to disjoint connected subgraphs of H and edges of P map to edges of H.

All these problems are inherently NP-complete when the pattern graph P and the host graph H are not fixed [9]. For fixed P, all are solvable in polynomial time, which in case of subgraph homeomorphism and minor containment is highly non-trivial to show [16]. They remain NP-complete for several special graph classes, e.g., for graphs of bounded treewidth [10,15]. Restricting the pattern graph P to complete graphs or simple cycles or paths does not help in the case of subgraph isomorphism: the maximum clique, Hamiltonian cycle and Hamiltonian path problems are well known as basic NP-complete problems [9].

There is an extensive literature on the exact complexity of the maximum clique problem (or, equivalently the maximum independent set problem) and the Hamiltonian cycle or path problem. At present, the best known upper-time bounds are respectively $2^{0.288n}n^{O(1)}$ [7] and $2^n n^{O(1)}$ [12], where n is the number of vertices of the host graph (see also [2,14] for the recent upper time-bounds for the related problem of graph coloring). For the general subgraph isomorphism problem the known upper bound is of the form $O(n^{\lceil p/3\rceil\omega+(p \bmod 3)})$ where p is the number of vertices of the pattern graph and ω is the exponent of the fastest matrix multiplication algorithm (cf. [5]).[1]

For general subgraph homeomorphism and general minor containment, the authors are not familiar with any non-trivial upper bounds on the time complexity of these problems. For example, Downey et al. mention on page 149 in [6], that

* Corresponding author.
E-mail addresses: Andrzej.Lingas@cs.lth.se (A. Lingas), martin@cs.lth.se (M. Wahlen).

[1] This upper bound can be marginally improved by using the rectangular matrix multiplication instead of the square one.

1570-8667/$ – see front matter © 2008 Elsevier B.V. All rights reserved.
doi:10.1016/j.jda.2008.10.003

subgraph homeomorphism (termed as topological containment) can be solved in time $n^{O(m)}$ where n is the number of vertices and m is the number of edges of the host graph. Note that if the host graph is dense then $m = \Omega(n^2)$. A better upper time bound, namely $n^{n+O(1)}$, can be deduced from the result of Gupta et al. [11] stating that subgraph homeomorphism for graphs of pathwidth bounded by k can be solved in time $n^{k+O(1)}$.

A natural simplification of subgraph homeomorphism is *fixed-vertex subgraph homeomorphism*: Given an injective map f from the vertices of the pattern graph to those of the host graph H, decide whether or not H contains a homeomorphic image of P in which each vertex of P is identified with its image under f.

Note that fixed-vertex subgraph homeomorphism includes as a sub-problem the well known *k vertex-disjoint paths* problem (see [9] for disjoint connecting paths): Given a graph G and k disjoint pairs (s_i, t_i) of its vertices, decide whether or not G contains k vertex-disjoint paths, one connecting each pair (s_i, t_i).

Similarly, fixed-vertex subgraph homeomorphism and k vertex-disjoint paths problems are solvable for fixed pattern graph or fixed k, respectively, in polynomial time [16], and no non-trivial upper bounds on their time complexity seem to be known in the literature.

In this paper, we show that fixed-vertex subgraph homeomorphism for a pattern graph on p vertices and a host graph on n vertices can be solved in time $2^{n-p}n^{O(1)}$ or in time $\tilde{O}(3^{n-p}n^7)$ and polynomial space; here and henceforth we use the notation $\tilde{O}(f(n))$ to denote $O(f(n)\log^d(n))$ for some constant d. Consequently, the k vertex-disjoint paths problem can be solved within the same asymptotic bounds. It follows that in the general case, where the pattern graph is not assumed to be fixed, subgraph homeomorphism can be solved in time $\binom{n}{p}p!2^{n-p}n^{O(1)}$ or in time $\tilde{O}(\binom{n}{p}p!3^{n-p}n^7)$ and polynomial space.

Our algorithm for fixed-subgraph homeomorphism is based on the use of the principle of exclusion–inclusion to count the number of feasible solutions. This method was originally applied by Karp in [13] (rediscovered in [1]) in order to count the number of Hamiltonian cycles using the concept of walks avoiding a subset of the vertex set. We rely on and introduce a generalization of the latter concept to include sets of avoiding walks. We also use the so called fast *zeta transform* (cf. Björklund et al. [3] or [4]) to reduce the term 3^{n-p} to 2^{n-p} in our upper time-bounds. In consequence, our upper time-bounds for fixed-vertex subgraph homeomorphism, and consequently, for subgraph homeomorphism, are primarily obtained for the more general problem of counting the number of different ways of implementing such subgraph homeomorphisms (two ways are different if they involve different sets of paths modeling the edges of the pattern graph).

In the next section, we introduce the concepts of subset avoiding walks and subset avoiding sets of walks. In Section 3, we present our algorithm for fixed-vertex subgraph homeomorphism. In the following section, we refine the algorithm by a reduction to the zeta transform on the subset lattice. In Section 5, we derive the upper time-bound for general subgraph homeomorphism.

2. Walks and sets of walks

Let $G = (V, E)$ be an undirected graph on n vertices. Define a walk in G as an alternating sequence of vertices and edges $x_0, e_1, x_1, \dots, e_l, x_l$ where $e_i = (x_{i-1}, x_i)$, the length of a walk is the number of edges in the sequence. A walk avoids a set of vertices S if $x_0, \dots, x_l \notin S$. For a subset $S \subseteq V$, $m \in \{1, \dots, n-1\}$, $(u, v) \in V$ let $WALK^m_{u,v}(S)$ be the set of all walks that start in vertex u, end in vertex v, avoid all vertices in S and have length m.

For a given subset $S \subseteq V$ and $m > 1$, we can compute the cardinalities $|WALK^m_{u,v}(S)|$ from the cardinalities $|WALK^{m-1}_{u',v'}(S)|$ by the recurrence $|WALK^m_{u,v}(S)| = \sum_{(v,v'')\in E} |WALK^{m-1}_{u,v''}(S)|$. The computation involves the edges of G with both endpoints outside S and requires $O(n^3)$ additions of $O(n \log n)$ bit numbers. Thus, it takes time $O(n^4 \log n)$, and space $O(n^3 \log n)$. Hence, we have the following lemma.

Lemma 1. *For a given subset $S \subseteq V$, all $m \in \{1, \dots, n-1\}$ and all $u, v \in V$, we can compute the cardinalities $|WALK^m_{u,v}(S)|$ in time $\tilde{O}(n^5)$ and space $\tilde{O}(n^4)$.*

We will now introduce the notion of a set of walks avoiding a subset of vertices and describe how to compute the cardinality of a set of walks. Let U be a set of vertex pairs (u, v) where $u, v \in V$, $m = n - p + |U|$ where p is the number of different vertices in the pairs in U, and a subset $S \subseteq V$. We define $SETWALK^m_U(S)$ as the family of all sets of walks T such that

- for each $(s, t) \in U$, T contains exactly one walk between s and t avoiding the vertices in S,
- the sum of the lengths of the walks in T is exactly m.

Given the cardinalities $|WALK^{m'}_u(S)|$ for $u \in U$ and $m' \in \{1, \dots, n-1\}$, we can compute the cardinality $|SETWALK^m_U(S)|$ by straightforward dynamic programming. To begin with, we label the vertex pairs of U by $e_1, \dots e_\ell$ where ℓ is the cardinality of U. Next we construct a sequence of tables A_k, $k = 1 \dots |U|$, each with m entries. For the vertex pair e_1, we set $A_1(j) = |WALK^j_{e_1}(S)|$. Given the table A_{k-1} we compute the table A_k by letting

$$A_k(j) = \sum_{i=1}^{j-1} A_{k-1}(i)\left|WALK^{j-i}_{e_k}(S)\right|.$$

Note that $A_\ell(m) = |SETWALK_U^m(S)|$. The algorithm uses $O((n+|U|)^2|U|)$ multiplications of $O((n+|U|)\log n)$ bit numbers. Thus, if we use the multiplication algorithm due to Fürer [8] improving on Schönhage and Strassen [17], the dynamic programming takes $O(n(n+|U|)^2|U|\log^2 n \log\log n 2^{\log^* n})$ time, using $O(|U|(n+|U|)n\log n + (n+|U|)(n+|U|)\log n)$ space. Hence, by Lemma 1, we obtain the following lemma.

Lemma 2. *For a set of vertex pairs U, $m = n - p + |U|$ where p is the number of different vertices in the pairs in U, and a subset $S \subseteq V$, we can compute the cardinality $|SETWALK_U^m(S)|$ in time $\tilde{O}(n^7)$ and space $\tilde{O}(n^5)$.*

3. An exact algorithm for fixed-vertex homeomorphism

To solve the fixed-vertex homeomorphism problem for a pattern graph P on p vertices and a host graph $H = (V, E)$ on n vertices, let us consider all possible choices of the set I of l internal vertices on the paths interconnecting the p vertices in H in one-to-one correspondence with the edges of P ($\binom{n-p}{l}$ ways). We intend to compute, for each I, the number of ways to connect the p vertices in H using disjoint paths whose internal vertices are those in I.

For the given p vertices with their assignments to the vertices of P, and the subset I, define the graph $G = (W, F)$ as the subgraph of H induced by the union of the p vertices with I. Note that $|W| = l + p$. Let U be the set of pairs of the p vertices in one-to-one correspondence with the edges of the pattern graph P.

For $m_l = |W| - p + |U|$, i.e., $m_l = l + |U|$, consider the family of sets of walks $SETWALK_U^{m_l}(S)$ for G according to the definition from the previous section.

The next lemma follows from the inclusion–exclusion principle.

Lemma 3. *The number of sets of $|U|$ (internally) vertex-disjoint paths interconnecting the p vertices in G in one-to-one correspondence with the edges of P and covering all vertices in G is $\sum_{S\subseteq I}(-1)^{|S|}|SETWALK_U^{m_l}(S)|$.*

Proof. Note that $SETWALK_U^{m_l}(\emptyset)$ contains the sets of walks that correspond to the desired sets of paths. From $|SETWALK_U^{m_l}(\emptyset)|$ we need to remove the number of sets of walks that do not correspond to the desired sets of paths, that is $|\bigcup_{i\in I} SETWALK_U^{m_l}(\{i\})|$. By the inclusion–exclusion principle

$$\left|SETWALK_U^{m_l}(\emptyset)\right| - \left|\bigcup_{i\in I} SETWALK_U^{m_l}(\{i\})\right| = \sum_{S\subseteq I}(-1)^{|S|}\left|SETWALK_U^{m_l}(S)\right|. \qquad \square$$

By combing Lemmas 2 and 3, we can compute the number of sets of (internally) vertex-disjoint paths interconnecting the p vertices in H in one-to-one correspondence with the edges of P and covering all vertices in G in time $\tilde{O}(2^l n^7)$ and space $\tilde{O}(n^5)$.

By summing up the counts for each I, we can solve the fixed subgraph homeomorphism problem for P and H in time $\tilde{O}(\sum_l^{n-p}\binom{n-p}{l}2^l n^7)$, i.e., in time $\tilde{O}(3^{n-p}n^7)$, and space $\tilde{O}(n^5)$.

Theorem 4. *The fixed-vertex homeomorphism problem for a pattern graph on p vertices and a host graph on n vertices can be solved in time $\tilde{O}(3^{n-p}n^7)$ and space $\tilde{O}(n^5)$.*

Corollary 5. *The k vertex-disjoint paths problem in a graph on n vertices can be solved in time $\tilde{O}(3^{n-2k}n^7)$, and space $\tilde{O}(n^5)$.*

4. A faster algorithm for fixed-vertex homeomorphism

By applying the upper time-bound on the fast zeta transform[2] using Yates method [18] (see Björklund et al. [3]), we can obtain a substantially better bound for fixed-vertex homeomorphism than that of Theorem 4.

To reduce the fixed subgraph homeomorphism problem to a collection of zeta transform problems observe that the value of $(-1)^{|S|}|SETWALK_U^{m_l}(S)|$ is really only a function of $I \setminus S$ (more precisely, of the subgraph of H induced by $T = I \setminus S$ and the p fixed vertices) and l, since $m_l = l + |U|$, U is fixed and the avoided vertices from S in G do not affect it. Given $l = |I|$ and $T = I \setminus S$ we note that $|S| = l - |T|$. Therefore, we may denote $(-1)^{|S|}|SETWALK_U^{m_l}(S)|$ by $f_l(T)$. Next let V' denote the set of all vertices in V different from the p fixed vertices. By Lemma 2, we may assume that the values of $f_l(T)$ are precomputed for all possible values of l, and all possible subsets T of V' in time $O(2^n n^8 \log^2 n \log\log n 2^{\log^* n})$. It follows by Lemma 3 that it remains to compute for each $I \subseteq V'$, the sum

$$h_{|I|}(I) = \sum_{T\subseteq I} f_{|I|}(T).$$

[2] Note that the fast zeta transform is sometimes called the fast Möbius transform, however we follow the terminology from [3] here.

Thus, in particular, if we compute the zeta transform $\{h_l(I) \mid I \subseteq V'\}$ given by

$$h_l(I) = \sum_{T \subseteq I} f_l(T)$$

for $l = 0, \ldots, n-p$, then we are done. As noted by [3], the fast zeta transform $\{h_l(I) \mid I \subseteq V'\}$ can be computed by using $O(|V|2^{|V|})$ additions with $O(|V|\log M)$-bit integers, where M is an upper bound on the absolute values of $f_l(T)$. Hence, since $|V'| = n-p$ and $|SETWALK_U^{m_l}(S)| \leqslant 2^{n^{O(1)}}$, we obtain the following improved upper time-bound.

Theorem 6. *The fixed-vertex homeomorphism problem for a pattern graph on p vertices and a host graph on n vertices can be solved in time* $2^{n-p}n^{O(1)}$.

Corollary 7. *The k vertex-disjoint paths problem in a graph on n vertices can be solved in time* $2^{n-2k}n^{O(1)}$.

5. Exact algorithms for subgraph homeomorphism

To reduce the subgraph homeomorphism problem for a pattern graph P on p vertices and a host graph H on n vertices, let us consider all possible choices of p vertices in H ($\binom{n}{p}$ ways) and all possible one-to-one assignments of these p vertices to the vertices of P ($p!$ ways). Hence, we obtain the following theorem by Theorems 4 and 6.

Theorem 8. *Subgraph homeomorphism can be solved in time* $\binom{n}{p}p!2^{n-p}n^{O(1)}$ *or in time* $\tilde{O}(\binom{n}{p}p!3^{n-p}n^7)$ *and space* $\tilde{O}(n^5)$.

Note that if the pattern graph has a easily computable automorphisms then we can replace the term $p!$ by $p!/a$.

Since $\binom{n}{p}p! \leqslant n^p$, our algorithms for subgraph homeomorphism run in exponential time for $p = O(n/\log n)$. On the other hand, since $\binom{n}{p}p! \geqslant n^{p/2}$ for sufficiently large n, they run in super-exponential time for $p = \omega(n/\log n)$. Of course, the known special cases of subgraph homeomorphism, e.g., the $O(1)$ vertex-disjoint path problem or the Hamiltonian cycle problem, typically admit more efficient specialized algorithmic solutions (see Introduction).

6. Final remarks

The reader familiar with recent developments on counting set partitions (cf. [3]) might observe the following. To solve the decision version of the fixed-vertex subgraph homeomorphism problem it is sufficient to partition the largest possible vertex set I (i.e., the set of vertices of the host graph different from the images of the p vertices of the pattern graph) into disjoint sets $I_1, \ldots, I_U$ in one-to-one correspondence with the edges of the pattern graph such that the vertices in I_i connect the images of the endpoints of the ith edge. To count such partitions, we can define functions f_1 through f_U on subsets of vertices such that $f_i(S)$ is 1 if the vertices in S connect the images of the endpoints of the ith edge in the pattern graph, and otherwise $f_i(S) = 0$. By [3], we can count the number of partitions $I_1, \ldots, I_U$ maximizing the sum $f_1(I_1) + f_2(I_2) + \cdots + f_U(I_U)$ in time $2^n n^{O(1)}$. This yields solution solely to the decision version of fixed-vertex subgraph homeomorphism since the aforementioned partitions are not necessarily in one-to-one correspondence with different ways of implementing the fixed-vertex subgraph homeomorphisms.

Recall that our upper time-bounds for fixed-vertex subgraph homeomorphism hold not only for the decision versions of these problems but also for counting different ways of implementing such subgraph homeomorphisms (two ways are different if they involve different sets of paths modeling the edges of the pattern graph).

In fact, the aforementioned alternative approach is not radically different from that used but for the lowest level where disjoint connecting sets are used instead of disjoint walks/paths. The walks/paths approach is more direct and efficient on the lowest level whereas the alternative approach is a more direct application of the technique from [3].

Given an algorithm for the decision version of fixed-vertex homeomorphism we can obtain an embedding in the host graph by proceeding as follows. Prune the host graph to a minimal subgraph satisfying the fixed subgraph homeomorphism test by repeatedly removing edges and isolated vertices. The pruning adds only a multiplicative polynomial factor to the aforementioned upper time-bounds.

It is an interesting question whether or not one could use the inclusion–exclusion principle to derive similar upper time-bounds for the minor containment problem. The underlying interconnection structure in case of minor containment is a set of trees while in case of subgraph homeomorphism just a set of simple paths. This difference can make counting solutions to the minor containment problem more difficult.

Acknowledgements

The authors are grateful to Fedor Fomin for inspiration, to Andreas Björklund and Thore Husfeldt for informing about their recent results, and to unknown referees for valuable comments including among other things the observation on the alternative set partition based method in Final remarks.

References

[1] E.T. Bax, Inclusion and exclusion algorithm for the Hamiltonian path problem, Information Processing Letters 47 (4) (1993) 203–207.
[2] A. Björklund, T. Husfeldt, Inclusion–exclusion algorithms for set partitions, in: Proc. 47th IEEE Symposium on Foundations of Computer Science, Berkeley 2006, pp. 575–582.
[3] A. Björklund, T. Husfeldt, M. Koivisto, Set partitioning via inclusion–exclusion, SIAM Journal on Computing, Special Issue for FOCS 2006, in press. Prelim. versions in Proc. 47th IEEE Symposium on Foundations of Computer Science, IEEE Computer Society, Los Alamitos, CA, 2006, pp. 575–582, 583–590.
[4] A. Björklund, T. Husfeldt, P. Kaski, M. Koivisto, Fourier meets Möbius: Fast subset convolution, in: Proc. of the 39th annual ACM Symposium on Theory of Computing, San Diego 2007, pp. 67–74.
[5] A. Czumaj, A. Lingas, Finding a heaviest triangle is not harder than matrix multiplication, in: Proc. of the 18th ACM–SIAM Symposium on Discrete Algorithms (SODA '07), 2007.
[6] R.G. Downey, M.R. Fellows, Parametrized Complexity, Springer, New York, 1999.
[7] F.V. Fomin, F. Grandoni, D. Kratch, Measure and conquer: A simple $2^{0.228n}$ independent set algorithm, in: Proceedings of the 17th ACM–SIAM Symposium on Discrete Algorithms (SODA '06), 2006.
[8] M. Fürer, Faster integer multiplication, in: Proc. of the 39th Annual ACM Symposium on Theory of Computing, San Diego 2007, pp. 57–66.
[9] M.R. Garey, D.S. Johnson, Computers and Intractability. A Guide to the Theory of NP-completeness, W.H. Freeman and Company, New York, 2003.
[10] A. Gupta, N. Nishimura, The complexity of subgraph isomorphism for classes of partial k-trees, Theoretical Computer Science 164 (1996) 287–298.
[11] A. Gupta, N. Nishimura, A. Proskurowski, P. Ragde, Embeddings of k-connected graphs of pathwidth k, Discrete Applied Mathematics 145 (2) (2005) 242–265.
[12] M. Held, R. Karp, A dynamic programming approach to sequencing problems, Journal of SIAM 10 (1) (1962) 196–210.
[13] R.M. Karp, Dynamic programming meets the principle of inclusion and exclusion, Operations Research Letters 1 (2) (1982) 49–51.
[14] M. Koivisto, An $O^*(2^n)$ algorithm for graph coloring and other partitioning problems via inclusion exclusion, in: Proc. 47th IEEE Symposium on Foundations of Computer Science, Berkeley 2006, pp. 582–590.
[15] J. Matoušek, R. Thomas, On the complexity of finding iso- and other morphisms for partial k-trees, Discrete Mathematics 108 (1992) 343–364.
[16] N. Robertson, P.D. Seymour, Graph minors. XIII. The disjoint paths problem, Journal of Combinatorial Theory B 63 (1995) 65–110.
[17] A. Schönhage, V. Strassen, Schnelle Multiplikation großer Zahlen, Computing 7 (1971) 281–292.
[18] F. Yates, The design and analysis of factorial experiments, Technical Communications no. 35 of the Commonwealth Bureau of Soils, 1937.