TAIWANESE JOURNAL OF MATHEMATICS
Vol. 7, No. 1, pp. 165-171, March 2003
This paper is available online at http://www.math.nthu.edu.tw/tjm/

A LAYER-DECOMPOSITION WITH DEGREE METHOD FOR SUBGRAPHS RECOGNITION PROBLEM

Yifan Xu

Abstract. For two planar undirected graphs G and H, the *subgraphs recognition problem* (SRP) is to find and list all subgraphs of G which are isomorphic to H. In this paper, we introduce the idea of layer-decomposition with degree and present an algorithm based on this idea for SRP. Since subgraphs isomorphic to each other contain their spanning trees with the same decomposition, the SRP can be decomposed into two subproblems: First, find subtrees of G which have the same layer-decomposition as that of H. Then, test whether the induced subgraphs generated by these subtrees are isomorphic to H. By this scheme, we greatly decrease the complexity to $O(n(\Delta-1)^{k-1}k^2)$, where Δ is the degree of G and n, k are the orders of G, H respectively.

1. Introduction

Let G be a planar undirected graph with order n and degree Δ and let H be another planar undirected graph with order k. The *subgraph recognition problem* (SRP) is to recognize and list all subgraphs of G which are isomorphic to H. This problem has a strong application background in VLSI circuit design and structure analysis. In practice, we often need to track some functional modules and determine their positions to investigate the structure of VLSI circuit. Since the VLSI circuit and its functional modules can be represented by a variety of specific graphs, the method for SRP can be taken as a tool for VLSI circuit structure analysis. Hence, finding a way to effectively solve SRP is an interesting and valuable task.

The trivial way to solve SRP requires $O(k^2n^k)$ time. However, in a practical VLSI circuit, n can reach one million or more and k may be great than

Received October 12, 2000; revised September 24, 2001.
Communicated by Frank Hwang.
2000 *Mathematics Subject Classification*: 68R10, 05C38.
Key words and phrases: Recognition, planar undirected graph, spanning tree, decomposition.

one hundred. Obviously it is beyond the ability of computer at present time. Hence, previous work on SRP has always concentrated on recognizing special families of subgraphs, such as short paths [4], cycles [2, 7, 8], cliques [5] and regular subgraphs [11].

There are two results we must mention here which belong to Plehn and Voigt [6] and Sundaram and Skiema [10] respectively. In 1990, Plehn and Voigt [6] used the tree-decomposition technique to show that the SRP can be determined in time $O(k^{3k}n^{w+1})$, where $w \geq 1$ is the tree-width of H. Several years later, in 1995, Sundaram and Skiena [10] introduced the concept of flower number and showed that if f is the flower number of graph H, then subgraphs of G isomorphic to H can be recognized in $O((k-f)^{3(k-f)}n^f m)$, where m is the edge number of G. According to [10], a flower set of a graph is a set of nodes whose removal reduces the graph to vertex disjoint paths. The flower number of a graph H is the cardinality of the smallest flower set of H. The common weakness of the two above methods is that the amount of computation is still great, especially for n pretty large and k not very small. So, to construct an algorithm such that it can deal with general SRP in a tolerable complexity is a key step to analyze the VLSI circuit.

In our previous work on this topic [12], we investigate SRP with a new idea, layer-decomposition. For a given planar graph, we select a spanning tree for the graph and take a vertex of the graph as the root of the spanning tree. We partition the vertices of the graph in the following way. Put all vertices of the same distance to the root of the spanning tree into one set. Denote the family of all such sets a layer-decomposition for the graph. The complexity of our algorithm in [12] is $O(n(\Delta-1)^{k-1}\alpha(H))$, where $\alpha(H)$ is a variant with respect to the structure of H. In some practical applications such as the VLSI circuit, since Δ and k are small, the advantage of our method is remarkable. However, in some special cases, $\alpha(H)$ could be large. This will result in low efficiency of the algorithm. In order to improve the algorithm, in this paper we pay attention to minimizing $\alpha(H)$. We introduce the idea of layer-decomposition with degree, i.e., attaching a degree to each vertices in the layer-decomposition and establish an algorithm for SRP. The computational complexity of the algorithm is $O(n(\Delta-1)^{k-1}k^2)$ which has nothing to do with $\alpha(H)$.

This paper is organized in this way. In section 2 we introduce the concept of layer-decomposition with degree. In section 3 we state the algorithm for SRP and in section 4 we analyze the computational complexity of the algorithm. Then in section 5 we make the numerical test. Last in section 6, we give the conclusion.

2. Layer-Decomposition with Degree

Let $h_0, h_1, \ldots, h_{k-1}$ be k vertices of graph H and T be a spanning tree of graph H. For each vertex $h_j, (0 \leq j \leq k-1)$, we denote by α_j the degree of vertex h_j in the spanning tree T. Take one vertex, i.e., h_0, as the root of tree T. We define $d_j, (0 \leq j \leq k-1)$, as the distance between vertex h_j and root h_0 in spanning tree T. After selecting the spanning tree T and its root h_0 we obtain a triple $(h_j, d_j, \alpha_j)_{(T,h_0)}$, or simply (h_j, d_j, α_j), for each vertex $h_j, (0 \leq j \leq k-1)$ of graph H.

Definition 1. *Given a spanning tree T of graph H with k vertices $\{h_0, h_1, \cdots, h_{k-1}\}$ and a root h_0. We define subset $L_l = \{(h_j, d_j, \alpha_j) : j \in I_l\}$, where $I_l = \{j : d_j = l, 0 \leq j \leq k-1\}$, as the l-layer of tree T at root h_0 and define $L_{(T,h_0)} = \{L_l\}$ the layer-decomposition with degree of T at root h_0, or simply layer-decomposition.*

If the number of layers in layer-decomposition $L_{(T,h_0)}$ is s and $|I_l|$, the number of vertices in the lth layer is k_l for any $0 \leq l \leq s$, then the *distribution* of $L_{(T,h_0)}$ is defined by

$$\{\{\alpha_j\}_{I_0}; \{\alpha_j\}_{I_1}; \cdots; \{\alpha_j\}_{I_s}\}. \tag{1}$$

Clearly, $|I_0| = k_0 = 1$ and $\max\{d_j : 0 \leq j \leq k-1\} = s$. About distribution (1), we have

$$\begin{aligned} k &= k_0 + k_1 + \ldots + k_s \\ &= 2 + \sum_{j \in I_0 \cup I_1 \cup \cdots \cup I_{s-1}} (\alpha_j - 1). \end{aligned} \tag{2}$$

We say two layer-decompositions are *same* if they have the same distribution, i.e., the number of layers, the number of vertices in each layer and degrees of vertices in each layer are all same.

3. Algorithm

In this section, we present the algorithm based on the layer-decomposition for SRP.

We know that two graphs which are isomorphic to each other certainly have their spanning trees isomorphic to each other, and furthermore, have the same layer-decompositions for these spanning trees. In other words, if in one graph there is no spanning tree isomorphic to one in another graph, or no tree having the same layer-decomposition with that of a spanning tree in the other graph, then these two graphs are not isomorphic. To our problem recognizing whether graph G has subgraphs which are isomorphic to graph H, we only

need to find those subtrees in G which are isomorphic to one spanning tree of H, or furthermore, only need to find those subtrees in G which have the same layer-decomposition with that of one spanning tree of H. Then, check whether the induced subgraphs generated by those subtrees are isomorphic to H. Based on this idea, we state our algorithm in follow.

Algorithm for SRP:

Given input planar graphs H, G with order k, n respectively.

1. *Choose a spanning tree T of H and a root h_0 of T. Make the layer-decomposition $L_{(T,h_0)}$ and obtain the distribution $\{\{\alpha_j\}_{I_0}; \{\alpha_j\}_{I_1}; \cdots; \{\alpha_j\}_{I_s}\}$.*

2. *Find subtrees in G with the same layer-decomposition $L_{(T,h_0)}$.*

3. *If there are some, test whether induced subgraphs generated by these subtrees are isomorphic to H.*

4. Complexity Analysis

Following theorem states the complexity of algorithm for SRP.

Theorem 1. *Let k, n be orders of planar undirected graphs H, G respectively and Δ be the degree of G. Then we can find all subgraphs of G which are isomorphic to H in time $O(n(\Delta-1)^{k-1}k^2)$.*

Proof. The proof is constructive. In the first step of Algorithm, select a spanning tree T of H and its root h_0 and make the layer-decomposition $L_{(T,h_0)}$. These can be done in time $O(k)$.

In step 3, the complexity of checking whether two planar graphs with order k are isomorphic is $O(k^2)$ by [3].

Now we consider the complexity in step 2 to find subtrees with layer-decomposition $L_{(T,h_0)}$ in G. Suppose the distribution of $L_{(T,h_0)}$ be $\{\{\alpha_j\}_{I_0}, \{\alpha_j\}_{I_1}, \cdots, \{\alpha_j\}_{I_s}\}$. By definition, the tree with layer-decomposition $L_{(T,h_0)}$ can be generated layer by layer from its root gradually. In layer 0, there is only one vertex, the root with degree α_0. In layer 1, there are $k_1 = \alpha_0$ vertices which are incident with the root and the distribution in layer 1 is $\{\alpha_j\}_{I_1}$. In layer 2, there are $k_2 = \sum_{j\in I_1}(\alpha_j - 1)$ vertices and each vertex is incident with one of k_1 vertices in layer 1. The distribution in layer 2 is $\{\alpha_j\}_{I_2}$. In general, for any $1 \le l \le s$, there are $k_l = \sum_{j\in I_{l-1}}(\alpha_j - 1)$ vertices in layer l and each vertex in layer l is incident with one of vertices in layer $l-1$. The distribution in layer l is $\{\alpha_j\}_{I_l}$. Hence, for any subtree with layer-decomposition $L_{(T,h_0)}$ and distribution $\{\{\alpha_j\}_{I_0}; \{\alpha_j\}_{I_1}; \cdots; \{\alpha_j\}_{I_s}\}$, we have, in G, at most n choices to produce its root, at most $A_\Delta^{\alpha_0}$ choices to produce its 1st layer, at most $\prod_{j\in I_1} A_{\Delta-1}^{\alpha_j-1}$ choices to produce its 2nd layer, generally for $1 \le l \le s$, at most $\prod_{j\in I_{l-1}} A_{\Delta-1}^{\alpha_j-1}$ choices to produce its lth layer. So, the total complexity

to pick out all subtrees in G with distribution $\{\{\alpha_j\}_{I_0}; \{\alpha_j\}_{I_1}; \cdots; \{\alpha_j\}_{I_s}\}$, denoted by Ω, is that:

$$\begin{aligned}
\Omega &\leq nA_{\Delta}^{\alpha_0}\prod_{j\in I_1}A_{\Delta-1}^{\alpha_j-1}\prod_{j\in I_2}A_{\Delta-1}^{\alpha_j-1}\cdots\prod_{j\in I_{s-1}}A_{\Delta-1}^{\alpha_j-1}\\
&\leq n\Delta^{\alpha_0}\prod_{j\in I_1}(\Delta-1)^{\alpha_j-1}\prod_{j\in I_2}(\Delta-1)^{\alpha_j-1}\cdots\prod_{j\in I_{s-1}}(\Delta-1)^{\alpha_j-1}\\
&\leq n\left(\frac{\Delta}{\Delta-1}\right)^{\Delta}(\Delta-1)^{[1+\sum_{j\in I_0\cup I_1\cup\cdots\cup I_{s-1}}(\alpha_j-1)]}\\
&= O(n(\Delta-1)^{k-1}),
\end{aligned}$$

where the last equality holds is from (2). We complete the proof. ■

Remark: In practical applications such as the VLSI circuit structure analysis, the Δ may be great occasionally. However, there are few vertices whose degrees are Δ or so. In the algorithm for SRP, The mean degree of G, $\bar{\Delta}$, instead of Δ being used is more reasonable.

5. Numerical Test

In order to check the efficiency of the Algorithm, we make the numerical test on the computer with Pentium MMX-166 processor and 40MB RAM. We set $\Delta \leq 10$ and select random G and H with various sizes. We list the result in the following table. In the table, $No.$ is the serial number we make test, n is the order of G, k is the order of H and $CPUtime(second)$ is the total seconds CPU spending at finding and listing all subgraphs within G which are isomorphic to H.

No.	n	k	CPU time (second)
1	5004	44	878
2	4002	45	634
3	3000	52	227
4	2002	66	462
5	2002	51	233
6	1004	64	175
7	750	54	120
8	501	54	81
9	501	36	33
10	261	51	34
11	261	36	14
12	173	35	12
13	173	21	4

6. Conclusion

In this paper, we propose the idea of layer-decomposition with degree and establish an algorithm to detect subgraphs of graph G isomorphic to another graph H. Theorem 1 and numerical test show that the algorithm is very effective in case the degree of G and the order of graph H are not so large. The biggest advantage of the algorithm is that the complexity of the algorithm is linear w.r.t. the order of G and the method does not depend on how the layer-decomposition is taken. However, when the order of H is large the complexity of the algorithm would be very high. How to construct a method to handle this case, we leave it as a open problem.

Acknowledgements

The author wishes to thank anonymous referees for their valuable comments for revision.

References

1. D. J. Brown, M. R. Fellows and M. Langston, Polynomial-time self reducibility: Theoretical motivations and practical results. *Int. J. Compu. Math.* **31** (1989), 1-9.
2. N. Chiba and T. Nishizeki, Arboricity and subgrapph listing algorithm, *SIAM J. Computing* **14:1** (1985), 210-223.

3. M. Fontet, A linear algorithm for testing isomorphism of planar graphs, *Proceedings 3rd Colloquium on Automata, Languages and Programming*, (1976) 411-423.

4. B. Monien, How to find long paths efficiently, *Ann. Disc. Math.* **5** (1985), 239-254.

5. J. Nesetril and S. Poljak, On the complexity of the subgraph problem. *Comm. Math. Univ. Carol.* **26** (1985), 415-419.

6. J. Plehn and B. Voigr, *Finding minimally weighted subgraphs*, Workshop on Graph Theoretic Concepts in Computer Sciences (1990).

7. D. Richards, Finding short cycles in planar graphs using seperators, *J. Algorithms* **7** (1986), 382-394.

8. D. Richards and A. Liestman, Finding short cycles of a given length, *Ann. Discr. Math.* **27** (1985), 249-256.

9. N. Robertson and P. D. Deymour, Graph minor II: algorithmic aspects of tree-width, *J. Algorithm* **7** (1986), 309-322.

10. G. Sundaram and S. S. Skiena, Recognizing small subgraphs, *Networks*, **25** (1995), 183-191.

11. I. A. Stewar, Finding regular subgraphs in both arbitrary and planar graphs, *Discrete Applied Math.* **68** (1996), 223-235.

12. Y. F. Xu, *Detect Subgraphs by means of Layer-Decomposition*, Postdoctoral Working Report, Institute of Automation, Chinese Academy of Sciences, 1999.

Yifan Xu
School of Management, Fudan University, Shaighai, 200433, P.R.China
E-mail: yifanxu@online.sh.cn