数理解析研究所講究録
第427巻 1981年 148-164

Applications of the Lipton and Tarjan's planar separator theorem

by

Norishige Chiba, Takao Nishizeki and Nobuji Saito

Department of Electrical Communications
Faculty of Engineering, Tohoku University
Sendai, Japan 980

Abstract. Using the planar separator theorem of Lipton and Tarjan, we give approximation algorithms with time complexity O(nlogn) and asymptotic worst-case ratio tending to 1 for the following problems on planar graphs: the maximum induced subgraph problems with respect to ceartain graph properties; the maximum matching problem; and the minimum vertex cover problem.

1. Introduction

Lipton and Tarjan have given a planar separator theorem which provides a basis for exploiting the divide-and-conquer paradigm [6]. As an application of the separator theorem, they have presented an approximation algorithm for the maximum independent set problem with time complexity O(nlogn) and worst-case ratio $1-O(1/\sqrt{\log\log n})$ asymptotically tending to 1 as $n\rightarrow\infty$ [7]. In this paper n denotes the number of vertices of a graph, and the worst-case ratio is defined to be the smallest ratio of the size of an approximate-solution to the size of a maximum solution taken over all problem instances.

We wish to study the general conditions under which the Lipton and Tarjan's approach is useful for combinatorial problems on planar graphs. 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 [5][11]. In this paper we first observe the requirements for their approach to yield an efficient approximation algorithm for the maximum induced subgraph problem on planar graphs. It immediately follows that there exist efficient approximation algorithms for a broad class of the maximum induced subgraph problems. We next give an approximation algorithm with time complexity O(nlogn) 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 [7][8]). We finally present an O(nlogn) time approximation algorithm for the minimum vertex cover problem, which is NP-complete even for planar graphs. In the latter two algorithms, we reduce the problems on general planar graphs to those on planar graphs with minimum degree 3 so that the Lipton and Tarjan's approach is successful. Our terminology on algorithms and graphs is standard; all undefined terms should be refered to [1] or [4].

2. The maximum induced subgraph problem

In this section we show in a unified manner that there exist efficient approximation algorithms for a broad class of combinatorial problems: the maximum induced subgaph problems with respect to various graph properties Q.

Lipton and Tarjan have given an approximation algorithm for the maximum independent set problem, using the following form of their separator theorem.

THEOREM 1. (Lipton and Tarjan[7]) Every planar graph of n vertices contains a set C of $O(\sqrt{n/\epsilon})$ vertices whose removal leaves no connected component with more than ϵn vertices, where $0<\epsilon<1$. Furthermore the set C can be found in O(nlogn) time.

A number of problems on a graph G=(V,E) are formulated as a maximum induced subgraph problem P with some graph property Q; the problem asks for a maximum set S of V that induces a subgraph satisfying Q. For example, the maximum independent set problem is a maximum induced subgraph problem P with property "independent (i.e. pairwise nonadjacent)". From Lipton and Tarjan's approximation algorithm for the maximum independent set problem, we naturally have the following "algorithm" MISP for the maximum induced subgraph problem P with Q.

Algorithm MISP(ϵ).

Comment $0<\epsilon<1$.

Step 1. Applying Theorem 1 to a given graph $G=(V,E)$, find a set of vertices C of size $O(\sqrt{n/\epsilon})$, such that each connected component of G-C (i.e. the graph obtained from G by deleting the vertices in C) contains at most ϵn vertices.

Step 2. In each connected component $G_i=(V_i,E_i)$ of G-C, find a maximum set S_i inducing a subgraph with property Q by checking every subset of V_i.

Step 3. Form S as a union of maximum sets, one from each component, that is, $S=\Sigma S_i$.

One can easily observe that the conditions given in Lemma 1 below are sufficient for the success of the algorithm MISP.

LEMMA 1. The approximation algorithm MISP(ϵ) has time complexity $O(\max\{n\log n, n2^{\epsilon n}\})$ and worst-case ratio $1-O(1/\sqrt{\epsilon n})$ if the following conditions are satisfied:

(C1) The subgraph of G induced by S satisfies property Q.

(C2) The error $|S^*|-|S|$ is bounded by the number of vertices of C, that is, $|S^*|-|S|=O(|C|)$, where S^* is a maximum vertex set inducing a subgraph of G with Q.

(C3) $|S^*|$ is a positive fraction of n.

(C4) Property Q is recognizable in linear time (, that is, one can determine in linear time whether a graph satisfies Q or not).

Proof. Similar to the proof of Lipton and Tarjan's algorithm in [7]. Q.E.D.

A graph property Q is hereditary if every subgraph of G satisfies Q whenever G satisfies Q. Q is determined by the components if a graph G satisfies Q whenever every connected component of G satisfies Q. For example, the property "planar" is hereditary and determined by the components since every subgraph of a planar graph is planar and a graph is planar if and only if every connected component is planar. It has been known that a number of graph properties are hereditary and determined by the components [11]. In order to avoid a trivial case, we now assume that at least one nonempty graph satisfies Q.

We have the following theorem from Lemma 1.

THEOREM 2. If P is a maximum induced subgraph problem with respect to property Q which is

(a) hereditary;

(b) determined by the components; and

(c) recognizable in linear time,

then there exists an approximation algorithm for P on planar graphs with time complexity O(nlogn) and worst-case ratio $1-O(1/\sqrt{\log\log n})$.

Proof. We claim that the conditions (C1)-(C4) in Lemma 1 are all satisfied. If so, we immediately have the theorem from Lemma 1 by setting $\epsilon=(\log\log n)/n$.

Since Q is determined by components, Q holds for the union S of the solutions S_i of the components of $G-C$. Therefore the condition (C1) is satisfied.

Next consider a maximum set S^* inducing a subgraph with Q in G. Since Q is hereditary, $S'=S^*-C$ induces a subgraph satisfying Q in G, and furthermore each $S_i'=S^*\cap V_i$ induces a subgraph of $G_i=(V_i,E_i)$ satisfying Q. Thus we have $|S_i| \geq |S_i'|$ since S_i is a maximum set of V_i inducing a subgraph with Q in G_i. Therefore we have

$$|S| = \Sigma|S_i| \geq \Sigma|S_i'| = |S'| \geq |S^*|-|C|,$$

so the condition (C2) is satisfied.

Since at least one nonempty graph satisfies the hereditary property Q, K_1 (the single vertex graph) satisfies Q. Since Q is determined by the components, a maximum independent set I^* of the planar graph G satisfies Q. Therefore

$$|S^*| \geq |I^*| \geq n/4.$$

The last inequality follows from the four color theorem. Thus the condition (C3) is satisfied. The condition (C4) is identical to (c). Q.E.D.

If Q is recognizable in polynomial-time instead of linear time, there exists a polynomial-time approximation algorithm with the same worst-case ratio although the time complexity is no longer $O(n\log n)$. The following corollary is an immediate consequence of Theorem 2.

COROLLARY. For n-vertex planar graphs, there exist approximation algorithms with time complexity $O(n\log n)$ and worst-case ratio $1-O(1/\sqrt{\log\log n})$ for the maximum induced subgraph problems with respect to the following properties among others:

(1) independent;

(2) bipartite;

(3) forest; and

(4) outerplanar.

3. The maximum matching problem

The maximum matching problem asks for a maximum number of pairwise nonadjacent edges in a graph. Although there exists an $O(n^{1.5})$ exact algorithm for the maximum matching problem on planar graphs [7][8], the efficient approximation algorithm would be useful if the available computation time is limited and an approximate maximum matching is sufficient for some practical purpose. In this section, using Algorithm MISP, we give an O(nlogn) approximation algorithm for this problem.

The problem is an instance of the maximum subgraph problem with property Q, which is defined similarly as the maximum induced subgraph problem in the preceding section. (The assertion on the maximum subgraph problem similar to Theorem 2 does not always hold true.) One can easily see that the conditions (C1)-(C4) are also sufficient for the success of Algorithm MISP to the maximum matching problem, where S and S* should be subsets of the edge set E (instead of V) of a given graph G=(V,E). Clearly the conditions (C1) and (C4) are satisfied, and (C2) is also satisfied as will be shown later. However (C3) is not satisfied since a planar graph does not always contain a maximum matching of linear size as $K_{1,n-1}$ or $K_{2,n-2}$ indicate [9]. Thus a direct application of MISP cannot gurantee the worst-case ratio $1-O(1/\sqrt{\log\log n})$. However if the problem on a general planar graph can be reduced to the same problem on a particular planar graph having a matching of linear size, for example a planar graph with minimum degree 3 [9], then a modified MISP may guarantee the desired worst-case ratio.

The following is the approximation algorithm MATCHING for the maximum matching problem on planar graphs, which is based on the idea above. The algorithm finds a matching S(G) of a given graph G. We denote by deg(v) the degree of a vertex v, and by S*(G) a maximum matching of graph G.

```
procedure MATCHING;
 procedure REDUCE(G,S(G));
  begin
   let G contain N vertices, and let v be a vertex of minimum
   degree of G;
   if N < loglogn
    then find a maximum matching S*(G) of G by applying any
         reasonable (polynomial or exponential time) exact
         algorithm and let S(G):=S*(G)
    else if deg(v) < 3
          then if deg(v)=0
                then
                 begin
                  G':=G-v;
                  REDUCE(G',S(G'));
                  S(G):=S(G')
                 end
                else if deg(v)=1
                      then
                       begin
                        let u be the vertex adjacent to v;
                        G':=G-{u,v};
                        REDUCE(G',S(G'));
                        S(G):=S(G')∪{(v,u)}
                       end
                      else comment deg(v)=2
                       begin
                        let u,w be the vertices adjacent to v;
                        let G' be the graph obtained from G by
                        identifying the three vertices v,u and
                        w;
                        REDUCE(G',S(G'));
                        if S(G') contains no edge which was
                           adjacent to u in G
                          then S(G):=S(G')∪{(v,u)}
                          else S(G):=S(G')∪{(v,w)};
                       end
          else comment deg(v) ≥ 3 and N > loglogn;
            apply Algorithm MISP with ε=(loglogn)/N to G to
            obtain a matching S(G)
  end REDUCE;
```

```
begin
  let G be a given planar graph of n vertices;
  REDUCE(G,S(G))
end.
```

THEOREM 3. The approximation algorithm MATCHING for the maximum matching problem on planar graphs has worst-case ratio $1-O(1/\sqrt{\log\log n})$ and time complexity $O(n\log n)$.

Proof. (a) Correctness and worst-case ratio.

The algorithm MATCHING reduces a graph G to a smaller one G' whenever G contains more than nlogn vertices including a vertex of degree 2 or less. So one eventually arrives at a graph G_0 of N ($\le$ loglogn) vertices or of minimum degree $\ge$ 3.

We first show that MATCHING correctly finds a matching of the graph G_0 within the desired worst-case ratio. If $N \le \log\log n$, then MATCHING finds a maximum matching of G_0 by applying an exact algorithm, so $|S(G_0)|/|S^*(G_0)|=1$. (Note that we set $|S(G)|/|S^*(G)|$ to 1 if $|S^*(G)|=0$.) Otherwise (that is, if $N > \log\log n$ and G_0 is of minimum degree ≥ 3) MATCHING applies Algorithm MISP with $\epsilon=(\log\log n)/N$ to find a matching $S(G_0)$. We shall show that the conditions (C1)-(C4) of Lemma 1 are all satisfied in this case. Since G_0 is a planar graph with minimum degree $\ge$ 3, we have $|S^*(G_0)| \ge N/3$ [9], so the condition (C3) of Lemma 1 is satisfied. Clearly the conditions (C1) and (C4) are satisfied. Finally (C2) is satisfied: all the edges of $S^*(G_0)$ not contained in components of G_0-C are adjacent to vertices of C; furthermore each vertex of C is adjacent to at most one edge in $S^*(G_0)$; therefore $|S^*(G_0)|-|S(G_0)| \le |C|$. Thus in this case Algorithm MISP with $\epsilon=(\log\log n)/N$ correctly finds a matching $S(G_0)$ within the worst-case ratio $1-O(1/\sqrt{\log\log n})$.

We shall next show that MATCHING correctly forms an approximate matching S(G) of G from a matching S(G') of a smaller graph G' with preserving the desired worst-case ratio, that is, $|S(G)|/|S^*(G)| \geq |S(G')|/|S^*(G')|$. Consider the following three cases according to the employed reductions.

<u>Case 1: G contains an isolated vertex v, i,e, a vertex of degree 0.</u>

If S(G') is a matching of G'=G-v, then S(G)=S(G') is a matching of G. Furthermore $|S(G)|/|S^*(G)| = |S(G')|/|S^*(G')|$.

<u>Case 2: G contains a vertex v of degree 1.</u>

Let u be the vertex adjacent to v in G, and let $G'=G-\{u,v\}$. If S(G') is a matching of G', then clearly $S(G)=S(G')\cup\{(u,v)\}$ is also a matching of G. Thus $|S(G)| = |S(G')|+1$. Since a maximum matching $S^*(G)$ of G contains exactly one edge adjacent to u, we have $|S^*(G)| \leq |S^*(G')|+1$. Therefore we have $|S(G)|/|S^*(G)| \geq |S(G')|/|S^*(G')|$.

<u>Case 3: G contains a vertex of degree 2.</u>

Let u,w be the vertices adjacent to v, and let G' be the graph obtained from G by identifying u,v, and w into a single vertex. Let $S(G)=S(G')\cup\{(v,u)\}$ if a matching S(G') of G contains no edge which was adjacent to u in G, and otherwise let $S(G)=S(G')\cup\{(v,w)\}$. Clearly S(G) is a matching of G if S(G') is so in G'. Thus we have $|S(G)| = |S(G')|+1$. If a maximum matching $S^*(G)$ of G contains either (v,u) or (v,w), say (v,u), then $S^*(G)-(v,u)$ is a matching of G'. Otherwise, $S^*(G)$ contains an edge (u,x) with $x \neq v$, and then $S^*(G)-(u,x)$ is a matching of G'. Thus we have $|S^*(G)| \leq |S^*(G')|+1$. These two equations imply $|S(G)|/|S^*(G)| \geq |S(G')|/|S^*(G')|$.

Thus we have verified the correctness and the worst-case ratio.

(b) Time complexity.

First consider the computation time required to the graph G_0. Let G_0 contain N vertices. If $N \leq \log\log n$, MATCHING applies to G_0 an exact algorithm. Even if the exact algorithm has time complexity $O(N2^N)$, it requires at most $O(n\log n)$ time. If $N > \log\log n$ and G_0 is of minimum degree ≥ 3, then the algorithm MISP with $\epsilon=(\log\log n)/N$ applied to G_0 requires at most $O(\max\{N\log N, N2^{\epsilon N}\}) = O(N\log n) \leq O(n\log n)$ time.

Next consider the time required for reducing a given graph G to the graph G_0. The exhaustive operations are vertex-deletions and vertex-identifications appeared in the reductions. Since one can execute a single vertex-deletion of a vertex v in $O(\deg(v))$ time using the adjacency lists of a graph, and every vertex appears in at most one vertex-deletion, all the vertex-deletions involved in the execution of MATCHING requires at most $O(n)$ time. On the other hand, any sequence of vertex-identifications can be executed in $O(n\log n)$ time by using one of the following devises: adjacency lists together with an adjacency matrix [2]; or AVL trees together with an efficient list merging algorithm [10].

Finally it is clear that the time required for forming $S(G)$ from $S(G')$ is, in total, as much as the time required for the reductions. Thus we can implement the algorithm MATCHING to run in at most $O(n\log n)$ time. Q.E.D.

4. The minimum vertex cover problem

The *minimum vertex cover problem* asks for a minimum number of vertices such that every edge of a graph is incident to at least one of the vertices. In this section we give an approximation algorithm for this problem.

The minimum vertex cover problem and the maximum independent set problem are quite closely related. For a graph $G=(V,E)$, a vertex set $V' \subset V$ is a (maximum) independent set of G if and only if the complementary set $V-V'$ is a (minimum) vertex cover set of G. However one can easily see that this transformation does not preserve the worst-case ratio [3, p. 134]. So we must design an approximation algorithm for each problem. We will show that the reduction given in Section 3 is useful also for the minimum vertex cover problem. The following is the algorithm for the minimum vertex cover problem on planar graphs.

```
procedure COVER;
 procedure REDUCE(G,C(G));
  begin
   let G be contain N vertices, and let v be a vertex of
   minimum degree of G;
   if N < loglogn
    then obtain a minimum vertex cover set C*(G) of G by
         cheking every subset of vertices and let C(G):=C*(G)
    else if deg(v) < 3
           then
            if deg(v)=0
             then
              begin
               G':=G-v;
               REDUCE(G',C(G'));
               C(G):=C(G')
              end
             else
              if deg(v)=1
```

```
          then
           begin
             let u be the vertex adjacent to v;
             G':=G-{u,v};
             REDUCE(G',C(G'));
             C(G):=C(G')+u
           end
          else comment deg(v)=2
           begin
             let u,w be the vertices adjacent to v;
             let G' be the graph obtained from G by
             identifying the three vertices v,u and w into
             a new vertex x;
             REDUCE(G',C(G'));
             if x∈C(G')
                then C(G):=C(G')-x+{u,w}
                else C(G):=C(G')+v;
           end
      else comment deg(v) ≥ 3 and N > loglogn
       begin
        apply the Lipton and Tarjan's algorithm with
        ε=loglogn/N to find an approximate independent set
        I(G) of G;
        C(G):=V-I(G)
       end;
end REDUCE;
begin
 let G be a given planar graph with n vertices;
 REDUCE(G,C(G))
end.
```

THEOREM 4. The approximation algorithm COVER for the minimum vertex cover problem has worst-case ratio $1+O(1/\sqrt{\log\log n})$ and time complexity $O(n\log n)$.

Proof. Clearly the algorithm has time complexity $O(n\log n)$, which is same as MATCHING. As similar to the proof of Theorem 3, one can verify the correctness and the worst-case ratio of the algorithm. Consider only the case when G_0 is of $N \geq \log\log n$ vertices and with minimum degree 3 or more. In this case a maximum matching of G_0 contains at least $N/3$ edges [9]. Therefore a minimum vertex cover $C^*(G_0)$ of G_0 contains at least $N/3$ vertices. Hence

$$
\begin{aligned}
|C(G_0)|/|C^*(G_0)| &= (N-|I(G_0)|)/(N-|I^*(G_0)|) \\
&= 1+(|I^*(G_0)|-|I(G_0)|)/(N-|I^*(G_0)|) \\
&\leq 1+O(1/\sqrt{\log\log n}).
\end{aligned}
$$

The last inequality follows from $|C^*(G_0)| = N-|I^*(G_0)| \geq N/3$ and $|I^*(G_0)|-|I(G_0)| \leq |C|=O(N/\sqrt{\log\log n})$, which is ensured by Lipton and Tarjan's algorithm with $\varepsilon=\log\log n/N$. Thus in this case the algorithm COVER correctly finds a vertex cover within the desired worst-case ratio. Furthermore each reduction clearly preserves the worst-case ratio, that is, $|C(G)|/|C^*(G)| \leq |C(G')|/|C^*(G')|$. Q.E.D.

Acknowledgement.

We wish to thank an anonymous referee for spotting an error in the early version of this note. This work was partly supported by the Grant in Aid for Scientific Research of the Ministry of Education, Science and Culture of Japan under Grant: Cooperative Research (A) 435013 (1980).

REFERENCES

[1] A. V. Aho, J. E. Hopcroft, and J. D. Ullman, The Design and Analysis of Computer Algorithms, Addison-Wesley, Reading, Mass., 1974.

[2] N. Chiba, T. Nishizeki and N. Saito, An approximation algorithm for the maximum independent set problem on planar graphs, submitted to SIAM J. on Comput., 1980.

[3] M. R. Garey and D. S. Jhonson, Computers and Intractability, W. H. Freeman and Company, 1979.

[4] F. Harary, Graph Theory, Addison-Wesley, Reading, Mass., 1972 (revised).

[5] M. S. Krishnamoorthy and N. Deo, Node-deletion NP-complete problems, SIAM J. on Comput., 8, 4, pp. 619-625, 1979.

[6] R. J. Lipton and R. E. Tarjan, A separator theorem for planar graphs, SIAM J. Appl. Math., 36, pp. 177-189, 1979.

[7] R. J. Lipton and R. E. Tarjan, Applications of a planar separator theorem, SIAM J. on Comput., 9, 3, pp. 615-627, 1980.

[8] S. Micali and V. V. Vazirani, An $O(\sqrt{|V|}|E|)$ algorithm for finding maximum matching in general graphs, 21th Ann. Symp. on Found. of Comp. Sci., pp. 17-27, 1980.

[9] T. Nishizeki and I. Baybars, Lower bounds on the cardinality of the maximum matchings of planar graphs, Discrete Mathematics, 28, pp. 255-267, 1979.

[10] Y. Shiloach, Union-member algorithms for non-disjoint sets, Technical Report STAN-CS-728, Computer Science Department, Stanford University, 1979, submitted to Journal of ACM.

[11] M. Yannakakis, Node- and edge-deletion NP-complete problems, Proc. of the 10th ACM Symp. on Theory of Computing, pp. 253-264, 1978.