氏　　　名	千（ち）葉（ば）則（のり）茂（しげ）
授与学位	工学博士
学位授与年月日	昭和59年3月27日
学位授与の根拠法規	学位規則第5条第1項
研究科，専攻の名称	東北大学大学院工学研究科 （博士課程）情報工学専攻
学位論文題目	Algorithms for Planar Graphs （平面グラフ　アルゴリズム）
指導教官	東北大学教授 斎藤　伸自
論文審査委員	東北大学教授 斎藤　伸自　　東北大学教授 木村　正行 東北大学教授 伊藤　貴康　　東北大学助教授 西関　隆夫

論文内容要旨

Chapter 1　Introduction

Recent research efforts in algorithm theory have concentrated on designing efficient algorithms for solving combinatorial problems, particularly graph problems. Graph problems often arises in various areas of engineering and computer science. Thus it would be important to give practical algorithms for typical problems. However, it has been shown that many of practically important problems belong to the class of "NP−complete" problems.
A "polynomial−time" algorithm for any one of these NP−complete problems can be effectively translated into polynomial−time algorithms for all other problems in this class, and all known algorithms for solving NP−complete problems run in exponential time. Given these facts, there appears to be little hope of being able to design polynomial−time algorithms for exactly solving any NP−complete problem.

Almost all graphs appearing in a particular area are often contained in a special "subclass" of graphs, for example, planar graphs in the case of traffic networks and series−pararel graphs in the case of electrical networks. Hence, it would be useful to design efficient algorithms for a special class of graphs. All known exact algorithms for solving NP−complete problems run *in exponentiai time. However, an approximate solution is often sufficient* for some practical purpose. Thus, an "approximation approach" is useful for NP−complete problems. Furthermore, an approximation approach is expected to be useful even for polynomial time solvable problems by the same reason.

Algorithms are usually evaluated by their running time and storage space, since these are dominant factors of efficiency. The notion of practicality of algorithms must involve all the various resources needed for implementing and executing algorithms. Thus the simplicity and comprehensibility of algorithms are also significant for the practicality.

Restricting the input graphs into the class of planar graphs, we present very simple and efficient algorithms for the following eight graph problems containing five NP−complete problems: the plane embedding problem, the graph drawing problem, the coloring problem, the maximum independent set problem, the maximum clique problem, the maximun induced subgraph problem, *the minimun vertex cover problem and the maximun matching problem.* The time complexity of these algorithms is $O(n)$, $O(n \log n)$ or $O(n^2)$ at worst, where n is the number of vertices of a graph. These are superior to or at least as good as the best known algorithms, and are often improved in simplicity. We also introduce new simple algorithmic techniques useful especially for combinatorial problems on planar graphs.

Chapter 2 Embedding planar Graphs

Planarity testing, that is, determining whether a given graph is planar or not, has many applications, such as the design of VLSI circuits and determining isomorphism of chemical structures. Two planarity testing algorithms of different types are known, both running in linear time. One is called a "path addition algorithm", and the other a "vertex addition algorithm". The path addition algorithm was first presented by Auslander, Parter and Goldstein, and improved later into a linear algorithm by Hopcroft and Tarjan. The vertex addition algorithm was first presented by Lempel, Even and Cederboum, and improved later into a linear algorithm by Booth and Lueker employing an st−numbering algorithm and a data structure called a "PQ−tree".

Many applications require not only testing the planarity but also embedding (or drawing) a planar graph in the plane. Hopcroft and Tarjan mentioned that an embedding algorithm can be constructed by modifying their testing algorithm. However the modification looks to be fairly complicated, paticularly it is quite difficult to implement a part of the algorithm for embedding an intractable path called a "special path".

In this chapter we present a very simple linear algorithm for embedding planar graphs, which is based on the vertex addition algorithm of Booth and Lueker. If a planar graph is not 3-connected then an embedding of the graph is not unique. We also show that our algorithm can be easily modified so as to construct an expression for all the embeddings of a planar graph.

Chapter 3 Drawing Planar Graphs

The problem of drawing a planar graph often arises in many applications, including Design Automation of VLSI circuits. In this chapter we are not interested in a specific practical application, but interested in producing a pleasing drawing of a given planar graph. Restricting given graphs to trees, some recent papers have studied the problem of producing well-shaped drawings of trees. Obviously there are no absolute criteria that accurately capture our intuitive notion of nice drawings of planar graphs. However, it seems that the following are desirable properties of pleasing drawings:

(a) all the edges are drawn by straight line segments without crossing lines;

(b) face boundaries are drawn by convex polygons as far as possible;

(c) boundaries of 3-connected components are drawn by convex polygons.

We present two linear algorithms for the convex drawing problem of planar graphs: convex drawing and testing algorithms. The first draws a given planar graph convex if possible. The second tests the possibility, that is, determines whether a givn planar graph has a convex drawing or not. We also present a drawing algorithm which uses the convex drawing algorithm, and produces a pleasing drawing satifying properties (a)-(c).

Chapter 4 Five-Coloring

A coloring of a graph is an assignment of colors to the vertices in such a way that adjacent vertices have distinct colors. Although the problem of coloring a graph with the minimal number of colors has practical applications in some schedulings, it is known to be NP-complete even for the class of planar graphs.

We present here a linear algorithm for finding a coloring of a planar graph with at most five colors, that is, 5-coloring. Based on the well-known Kempe-chain argument, one can easily design an $O(n^2)$ time algorithm for the purpose by employing a simple recursive reduction of a graph involving the deletion of a vertex of degree 5 or less possibly together with the interchange of colors in a two-colored subgraph. Lipton and Miller have given an $O(n \log n)$ algorithm for the problem by removing a "batch" of vertices rather than just a single vertex. Their algorithm and its proof are a little complicated. In this chapter we give a simple linear algorithm for the purpose. The algorithm does not use the Kempe-chain argument, but uses a recursive reduction of a graph involving the deletion of a vertex of degree 6 or less possibly together with the identification of several neighbors of the vertex.

Chapter 5 Independent Sets

A subset of the vertices of a graph is independent if no two vertices in the set are adjacent. The maximum independent set problem, in which one would like to find a maximum independent set in a given graph, is NP-complete, and still remains so even if restricted to the class of planar graphs. In this chapter we give two simple and efficient algorithms for the purpose.

The *four-color theorem implies that every planar graph has an independent* set containing at least $n/4$ vertices. Albertson showed, independently of the four-color theorem, that a planar graph has an independent set containing more than $2n/9$ vertices, and furthermore proposed an "algorithm" for actually finding such a set. Unfortunately the straightforword analysis cannot guarantee the polynomial-boundedness of the algorithm, due to only one troublesome step. We give an $O(n^2)$ time algorithm for the same purpose, by modifying his algorithm, mainly avoiding the troublesome step.

An approximation algorithm is often evaluated by the worst case ratio: the smallest ratio of the size of an approximation-solution to the size of a maximum solution, where the ratio is taken over all problem instances. Lipton and Tarjan have given an $O(n \log n)$ time approximation algorithm with worst case ratio $1-O(1/(\log\log n)^{1/2})$, asymptotically tending to 1 as $n \to \infty$, for the problem on a planar graph with n vertices. Such a ratio is called an "asymptotic worst case ratio". On the other hand, some approximation algorithms have an "absolute worst case ratio", which does not depend on the size n of a graph. For example, the 5-coloring algorithm in Chapter 4 provides a linear time approximation algorithm with absolute worst case ratio 1/5. We present an $O(n \log n)$ time approximation algorithm with absolute worst case ratio 1/2 for the maximum independent set problem on planar graphs.

Chapter 6 Cliques

The problems to list certain kind of subgraphs of a graph arise in many practical applications. In this chapter we introduce a new simple strategy into edge-searching of a graph, which is useful to the various subgraph listing problems. We choose a vertex v in a graph and scan the edges of the subgraph induced by the neighbors of v to find the pattern subgraphs containing v. The feature of the strategy is to repeat the searching above for each vertex v in nonincreasing order of degree and to delete v after v is processed so that no duplication occurs. We will show that the procedure above requires $O(a(G)m)$ time. Throughout this chapter $a(G)$ is the arboricity of G, that is, the minimum number of edge-disjoint spanning forests into which G can be decomposed. We use the rather unfamiliar graph invariant $a(G)$ as a parameter in bounding the running time of algorithms.

The strategy yields simple algorithms for the problems to list certain kinds of subgraphs of a graph. The kinds of these subgraphs include "triangle", "quadrangle", "clique of a fixed order", and "maximal clique". Our algorithms are as fast as the known ones if any, and a factor n is often reduced to $a(G)$ in the time complexity. All our algorithms require linear space and exceed the known algorithms for the same purposes in running time, storage space, or simplicity.

Chapter 7 Applications of planar Separator Theorem

Lipton and Tarjan have given a planar separator theorem which provides a basis for exploiting the divide-and-conquer paradigm. A number of combinatorial problems, including the maximum independent set problem, are formulated as a "maximum induced subgraph problem" with respect to some graph property Q. Furthermore it has been shown in a unified way that the maximum induced subgraph problem together with the approximation problem is NP-complete for general graphs if Q satisfies some conditions. As an application of the planar separator theorem, We first present efficient $O(n \log n)$ time approximation algorithms for a broad class of the maximum induced subgraph problems. We next give an approximation algorithm with time complexity $O(n \log n)$ for the maximum matching problem on planar graphs, which is polynomial-time solvable (the best known exact algorithm has time complexity $O(n^{1.5})$ for planar graphs). We finally present an $O(n \log n)$ time approximation algorithm for the minimum vertex cover problem, which is NP-complete even for planar graphs. The worst case ratio of these algorithms is $1-O(1/(\log\log n)^{1/2})$ asymptotically tending to 1 as $n \to \infty$.

Chapter 8 Conclusions

This thesis gave simple and efficient algorithms for various combinatorial problems on planar graphs by employing several techniques, such as efficient algorithmic tools, new concise and constructive theorems or proofs on graphs, approximation approaches, and sophisticated time analyses.

審 査 結 果 の 要 旨

工学の分野における多くの問題はネットワークモデルを利用して解かれることが多い。ネットワークの構造的あるいは位相幾何学的な性質を論じるのにグラフ理論が有効である事が知られている。グラフ問題に対する計算機アルゴリズムは計算時間の点から実用上問題となるものが多い。

著者は，VLSIのCADを初めとして応用上よく現われる平面グラフに着目し，平面グラフの持つ固有の性質を利用して，効率がよくかつ明解な構造を持ったアルゴリズムを数多くの典型的なグラフ問題に対して与えた。本論文はその成果をまとめたもので，全編8章よりなる。

第1章は序論である。

第2章は平面グラフを平面上に埋め込む線形時間アルゴリズムを与えている。従来のアルゴリズムと比べて，簡明な構造をしている。

第3章では平面グラフを描画する線形時間アルゴリズムを与えている。まず，平面グラフが凸描画可能か否かを判定し，可能ならば具体的に凸描画するアルゴリズムを与えた。次いで，凸描画とは限らないが，構造的特徴を把握し易いように描画するアルゴリズムを与えた。

第4章では平面グラフの5－点彩色問題について線形時間アルゴリズムを与えている。従来のオーダ$O(n \log n)$のものより高速で，かつ単純な構造をしている。ここでnはグラフの点数である。

第5章では平面グラフの最大独立点集合を求める2つの近似アルゴリズムを与えている。1つは，最大独立点集合の半分より大きいものを求める$O(n \log n)$時間アルゴリズムであり，もう1つは$\frac{2}{9}n$個以上の独立点を求める$O(n^2)$時間アルゴリズムである。これらは興味深いアルゴリズムである。

第6章ではグラフの枝探索に新しい手法を導入し，3角形，4角形，クリーク，極大完全グラフを列挙する単純で高速なアルゴリズムを与えている。

第7章では最大部分グラフ問題，最小点被覆問題及び最大マッチング問題に対し，最悪値比が漸近的に1に近づくような$O(n \log n)$時間の近似アルゴリズムを系統的に与えている。

第8章は結論である。

以上要するに，本論文はグラフ問題に対する効率のよいアルゴリズムを設計するのに有用な種々の手法を提案し，平面グラフの諸問題に対して効率がよいすぐれたアルゴリズムを与えたもので，情報工学の発展に寄与するところが少なくない。

よって，本論文は工学博士の学位論文として合格と認める。